
This is the **published version** of the bachelor thesis:

Juvillà Belmonte, Lluna. *Implementació d'un sistema mòbil d'instrumentació i mapeig del terreny amb posicionament basat en un filtre de Kalman*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Grau en Enginyeria Electrònica de Telecomunicació)

This version is available at <https://ddd.uab.cat/record/336130>

under the terms of the  license.



TREBALL DE FI DE GRAU

GRAU EN ENGINYERIA ELECTRÒNICA DE TELECOMUNICACIÓ

IMPLEMENTACIÓ D'UN SISTEMA MÒBIL
D'INSTRUMENTACIÓ I MAPEIG DEL TERRENY AMB
POSICIONAMENT BASAT EN UN FILTRE DE
KALMAN

Lluna Juvillà Belmonte

DIRECTOR: Albert Crespo Yepes

DEPARTAMENT D'ENGINYERIA ELECTRÒNICA

UNIVERSITAT AUTÒNOMA DE BARCELONA

Bellaterra, Juny 2026

Agraïments

Vull començar agraint al meu tutor, l'Albert Crespo, per tot el seu suport, la seva dedicació i la confiança que m'ha transmès des del principi.

També vull expressar el meu agraïment al Javier Hellín per tota l'ajuda proporcionada.

De manera molt especial, vull donar les gràcies a la meva mare, per no haver dubtat mai de mi i per haver-me donat un suport incondicionalment sempre.

Finalment, vull agrair a tots els amics, companys i companyes del grau que m'han acompanyat al llarg d'aquesta etapa. Gràcies per tots els moments compartits, pel suport mutu i per haver fet aquest camí molt més enriquidor i lleuger.

A tots i totes, moltes gràcies.

Abstract

This Bachelor's Thesis addresses the design and implementation of a mobile robot prototype capable of estimating its own relative position and orientation without relying on any external localization or connectivity system, a scenario representative of planetary exploration, underground caves or indoor environments. The robot, based on an ESP32-S3 microcontroller, integrates an MPU6050 inertial measurement unit (IMU), wheel encoders, ultrasonic sensors for obstacle detection and a DHT11 temperature and humidity sensor. The core of the work is the design and implementation of a Kalman filter that fuses the IMU and encoder data to obtain a position and orientation estimate more stable and accurate than that provided by either source on its own. The system has been designed to offer several operating modes, namely manual guidance, automatic movement and return to the origin point. In addition, a user interface application has been designed and programmed, connected to the robot via WiFi, which allows the real-time visualization of the data, as well as the relative position of the vehicle and the detected obstacles.

Resum

Aquest Treball de Fi de Grau aborda el disseny i la implementació d'un prototip de robot mòbil capaç d'estimar la seva pròpia posició relativa i orientació sense dependre de cap sistema de posicionament de localització ni de connectivitat, un escenari representatiu de l'exploració planetària, de coves subterrànies o d'entorns interiors. El robot, basat en un microcontrolador ESP32-S3, integra una unitat de mesura inercial (IMU) MPU6050, encoders a les rodes, sensors d'ultrasons per a la detecció d'obstacles i un sensor de temperatura i humitat DHT11. El nucli del treball és el disseny i la implementació d'un filtre de Kalman que fusiona les dades de la IMU i dels encoders per obtenir una estimació de posició i orientació més estable i precisa que la que oferiria cadascuna de les fonts per separat. El sistema s'ha dissenyat per oferir diversos modes d'operació, com la guia manual, el moviment automàtic o el retorn al punt d'origen. A més, s'ha dissenyat i programat un aplicatiu a mode d'interfície d'usuari, connectat al robot via WiFi, on es poden visualitzar les dades en temps real, així com la posició relativa del vehicle i els obstacles detectats.

Índex

Agraïments	i
Abstract	ii
Resum	iii
1 Introducció	1
1.1 Motivació i objectius	1
1.2 Requisits	2
1.3 Organització del treball	5
2 Estat de l'art	6
2.1 Placa de desenvolupament emprada	6
2.1.1 Arduino Mega 2560	6
2.1.2 Raspberry Pi	7
2.1.3 ESP32-S3 N16R8	7
2.1.4 Comparativa i decisió	8
2.2 Estimació de posició sense sistema de localització	9
2.2.1 El filtre de Kalman	10
2.2.2 La unitat de mesura inercial (IMU)	12
2.2.3 Els encoders de roda	17
2.2.4 Fusió de la IMU i els encoders amb el filtre de Kalman	18
3 Disseny del sistema	21
3.1 Selecció de components	21
3.1.1 Unitat de control: ESP32-S3 N16R8	21
3.1.2 Unitat de mesura inercial: MPU6050	22
3.1.3 Encoders òptics de roda	22
3.1.4 Sensors d'ultrasons: HC-SR04	23

3.1.5	Sensor de temperatura i humitat: DHT11	24
3.1.6	Driver de motors: L298N	24
3.1.7	Motors i xassís	26
3.1.8	Alimentació	26
3.2	Esquema global	26
3.3	Disseny de la PCB	27
3.3.1	Esquemàtic	28
3.3.2	Assignació de pins	29
3.3.3	Disseny de la placa i enrutament	30
3.3.4	Migració a Altium i fabricació	32
4	Modes de moviment i exploració	35
4.1	Control dels motors	35
4.2	Manual	36
4.3	Automàtic	37
4.4	Retorn	38
5	Muntatge del prototip	40
5.1	Disposició dels components al xassís	40
5.2	Alimentació i bateries	41
5.3	Del protoboard a la PCB	41
6	Interfície d'usuari i recopilació de dades	45
6.1	Arquitectura de comunicació	45
6.2	Protocol de dades	46
6.3	Interfície d'usuari	48
7	Proves, correccions i validacions	55
7.1	Proves	55
7.2	Correccions	56
7.3	Validacions	58

8	Conclusions	64
9	Treball futur	66
	Referències	67

Índex de figures

1.1	Planificació temporal dels grans blocs de treball del projecte.	5
2.1	Esquema de pins de la placa ESP32-S3-N16R8. Font: [2].	8
2.2	Els tres angles d'orientació: yaw, pitch i roll. Font: [3].	13
2.3	Especificacions de l'acceleròmetre de la MPU6050 segons el datasheet oficial. El biaix inicial (<i>Initial Calibration Tolerance del Zero-G Output</i>) és de ± 50 mg per als eixos X i Y. Font: InvenSense, MPU-6000/MPU-6050 Product Specification. Font: [4]	16
3.1	Mòdul de la IMU MPU6050 (placa GY-521) amb l'orientació dels eixos de mesura de l'acceleròmetre i el giroscopi.	22
3.2	Mòdul d'encoder òptic de ranura basat en el comparador LM393. Font: [6].	23
3.3	Sensor d'ultrasons HC-SR04 i la seva distribució de pins (Vcc, Trig, Echo i GND). Font: [8].	23
3.4	Sensor de temperatura i humitat DHT11. Font: [9].	24
3.5	Mòdul driver de motors L298N i la seva distribució de pins i borns de connexió. Font: [11].	25
3.6	Motor de corrent continu amb reductora del tipus TT.	26
3.7	Esquema global del sistema al disseny inicial, amb l'ESP32-S3 com a unitat central de control.	27
3.8	Esquemàtic del sistema dissenyat en KiCad.	28
3.9	Esquema del divisor de tensió a la línia d'echo de cada sensor HC-SR04, amb $R_1 = 1 \text{ k}\Omega$ i $R_2 = 2 \text{ k}\Omega$	29
3.10	Disposició dels components a la PCB en KiCad, abans de l'enrutament. . .	30
3.11	Procés d'enrutament automàtic de la placa amb el connector FreeRouting.	31
3.12	Placa enrutada en KiCad després d'aplicar el FreeRouting. S'observen algunes pistes molt fines.	31
3.13	Primera versió de la PCB fabricada per JLCPCB, amb pistes que faltaven i problemes a la màscara de soldadura dels pads.	33
3.14	Versió final de la PCB fabricada per JLCPCB, amb els problemes resolts i llesta per soldar-hi els components. Es mostren les dues cares de la placa. .	34

4.1	Principi de la tracció diferencial en un gir: amb el costat esquerre endavant i el dret enrere, el robot gira cap a la dreta.	36
4.2	Diagrama de flux del mode manual: cada ordre rebuda s'associa a una primitiva de moviment.	36
4.3	Diagrama de flux del mode automàtic d'evitació d'obstacles.	37
4.4	Diagrama de flux del mode retorn cap al punt d'origen.	39
5.1	Disposició de components sobre el xassís: (a) el driver L298N a la planta inferior i (b) la IMU MPU6050 a la planta superior, alineada amb els eixos del robot.	41
5.2	Prototip en la fase inicial, muntat sobre protoboard i amb la brida de protecció frontal.	42
5.3	Preparació de la PCB i el xassís: (a) la placa soldada amb tots els sòcols i (b) la perforació del xassís per fixar-la.	43
5.4	Prototip final amb la PCB muntada sobre el robot.	44
6.1	Arquitectura de comunicació entre el robot i la interfície d'usuari.	45
6.2	Disseny previ de la interfície amb el mode manual seleccionat.	49
6.3	Disseny previ de la interfície amb el mode automàtic seleccionat.	49
6.4	Disseny previ de la interfície amb el mode retorn seleccionat.	50
6.5	Aspecte general de la interfície d'usuari, amb el robot desconnectat.	51
7.1	Estat general de la interfície amb el robot connectat i en repòs.	59
7.2	Orientació: model 3D i gràfica del filtre de Kalman.	60
7.3	Robot al punt de partida, amb el metre marcant la distància d'un metre a recórrer.	61
7.4	Mode manual: avanç d'un metre i posició al mapa.	61
7.5	Mode automàtic: detecció i esquiva d'un obstacle.	62
7.6	Mode retorn: tornada cap al punt d'origen.	63
7.7	Posició final del robot un cop completat el mode retorn.	63

Índex de taules

2.1	Comparativa de les tres plataformes considerades per al prototip.	9
3.1	Especificacions principals de la unitat de control ESP32-S3 (versió N16R8).	21
3.2	Especificacions principals de la IMU MPU6050.	22
3.3	Especificacions principals del sensor d'ultrasons HC-SR04.	23
3.4	Especificacions principals del sensor DHT11.	24
3.5	Especificacions principals del mòdul driver de motors L298N.	25
3.6	Assignació final dels pins de l'ESP32-S3 als components del sistema.	30
6.1	Estructura de la trama binària de telemetria (96 bytes, little-endian).	47
6.2	Ordres de control que la interfície envia al robot.	48

1. Introducció

L'exploració d'entorns on no es disposa d'infraestructura de navegació global, com ara altres planetes, coves subterrànies o instal·lacions interiors, planteja un repte tecnològic rellevant: com pot un vehicle autònom desplaçar-se, orientar-se i cartografiar el seu entorn sense el suport de cap sistema de posicionament de localització ni de connectivitat? Aquest Treball de Fi de Grau aborda aquesta qüestió mitjançant el disseny i la implementació d'un prototip de robot mòbil basat en la placa ESP32, equipat amb una unitat de mesura inercial (IMU), encoders a les rodes, sensors d'ultrasons i un sensor ambiental de temperatura i humitat.

El plantejament parteix de la premissa que, en un escenari extraterrestre o en qualsevol indret mancat d'una constel·lació de satèl·lits de posicionament, el vehicle ha d'estimar la seva trajectòria a partir exclusivament de sensors embarcats. Aquesta aproximació, coneguda com a navegació autònoma per estimació, consisteix a calcular la posició relativa respecte a un origen conegut a partir de mesures internes de moviment. La IMU proporciona acceleració i velocitat angular, mentre que els encoders aporten una mesura directa del desplaçament de cada roda. Com que cap d'aquestes fonts és perfecta, les lectures de la IMU presenten soroll i biaixos que, integrats al llarg del temps, donen lloc a errors acumulatius significatius, i els encoders pateixen errors per lliscament o irregularitats del terreny, en aquest treball s'implementa un filtre de Kalman que fusiona les dades dels diferents sensors per obtenir una estimació de posició i orientació més estable i precisa que la que oferiria qualsevol d'ells per separat. Paral·lelament, el robot recull dades de temperatura, humitat i distància a obstacles, que permeten generar una caracterització bàsica del terreny explorat.

1.1 Motivació i objectius

La motivació principal d'aquest treball neix de l'interès per l'exploració planetària i, més concretament, per les missions robòtiques que operen en entorns on la comunicació amb la Terra és limitada i on no es disposa de cap sistema de posicionament de localització. En aquests escenaris, el vehicle ha de ser prou autònom per prendre decisions locals de navegació, evitar obstacles, mantenir una estimació coherent de la seva posició i recopilar informació rellevant sobre l'entorn que travessa.

Paral·lelament, el projecte respon a l'interès personal per integrar en un únic sistema els coneixements adquirits al llarg del grau: disseny de circuits analògics i digitals, adquisició i tractament de senyal provinent de sensors, implementació d'algorismes de fusió sensorial (concretament el filtre de Kalman), programació de microcontroladors, comunicacions sense fils i disseny de plaques de circuit imprès (PCB). El resultat ha de ser un prototip

funcional que serveixi com a prova de concepte d'un vehicle explorador a petita escala, capaç d'instrumentar el terreny alhora que estima de manera robusta la seva pròpia posició i orientació.

A partir d'aquest context, els objectius específics que es persegueixen són els següents:

- Dissenyar un prototip de robot mòbil basat en l'ESP32 capaç d'operar sense dependència de cap sistema de posicionament de localització ni de connectivitat externa.
- Integrar una IMU MPU6050 mitjançant bus I2C.
- Integrar encoders a les rodes per estimar la velocitat de gir de les rodes.
- Implementar un filtre de Kalman que fusioni les dades de la IMU amb les dels encoders de roda per estimar l'orientació i la posició del robot amb un error inferior al de cada sensor per separat.
- Dotar el sistema d'instrumentació ambiental amb un sensor DHT11 de temperatura i humitat.
- Dotar el sistema d'instrumentació de sensors d'ultrasons per a la detecció d'obstacles i encoders a les rodes per al càlcul d'odometria.
- Implementar diversos modes d'operació: manual, automàtic i retorn al punt d'origen, que permetin avaluar el comportament del sistema en diverses situacions.
- Dissenyar una PCB pròpia que integri l'electrònica del prototip, juntament amb un xassís fabricat a mida, per facilitar el muntatge i les proves experimentals.
- Establir una comunicació sense fils via WiFi entre el robot i un servidor.
- Desenvolupar una interfície d'usuari que permeti controlar a distància el vehicle, visualitzar l'estat dels sensors i comparar el senyal en cru amb el senyal filtrat pel filtre de Kalman.
- Avaluar experimentalment la precisió de l'estimació d'orientació i de posició, identificant les principals fonts d'error i les limitacions de l'enfocament escollit.

1.2 Requisites

Els requisits del sistema s'han organitzat seguint la metodologia MoSCoW, que classifica les funcionalitats en quatre categories segons la seva prioritat: *Must have* (imprescindibles), *Should have* (importantes però no crítiques), *Could have* (desitjables) i *Won't have* (fora de l'abast del projecte). Aquesta classificació permet delimitar de manera clara el

nucli del treball i les extensions opcionals, facilitant la planificació i la presa de decisions davant de limitacions de temps o de recursos.

Must have

Aquests són els requisits imprescindibles que el sistema ha de complir per considerar-se funcional. Es divideixen en cinc blocs:

Programari i maquinari

- El sistema ha d'utilitzar una o diverses ESP32 com a controladors principals.
- El robot ha d'incorporar una IMU, concretament una MPU6050, llegida mitjançant comunicació I2C.
- El robot ha d'integrar encoders a les rodes per al càlcul d'odometria.
- El robot ha d'integrar sensors d'ultrasons HC-SR04 per a la detecció d'obstacles.
- El robot ha d'integrar un sensor ambiental DHT11 per a la mesura de temperatura i humitat.
- El robot ha de disposar d'un sistema de locomoció amb motors controlats mitjançant un driver L298N.

Adquisició de dades

- El sistema ha d'adquirir dades de tots els sensors de manera periòdica, sincronitzant l'adquisició amb una freqüència definida.
- El sistema ha de realitzar una calibració bàsica de la IMU abans de començar a operar.

Processament de dades

- El sistema ha d'implementar un filtre de Kalman que fusioni les dades de l'acceleròmetre, el giroscopi i els encoders per estimar l'orientació i la posició del robot.
- El sistema ha de filtrar el soroll de les mesures dels sensors.

Moviment i modes de funcionament

- El robot s'ha de poder desplaçar endavant, endarrere, a l'esquerra i a la dreta, i disposar d'una opció d'aturada segura.
- El robot ha de disposar d'un mode manual, controlat per l'usuari a través de la interfície.

- El robot ha de disposar d'un mode automàtic, amb desplaçament continu, detecció d'obstacles i canvi de direcció.
- El robot ha de disposar d'un mode retorn que li permeti tornar automàticament al punt d'origen a partir de l'estimació de posició pròpia.

Comunicació i interfície

- El robot ha de transmetre les dades al servidor mitjançant WiFi.
- El sistema ha de disposar d'una interfície d'usuari que processi i mostri les dades, incloent-hi la comparació entre el senyal en cru i el senyal filtrat pel filtre de Kalman.

Should have

Requisits importants que milloren significativament la qualitat del projecte, però que no són crítics per al seu funcionament bàsic:

- El sistema ha de guardar les dades dels sensors per permetre'n una anàlisi posterior.
- La interfície ha de mostrar l'estat del robot: mode actual, valors del giroscopi i de l'acceleròmetre, i orientació estimada.
- El robot ha de disposar d'un xassís fabricat a mida.

Could have

Funcionalitats desitjables que s'implementaran si el temps i els recursos ho permeten, i que obren la porta a ampliacions futures del sistema:

- El robot ha de realitzar un escaneig de l'entorn utilitzant un sensor d'ultrasons muntat sobre un servomotor.
- El robot ha de disposar d'un mode d'indicar coordenades, que li permeti desplaçar-se de manera autònoma cap a un punt de destí indicat per l'usuari respecte al punt d'origen.
- El sistema ha de representar les dades de l'entorn en un mapa visual.
- La interfície ha de mostrar un model 3D de l'orientació del robot a partir de les dades de la IMU.

1.3 Organització del treball

El desenvolupament d'aquest treball s'ha organitzat en una sèrie de grans blocs de feina que, tot i tenir un ordre lògic, s'han solapat parcialment en el temps. La figura 1.1 en mostra la planificació temporal aproximada.

El treball va començar amb una fase de recerca i documentació, dedicada a estudiar l'estat de l'art, el filtre de Kalman i els components disponibles. A partir d'aquí es va abordar el disseny del sistema, inclosa la selecció de components i el disseny de la PCB. Tot seguit es van desenvolupar en paral·lel les dues grans parts de programari: el backend que s'executa a l'ESP32-S3 (control de motors, sensors, modes de moviment i filtre de Kalman) i la interfície d'usuari. El muntatge del prototip, primer en protoboard i després sobre la PCB, es va anar fent a mesura que el maquinari estava disponible, i va donar pas a la fase de proves, correccions i validacions. La redacció de la memòria s'ha dut a terme de manera transversal al llarg de bona part del projecte.

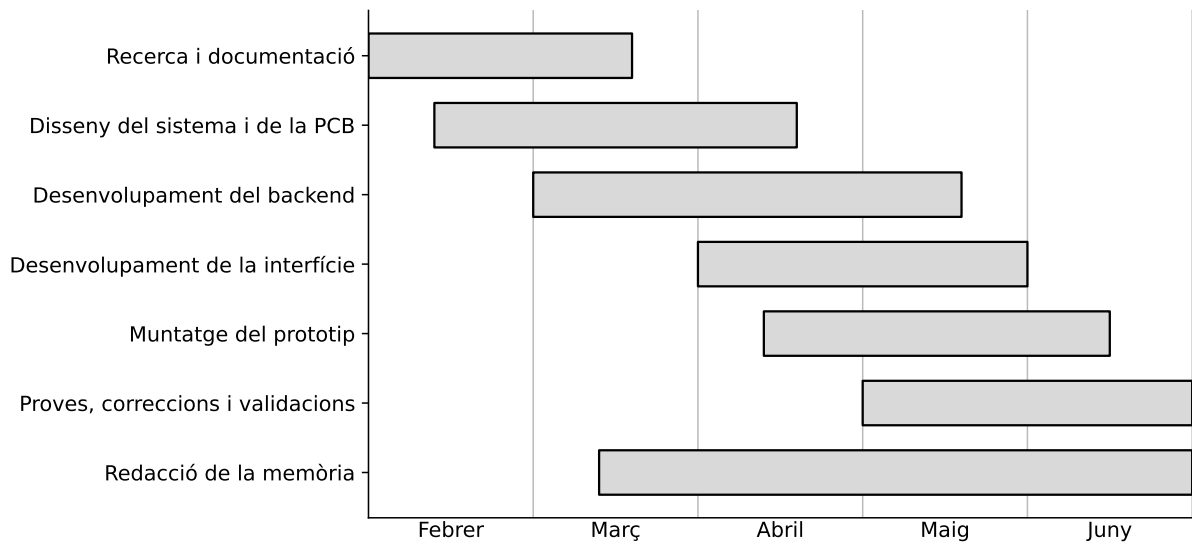


Figura 1.1: Planificació temporal dels grans blocs de treball del projecte.

Aquesta memòria s'estructura en nou capítols que segueixen aproximadament aquest mateix ordre. Després d'aquesta introducció, el capítol 2 presenta l'estat de l'art; el capítol 3, el disseny del sistema; el capítol 4, els modes de moviment; el capítol 5, el muntatge del prototip; el capítol 6, la interfície d'usuari i la recopilació de dades; i el capítol 7, les proves, correccions i validacions. Finalment, els capítols 8 i 9 recullen les conclusions i el treball futur.

2. Estat de l'art

En aquest capítol es fa una revisió de les tecnologies i les tècniques existents que han orientat el desenvolupament del prototip. En primer lloc, es presenten les principals plataformes de desenvolupament que es van considerar per a la unitat de control, justificant l'elecció de la placa finalment emprada. Tot seguit, s'analitzen les tècniques de moviment i d'estimació de la posició sense posicionament de localització ni connectivitat externa més habituals: el funcionament d'una unitat de mesura inercial (IMU), l'ús d'encoders de roda per al càlcul d'odometria i la fusió de totes aquestes fonts de dades mitjançant un filtre de Kalman.

2.1 Placa de desenvolupament emprada

El cervell del prototip és la placa de desenvolupament, encarregada de llegir els sensors, d'executar l'algorisme de fusió sensorial, de controlar els motors i de comunicar-se amb el servidor. A l'hora de triar-la es van valorar tres famílies de plataformes habituals en projectes de robòtica: Arduino, Raspberry Pi o ESP32.

2.1.1 Arduino Mega 2560

La família Arduino s'ha consolidat com la plataforma d'entrada més estesa en l'àmbit de l'electrònica educativa i de la creació d'aficionats. La seva gran virtut és la simplicitat: un entorn de desenvolupament molt accessible, una comunitat activa, i una gran quantitat de llibreries i projectes documentats. En particular, l'Arduino Mega 2560 es construeix al voltant del microcontrolador ATmega2560 de 8 bits, ofereix 54 pins digitals i és especialment adequat per a projectes que requereixen molts sensors i actuadors però poca potència de càlcul.

Tanmateix, aquesta placa presenta limitacions importants per al projecte. D'una banda, la seva freqüència de rellotge de 16 MHz i els només 8 KB de SRAM fan que l'execució en temps real d'un filtre de Kalman amb múltiples estats, que ha de fusionar simultàniament dades de la IMU i dels encoders, i la gestió simultània de diversos sensors sigui poc eficient. D'altra banda, i més rellevant en aquest cas, l'Arduino Mega no inclou connectivitat WiFi ni Bluetooth de manera nativa. Per poder transmetre dades de manera sense fils caldria incorporar un mòdul extern, la qual cosa augmenta la complexitat, el consum i el cost del conjunt.

2.1.2 Raspberry Pi

En l'extrem oposat trobem la família Raspberry Pi, que no és un microcontrolador sinó un ordinador d'una sola placa (SBC). Models com la Raspberry Pi 4 incorporen un processador ARM Cortex-A72 de 64 bits, 4 GB de memòria RAM en la versió més habitual, WiFi, Bluetooth, Ethernet, USB, HDMI i un sistema operatiu complet basat en Linux. Això li permet executar aplicacions complexes, interfícies gràfiques, servidors web o algorismes de visió per computador sense gaires restriccions.

Aquest potencial té un cost. Comparada amb un microcontrolador, la Raspberry Pi té un consum elèctric molt superior, típicament d'entre 600 mA i 1200 mA, davant dels 30 a 160 mA d'un ESP32. També presenta unes dimensions més grans i un preu més elevat. A més, l'arrencada d'un sistema operatiu i la falta de determinisme estricte en les interrupcions la fa menys adequada per a tasques de control en temps real, com la lectura periòdica d'una IMU o el comptatge de polsos d'encoders a freqüències elevades. Per a un robot petit alimentat per bateria i centrat en l'adquisició de sensors, la Raspberry Pi resulta sobredimensionada.

2.1.3 ESP32-S3 N16R8

L'ESP32-S3 forma part de la família de microcontroladors d'Espressif Systems, una plataforma molt utilitzada en projectes d'electrònica connectada i de l'Internet de les Coses (IoT) per la seva combinació de potència de càlcul, connectivitat sense fils integrada i cost reduït. És un microcontrolador de doble nucli de 32 bits que funciona fins a 240 MHz, amb 512 KB de SRAM interna i suport per a memòria PSRAM i Flash externes [1].

La característica més rellevant per al projecte és que incorpora WiFi i Bluetooth de manera nativa, cosa que permet comunicar el robot amb un servidor extern sense haver d'afegir mòduls addicionals. La placa concreta utilitzada és la versió N16R8, que afegeix 16 MB de Flash i 8 MB de PSRAM al mòdul base. Aquesta capacitat de memòria és més que suficient per emmagatzemar el programa, les llibreries dels sensors i els buffers de dades necessaris durant l'execució del filtre de Kalman.

A la figura 2.1 es pot veure l'esquema de pins de la placa, que ha estat la referència emprada per assignar els GPIOs als sensors i actuadors durant el disseny de la PCB.

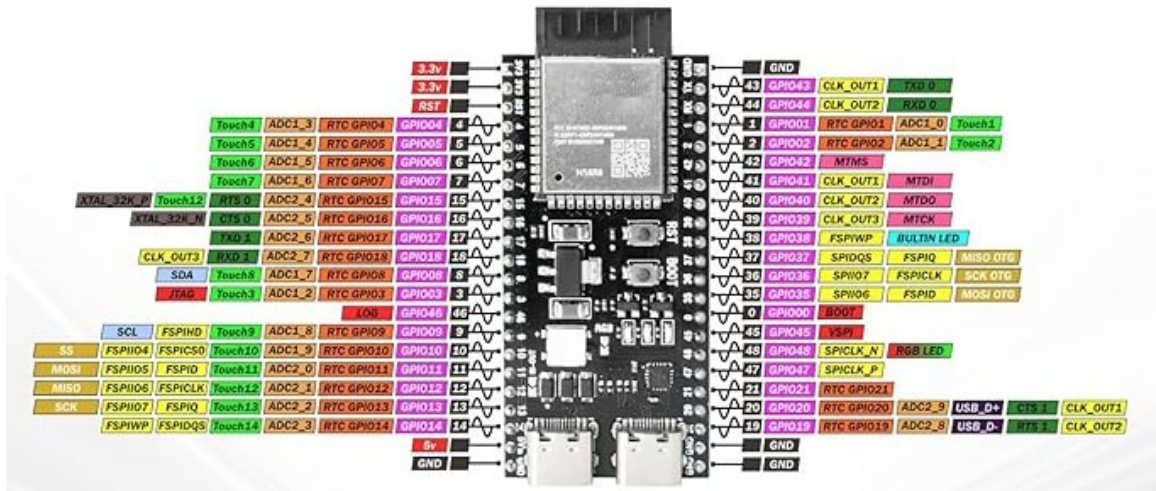


Figura 2.1: Esquema de pins de la placa ESP32-S3-N16R8. Font: [2].

Aquestes característiques fan l'ESP32-S3-N16R8 idoni per a un robot com el del present treball, per diversos motius.

- Disposa de potència de càlcul i de memòria suficients per executar el filtre de Kalman fusionant dades de la IMU i dels encoders, i per gestionar simultàniament la lectura dels ultrasons i del sensor DHT11 sense saturar-se.
- La connectivitat WiFi integrada permet enviar dades al servidor i rebre ordres des de la interfície d'usuari sense necessitat de maquinari addicional.
- El doble nucli permet repartir tasques crítiques. Per exemple, executar la fusió sensorial i el control en un nucli, i la comunicació WiFi en l'altre, garantint un comportament determinista en temps real.
- El seu consum és molt inferior al d'una Raspberry Pi, un factor crític en un vehicle alimentat per bateria.
- El seu cost, entorn dels 10 a 18 €, és significativament més baix que el d'un Arduino Mega amb mòdul WiFi o el d'una Raspberry Pi.

2.1.4 Comparativa i decisió

La taula 2.1 resumeix les característiques principals de les tres plataformes considerades.

Característica	Arduino Mega 2560	Raspberry Pi 4	ESP32-S3 N16R8
Tipus	Microcontrolador	Ordinador SBC	Microcontrolador
Processador	ATmega2560 8 bits, 16 MHz	ARM Cortex-A72 64 bits, 1,5 GHz	Xtensa LX7 32 bits doble nucli, fins a 240 MHz
Memòria	8 KB SRAM, 256 KB Flash	4 GB RAM	512 KB SRAM, 8 MB PSRAM, 16 MB Flash
WiFi / Bluetooth	No (requereix mòdul addicional)	Sí, integrats	Sí, WiFi i Bluetooth
GPIO	54 pins digitals	40 pins GPIO	45 pins programables
Sistema operatiu	Bare metal	Linux (Raspberry Pi OS)	Bare metal o FreeRTOS
Consum típic	90 mA	600 – 1200 mA	30 – 160 mA
Preu orientatiu	25 – 40 €	60 – 90 €	10 – 18 €

Taula 2.1: Comparativa de les tres plataformes considerades per al prototip.

Tenint en compte els requisits del sistema, és a dir, l'execució en temps real d'un filtre de Kalman que fusiona dades de la IMU i els encoders, la lectura periòdica de diversos sensors, la comunicació WiFi amb un servidor, l'alimentació per bateria i un pressupost ajustat, l'ESP32-S3-N16R8 és la plataforma que millor equilibra potència de càlcul, memòria, connectivitat, consum i cost. Aquests són els motius pels quals s'ha escollit com a unitat de control del prototip, descartant tant l'Arduino Mega, per la seva manca de WiFi nativa i la seva limitació de potència, com la Raspberry Pi, per ser una solució sobredimensionada per al cas d'ús.

2.2 Estimació de posició sense sistema de localització

Quan un vehicle autònom opera en un entorn on no hi ha un sistema global de posicionament disponible, com ara altres planetes, interiors d'edificis, túnels o coves, l'estimació de la posició i l'orientació s'ha de basar exclusivament en sensors embarcats. Aquesta tècnica, coneguda com a estimació de posició per integració sensorial, consisteix a calcular la posició actual sumant a una posició coneguda els increments de desplaçament i d'orientació estimats a partir dels sensors.

El problema fonamental d'aquesta aproximació és que qualsevol error en els sensors s'acumula al llarg del temps, de manera que l'error total creix de forma acumulativa.

En la pràctica, aquest creixement pot fer que l'estimació de posició esdevingui errònea després de pocs metres de recorregut si no es prenen mesures per mitigar aquest error. La manera més efectiva de fer-hi front és utilitzar múltiples fonts d'informació amb errors de naturalesa diferent i fusionar-les de manera que es pugui estimar la posició relativa de forma més fiable. En aquest projecte, l'estimació es basa en la combinació de dues fonts complementàries, una unitat de mesura inercial (IMU) i un parell d'encoders a les rodes, mitjançant un filtre de Kalman.

2.2.1 El filtre de Kalman

El filtre de Kalman és un algorisme recursiu d'estimació publicat per Rudolf Emil Kalman l'any 1960, que permet estimar l'estat d'un sistema dinàmic a partir d'una sèrie de mesures sorolloses. Sota les hipòtesis de linealitat del sistema i de soroll gaussià, és l'estimador de mínima variància, és a dir, no existeix cap altre estimador lineal que ofereixi una estimació amb menys error quadràtic mitjà.

La idea fonamental del filtre és que cap sensor és perfecte i cap model matemàtic descriu el món amb total exactitud, però si tenim una predicció basada en un model i una mesura presa amb un sensor, podem combinar-les de forma ponderada per obtenir una estimació millor que cadascuna per separat. El pes que es dona a cada font es decideix dinàmicament segons la fiabilitat de cadascuna en cada instant.

El filtre representa el coneixement sobre el sistema mitjançant dues magnituds clau a cada instant. La primera és el vector d'estat estimat $\hat{x}$, que conté les variables que es volen estimar, com ara la posició i l'orientació del robot. La segona és la matriu de covariància P , que quantifica la incertesa associada a aquesta estimació. El funcionament és iteratiu i a cada pas de temps s'executen dues etapes.

A les equacions que segueixen s'utilitza la notació següent. El subíndex k representa l'instant de temps actual, mentre que $k - 1$ correspon a l'instant anterior. L'expressió $\hat{x}_{k|k-1}$ indica l'estimació de l'estat a l'instant k feta amb la informació disponible fins a l'instant $k - 1$, és a dir, la predicció, i $\hat{x}_{k|k}$ indica l'estimació a l'instant k després d'haver incorporat la mesura presa en aquest mateix instant, és a dir, la correcció. La mateixa notació s'aplica a la matriu de covariància P i al guany K_k .

Etape de predicció

A la primera etapa, el filtre fa una predicció de quin serà el següent estat del sistema basant-se únicament en el coneixement previ, és a dir, en l'estat estimat a l'instant anterior i en el model que descriu com aquest estat evoluciona amb el temps. En el cas del robot, aquesta predicció respondria a una pregunta com ara “si fa un instant el robot estava aquí i sé com se suposa que es mou, on hauria d'estar ara?”.

Aquest model es representa matemàticament amb dues matrius. La matriu F , anomenada matriu de transició, descriu com l'estat passa d'un instant al següent en absència de qualsevol acció externa. La matriu B descriu com una acció de control, representada pel vector u , modifica aquest estat. Per exemple, en el cas del robot, u podria contenir les ordres de velocitat enviades als motors. A partir de l'estat anterior i de les matrius F i B , l'estat predit s'obté amb l'equació (2.1), i la incertesa associada amb l'equació (2.2):

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \quad (2.1)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (2.2)$$

Si no hi ha cap entrada de control coneguda, el terme $B_k u_k$ de l'equació (2.1) és zero. A l'equació (2.2), la matriu Q és la covariància del soroll del procés, que modela la incertesa intrínseca del model.

Etapas de correcció

Un cop disposem de la predicció a priori ($\hat{x}_{k|k-1}$, $P_{k|k-1}$) i arriba una nova mesura z_k , el filtre incorpora aquesta informació mitjançant l'etapa de correcció, també anomenada etapa d'actualització. L'objectiu és obtenir l'estimació a posteriori ($\hat{x}_{k|k}$, $P_{k|k}$) que minimitza l'error quadràtic mitjà condicionat a totes les mesures rebudes fins a l'instant k .

La relació entre l'estat i la mesura es modela amb la matriu d'observació H , que projecta l'estat a l'espai de mesura, i amb la covariància del soroll de mesura R , que caracteritza estadísticament la incertesa associada al sensor. Sota la hipòtesi de soroll gaussià, blanc i no correlacionat amb el soroll del procés, les equacions de correcció són les següents:

$$K_k = P_{k|k-1} H^T (H P_{k|k-1} H^T + R)^{-1} \quad (2.3)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H \hat{x}_{k|k-1}) \quad (2.4)$$

$$P_{k|k} = (I - K_k H) P_{k|k-1} \quad (2.5)$$

L'equació (2.3) defineix la matriu K_k , anomenada guany de Kalman, que actua com a ponderador òptim entre la predicció a priori i la innovació de la mesura. Es deriva de la minimització de la traça de la matriu de covariància a posteriori $P_{k|k}$, que equival a minimitzar l'error quadràtic mitjà de l'estimació. El factor $H P_{k|k-1} H^T$ representa la incertesa de la predicció projectada a l'espai de mesura, i quan se li suma R s'obté

la incertesa total associada a la innovació $z_k - H\hat{x}_{k|k-1}$. La inversa d'aquesta suma normalitza la contribució relativa de cada font d'informació.

D'aquesta formulació es desprenen dos comportaments asimptòtics rellevants. Quan la covariància del soroll de mesura R és molt gran respecte a la incertesa de la predicció, el guany K_k tendeix a zero i la correcció és pràcticament nul·la, de manera que el filtre confia gairebé exclusivament en el model. En canvi, quan R és petita en comparació amb la incertesa de la predicció, K_k tendeix a H^{-1} i la mesura imposa la nova estimació amb molt poc pes per a la dinàmica modelitzada. Aquesta capacitat d'ajust automàtic, derivada directament de la propagació estadística de la incertesa, és el que distingeix el filtre de Kalman de mètodes de fusió heurístics com el filtre complementari, on els pesos s'han de fixar manualment.

L'equació (2.4) és la pròpiament dita actualització de l'estat. El terme $z_k - H\hat{x}_{k|k-1}$ s'anomena innovació o residu i representa la discrepància entre la mesura observada i la mesura que s'esperaria a partir de la predicció. L'estat es corregeix afegint-hi aquesta innovació ponderada per K_k . Finalment, l'equació (2.5) actualitza la matriu de covariància, reduint la incertesa associada a l'estimació en la proporció en què la mesura ha aportat informació nova.

Sota les hipòtesis del model lineal gaussià, es pot demostrar que el filtre de Kalman és l'estimador de mínima variància entre tots els estimadors lineals i, alhora, l'estimador màxim a posteriori, de manera que constitueix una solució òptima en sentit estadístic.

El filtre de Kalman s'ha consolidat com l'algorisme estàndard de fusió i estimació en una gran varietat d'aplicacions de l'enginyeria, entre les quals destaquen els sistemes de navegació inercial, el control d'actitud de vehicles aeris no tripulats, la localització de robots mòbils o el seguiment d'objectius en sistemes radar. La seva força resideix en el tractament rigorós de la incertesa, que permet combinar fonts d'informació amb estadístiques d'error heterogènies i obtenir una estimació conjunta amb mínima variància. Per aquest motiu és l'algorisme escollit en aquest treball per a la fusió de les mesures de la IMU i dels encoders, tal com es desenvoluparà més endavant.

2.2.2 La unitat de mesura inercial (IMU)

Una unitat de mesura inercial (Inertial Measurement Unit, IMU) és un dispositiu que combina un acceleròmetre i un giroscopi, i en alguns casos també un magnetòmetre, per mesurar el moviment d'un cos en l'espai. En aquest projecte s'utilitza la MPU6050, una IMU de sis graus de llibertat molt estesa en projectes d'electrònica, que combina en un mateix encapsulat un acceleròmetre de tres eixos i un giroscopi de tres eixos, i es comunica amb el microcontrolador a través del bus I2C. La MPU6050 és un sensor de tipus MEMS (Micro-Electro-Mechanical Systems), una tecnologia que integra elements

mecànics microscòpics i circuits electrònics sobre un xip de silici fabricat amb processos similars als dels semiconductors. Aquesta tecnologia ha permès reduir dràsticament la mida i el cost dels sensors inercials, tot i que això ha comportat un soroll i un biaix més elevats que els dels sensors inercials industrials clàssics. Pràcticament totes les IMUs presents en mòbils, drons i robots de baix cost són MEMS.

Abans de continuar, convé recordar els tres angles que descriuen l'orientació d'un cos a l'espai, coneguts com a yaw, pitch i roll. El **yaw** és la rotació sobre l'eix vertical i indica cap a on apunta el robot al pla horitzontal, és a dir, el gir cap a la dreta o cap a l'esquerra. El **pitch** és la rotació sobre l'eix transversal i correspon a la inclinació cap endavant i cap enrere. El **roll** és la rotació sobre l'eix longitudinal i correspon a la inclinació lateral, cap a un costat o cap a l'altre. La figura 2.2 il·lustra aquests tres angles.

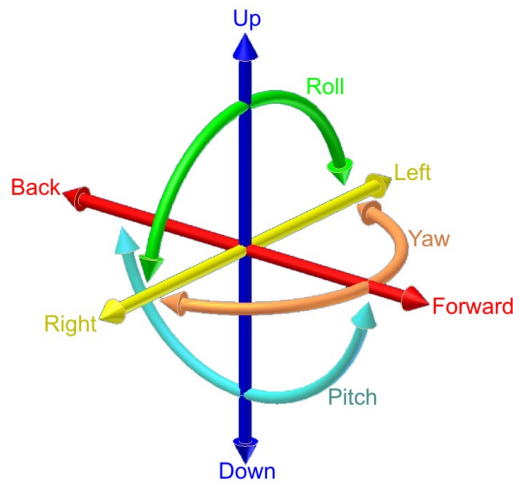


Figura 2.2: Els tres angles d'orientació: yaw, pitch i roll. Font: [3].

Acceleròmetre

L'acceleròmetre és un sensor que mesura l'acceleració lineal a què està sotmès al llarg dels seus tres eixos. Una propietat fonamental d'aquest tipus de sensor és que sempre detecta la component de la gravetat terrestre, que té una magnitud aproximada de $g = 9,81 \text{ m/s}^2$ i apunta sempre cap al centre de la Terra. Aquesta característica permet utilitzar-lo per estimar l'orientació del sensor respecte a la vertical.

Quan el sensor canvia la seva orientació, el vector de gravetat es projecta de manera diferent sobre cadascun dels seus eixos. Mesurant les tres components de l'acceleració, anomenades a_x , a_y i a_z , es poden recuperar els angles de roll (φ) i pitch (θ) aplicant les relacions trigonomètriques següents:

$$\varphi = \text{atan2}(a_y, a_z) \quad (2.6)$$

$$\theta = \text{atan2}\left(-a_x, \sqrt{a_y^2 + a_z^2}\right) \quad (2.7)$$

on a_x , a_y i a_z són les tres components de l'acceleració mesurades pels tres eixos del sensor. La suma vectorial d'aquestes tres components és igual al vector de gravetat, i és precisament aquest fet el que permet deduir-ne l'angle d'inclinació. La funció $\text{atan2}(y, x)$ és la versió de l'arctangent que pren els dos arguments per separat per determinar correctament el quadrant del resultat, garantint un rang de sortida complet de $[-\pi, \pi]$.

El seu gran avantatge és que en condicions estàtiques la gravetat actua com una referència física estable, ja que en un entorn local la seva magnitud i direcció es poden considerar constants. Això fa que l'orientació calculada amb l'acceleròmetre no acumuli error amb el temps i pugui servir per corregir altres sensors que sí pateixen deriva. El seu inconvenient principal és que les lectures són molt sorolloses i només són fiables quan el sensor està pràcticament aturat. Qualsevol acceleració externa, com una vibració, una frenada o un canvi de direcció, es barreja amb la component de la gravetat i distorsiona el càlcul de l'angle. Per aquest motiu, l'acceleròmetre és precís a llarg termini però poc fiable en dinàmica.

Giroscopi

El giroscopi mesura la velocitat angular $\omega(t)$ al voltant de cadascun dels tres eixos del sensor, és a dir, la rapidesa amb què el sensor està girant en cada instant. L'angle girat s'obté integrant aquesta velocitat al llarg del temps mitjançant la relació

$$\theta(t) = \theta_0 + \int_0^t \omega(\tau) d\tau \quad (2.8)$$

on θ_0 és l'angle inicial conegut. Aquesta integració proporciona una estimació d'orientació molt suau i precisa a curt termini, robusta davant de vibracions i acceleracions externes.

El seu inconvenient principal és la deriva. Les lectures en repòs no són exactament zero, sinó que presenten un petit biaix constant que, en integrar-se, fa créixer l'error d'angle linealment amb el temps. La magnitud d'aquesta deriva i la seva comparació amb el cas de l'acceleròmetre es desenvolupen amb més detall a l'apartat següent.

Així doncs, acceleròmetre i giroscopi presenten virtuts complementàries. L'un és fiable a llarg termini però sorollós a curt, i l'altre és precís a curt termini però deriva a llarg. Aquesta complementarietat és precisament la que el filtre de Kalman aprofita per estimar l'orientació del robot de manera estable.

El problema de la doble integració

Una qüestió fonamental que cal entendre és per què la IMU per si sola no n'hi ha prou per estimar la posició del robot, i per què l'estimació d'orientació basada en el giroscopi també acaba degradant-se amb el temps. Tots dos problemes es deriven del mateix fet: els sensors inercials no mesuren l'angle ni la posició directament, sinó magnituds derivades (velocitat angular i acceleració) que cal integrar respecte al temps. I qualsevol biaix en la mesura, per petit que sigui, es propaga i s'amplifica amb cada integració.

Per al giroscopi, el sensor no mesura la velocitat angular real $\omega(t)$, sinó una versió contaminada per un biaix b_g i per soroll:

$$\omega_{\text{mesurada}}(t) = \omega(t) + b_g + \eta_g(t) \quad (2.9)$$

En integrar aquesta lectura una vegada per obtenir l'angle, el biaix es transforma en un terme $b_g \cdot t$ que creix linealment amb el temps:

$$\theta_{\text{error}}(t) \approx b_g \cdot t \quad (2.10)$$

Aquest creixement lineal és l'origen de la deriva del giroscopi mencionada anteriorment. És un error gestionable a curt termini, però que acaba allunyant l'estimació del valor real al cap de pocs segons o minuts si no es corregeix.

Per a l'acceleròmetre el problema és més greu, perquè per obtenir la posició cal integrar dues vegades. Si denotem l'acceleració mesurada com $a_{\text{mesurada}}(t)$, la velocitat $v(t)$ i la posició $x(t)$, les relacions són:

$$v(t) = v_0 + \int_0^t a_{\text{mesurada}}(\tau) d\tau \quad (2.11)$$

$$x(t) = x_0 + v_0 \cdot t + \int_0^t \int_0^\tau a_{\text{mesurada}}(s) ds d\tau \quad (2.12)$$

A la pràctica l'acceleròmetre també mesura una versió contaminada per biaix i soroll:

$$a_{\text{mesurada}}(t) = a(t) + b_a + \eta_a(t) \quad (2.13)$$

Quan integrem una vegada per obtenir la velocitat, el biaix b_a contribueix amb un terme $b_a \cdot t$ que creix linealment, igual que passava amb el giroscopi. Però quan tornem a integrar per obtenir la posició, aquest mateix biaix contribueix amb un terme $\frac{1}{2}b_a \cdot t^2$ que creix quadràticament amb el temps:

$$x_{\text{error}}(t) \approx \frac{1}{2} b_a \cdot t^2 + \text{termes de soroll integrat} \quad (2.14)$$

Per fer-nos una idea de la magnitud del problema, es pot consultar el biaix típic de l'acceleròmetre de la MPU6050 directament al seu datasheet. El fabricant InvenSense especifica, a la secció 6.2 del MPU-6000/MPU-6050 Product Specification [4], una *Initial Calibration Tolerance* del *Zero-G Output* de ± 50 mg per als eixos X i Y, i de ± 80 mg per a l'eix Z, tal com es mostra a la figura 2.3.

6.2 Accelerometer Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V $\pm$ 5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0		±2		g	
	AFS_SEL=1		±4		g	
	AFS_SEL=2		±8		g	
	AFS_SEL=3		±16		g	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/g	
	AFS_SEL=1		8,192		LSB/g	
	AFS_SEL=2		4,096		LSB/g	
	AFS_SEL=3		2,048		LSB/g	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
ZERO-G OUTPUT						
Initial Calibration Tolerance	X and Y axes		±50		mg	1
	Z axis		±80		mg	
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C		±35			
	Z axis, 0°C to +70°C		±60		mg	
SELF TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	2
NOISE PERFORMANCE						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		µg/√Hz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		260	Hz	
OUTPUT DATA RATE						
	Programmable Range	4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT						
			32		mg/LSB	

Figura 2.3: Especificacions de l'acceleròmetre de la MPU6050 segons el datasheet oficial. El biaix inicial (*Initial Calibration Tolerance* del *Zero-G Output*) és de ± 50 mg per als eixos X i Y. Font: InvenSense, MPU-6000/MPU-6050 Product Specification. Font: [4]

Aquest valor de ± 50 mg correspon, en unitats del Sistema Internacional, a aproximadament $0,5 \text{ m/s}^2$. Aplicant la fórmula anterior, després de només 10 segons d'integració l'error de posició acumulat és de:

$$x_{\text{error}}(10) \approx \frac{1}{2} \cdot 0,5 \cdot 10^2 = 25 \text{ m} \quad (2.15)$$

La diferència entre els dos casos és reveladora. Un mateix biaix dona un error de l'ordre de pocs graus després d'uns segons quan només cal integrar una vegada, però arriba a desenes

de metres després del mateix interval quan cal integrar dues vegades. Aquesta asimetria, és a dir, orientació viable amb una integració però posició inviable amb dues, és la que explica per què l'orientació del robot es pot obtenir raonablement a partir del giroscopi (amb correccions periòdiques de l'acceleròmetre), però per a la posició cal recórrer a una font alternativa de mesura. Els encoders de roda, que es presenten a la secció següent, ofereixen precisament aquesta alternativa.

2.2.3 Els encoders de roda

Els encoders de roda són sensors acoblats a l'eix d'una roda que converteixen la rotació mecànica en polsos elèctrics. A diferència de l'acceleròmetre, els encoders proporcionen una mesura directa del desplaçament, sense necessitat de cap integració. Si una roda de radi r té un encoder que genera N polsos per volta, el desplaçament lineal corresponent a n polsos és simplement:

$$\Delta x = \frac{2\pi r}{N} \cdot n \quad (2.16)$$

A partir del desplaçament de cada roda i de la distància entre rodes es pot calcular el desplaçament i el gir del robot, com es veurà més endavant en aquesta mateixa secció. Aquesta tècnica és la base de la localització en multitud de plataformes robòtiques comercials i educatives.

L'odometria té diversos avantatges rellevants respecte a una IMU. En primer lloc, com s'ha vist a la secció anterior, evita el problema de la doble integració, de manera que els errors no creixen quadràticament. En segon lloc, en condicions estàtiques els encoders no presenten deriva. Si el robot no es mou, no es comptabilitzen polsos i la posició estimada queda congelada, cosa que no passa amb un giroscopi pur, que continua acumulant error fins i tot en repòs.

Tanmateix, l'odometria també té limitacions. Els errors d'odometria se solen classificar en dues categories. Els errors sistemàtics són els deguts a imperfeccions mecàniques constants, com ara rodes de diàmetre lleugerament desigual, una distància entre rodes mal modelada o desalineacions del xassís. Aquests errors es poden mitigar mitjançant calibratge. Els errors no sistemàtics, en canvi, són els deguts a fenòmens imprevisibles del terreny, com ara lliscaments de les rodes, irregularitats del paviment o petits xocs. Aquests errors són més difícils de corregir, ja que en cas de lliscament la roda gira sense que el robot avanci la distància corresponent, i l'encoder seguirà comptant polsos que no es tradueixen en un desplaçament real. Aquests errors també s'acumulen al llarg del recorregut [5].

En el prototip d'aquest treball s'utilitzen dos encoders òptics, un acoblat a la roda esquerra davantera i l'altre a la roda dreta davantera. La configuració del robot és diferencial, és a dir, els dos motors d'un mateix costat es controlen conjuntament i giren sincronitzats, de manera que un únic encoder per costat n'hi ha prou per estimar el desplaçament de tot el costat. Aquesta configuració és la més habitual en robots mòbils petits i ofereix una bona relació entre precisió, complexitat i cost.

A partir de la lectura dels dos encoders, en cada interval de temps Δt es calcula el desplaçament de cada costat, anomenats Δs_e i Δs_d per al costat esquerre i dret respectivament, i es deriven les magnituds globals del robot:

$$\Delta s = \frac{\Delta s_e + \Delta s_d}{2} \quad (2.17)$$

$$\Delta \theta = \frac{\Delta s_d - \Delta s_e}{b} \quad (2.18)$$

on Δs és el desplaçament lineal del centre del robot, $\Delta \theta$ és la variació de l'angle d'orientació i b és la distància entre les rodes esquerra i dreta. Combinant aquestes magnituds amb l'orientació estimada pel giroscopi de la IMU es poden actualitzar les coordenades x i y del robot al pla.

Comparant les dues fonts d'informació es fa evident la seva complementarietat. La IMU és sensible a vibracions i el seu giroscopi deriva amb el temps, però no es veu afectada per lliscaments ni per irregularitats puntuals del terreny. Els encoders, per la seva banda, són immunes a vibracions i ofereixen una mesura directa del moviment de les rodes, però fallen quan aquestes lliscen o quan el seu radi efectiu canvia. Per exemple, quan el vehicle pren un revolt amb una roda lliscant, els encoders donaran una lectura errònia mentre que el giroscopi mesurarà correctament la rotació. En canvi, quan el vehicle es mou suaument en línia recta, els encoders donaran una mesura molt més precisa que la doble integració de l'acceleròmetre. Per això, a la pràctica, l'estratègia que ofereix millors resultats és fusionar les dues fonts mitjançant el filtre de Kalman, tal com es presenta a la secció següent.

2.2.4 Fusió de la IMU i els encoders amb el filtre de Kalman

Un cop presentats el filtre de Kalman, la unitat de mesura inercial i els encoders de roda, en aquesta secció es descriu com es combinen els tres elements per obtenir una estimació robusta de l'estat del robot. La idea clau és que cap dels dos sensors és suficient per si sol, però quan es fusionen amb el filtre de Kalman cadascun aporta el que millor sap fer i compensa les debilitats de l'altre.

L'estat que volem estimar a cada instant està format per la posició del robot al pla i per la seva orientació, és a dir, x , y i θ . La predicció d'aquest estat es construeix a partir de les mesures dels encoders, que aporten el desplaçament lineal Δs i la variació d'orientació $\Delta\theta$ calculats amb les equacions de l'apartat anterior, i a partir de la velocitat angular del giroscopi, que ofereix una mesura suau i fiable a curt termini de com gira el robot. Amb aquestes dades el filtre projecta cap al següent instant l'estat anterior i obté una estimació prèvia.

Tot seguit, en l'etapa de correcció, el filtre incorpora la informació de l'acceleròmetre. L'acceleròmetre, recordem, no és útil per estimar la posició a causa del problema de la doble integració, però sí que dona una mesura absoluta de la inclinació del robot respecte de la vertical, és a dir, del pitch i del roll, ja que en repòs detecta directament la gravetat. Cal tenir present, però, una limitació important d'una IMU de sis eixos com la MPU6050: en girar el robot sobre l'eix vertical, la projecció de la gravetat sobre els eixos del sensor no canvia, de manera que l'acceleròmetre no pot observar ni corregir la deriva del yaw, que és precisament l'angle que determina cap a on es desplaça el robot al pla. Per aquest motiu, en aquest projecte la referència que corregeix la deriva del giroscopi en el yaw no és l'acceleròmetre sinó els encoders, que aporten una mesura directa del gir a partir del desplaçament de cada roda. L'acceleròmetre queda reservat com a referència absoluta de la inclinació. Aquesta distinció es desenvolupa amb més detall al capítol 6.

El resultat d'aquesta fusió és una estimació de l'estat amb propietats que cap dels sensors no aconseguiria per si sol. Els encoders aporten una mesura directa del desplaçament del robot que evita l'acumulació quadràtica d'error de la doble integració i, alhora, una referència del gir que permet corregir la deriva del giroscopi en el yaw. El giroscopi proporciona una orientació suau i precisa a curt termini, immune a les vibracions. Quan una roda llisca, la informació del giroscopi permet al filtre detectar que el robot no està girant tant com indicarien els encoders i ponderar les mesures en conseqüència. De manera anàloga, quan la IMU rep vibracions intenses, els encoders proporcionen una referència estable que evita estimacions erràtiques.

La matriu del guany de Kalman és precisament l'encarregada de gestionar aquest equilibri de manera dinàmica. En funció de la covariància assignada a cada sensor, el filtre confia més o menys en cada font en cada instant. Si els encoders són molt fiables en condicions normals, la seva covariància R serà petita i el guany corresponent serà gran, de manera que la seva mesura tindrà força pes en la correcció. Si en una situació concreta es detecta que el sensor podria estar fallant, com ara durant un lliscament, el sistema pot augmentar puntualment aquesta covariància per donar més credibilitat a la mesura inercial.

Aquesta fusió proporciona així una estimació de la posició i l'orientació del robot prou precisa per a les necessitats del prototip, sense requerir cap referència externa com un

sistema de posicionament de localització o balises fixes. És, en definitiva, la base de tot el sistema d'estimació de posició del present treball, i el motiu pel qual s'han escollit aquests sensors i aquest algorisme.

3. Disseny del sistema

En aquest capítol es descriu el disseny del prototip, partint de la selecció dels components que el formen, seguint amb l'esquema global de connexions i acabant amb el disseny de la placa de circuit imprès (PCB) que integra tota l'electrònica.

3.1 Selecció de components

A continuació es presenten els components escollits per construir el robot, justificant en cada cas el motiu de la seva elecció i recollint les seves especificacions tècniques principals, extretes dels datasheets oficials dels fabricants.

3.1.1 Unitat de control: ESP32-S3 N16R8

La unitat de control del robot és la placa de desenvolupament ESP32-S3 en la seva versió N16R8, l'elecció de la qual ja s'ha justificat en detall al capítol anterior. Es tracta d'un sistema en xip (SoC) construït al voltant d'un microprocessador de doble nucli Xtensa LX7 de 32 bits que pot funcionar fins a 240 MHz, amb connectivitat WiFi de 2,4 GHz i Bluetooth LE 5 integrada. La versió N16R8 incorpora 16 MB de memòria Flash i 8 MB de PSRAM, i disposa de 45 pins d'entrada/sortida programables, una xifra més que suficient per connectar tots els sensors i actuadors del sistema. Aquesta combinació de doble nucli, memòria àmplia i connectivitat sense fils és la que permet executar el filtre de Kalman en temps real alhora que es gestionen les comunicacions i la resta de perifèrics. La taula 3.1 en recull les especificacions principals, segons les dades del fabricant [1].

Paràmetre	Valor
Processador	Xtensa LX7 de 32 bits, doble nucli
Freqüència de rellotge	Fins a 240 MHz
Memòria SRAM interna	512 KB
Memòria Flash (N16R8)	16 MB
Memòria PSRAM (N16R8)	8 MB
Connectivitat sense fils	WiFi 802.11 b/g/n (2,4 GHz) i Bluetooth LE 5
Pins GPIO programables	45
Interfícies disponibles	I2C, SPI, UART, USB OTG, PWM, ADC, entre d'altres
Tensió d'alimentació lògica	3,3 V

Taula 3.1: Especificacions principals de la unitat de control ESP32-S3 (versió N16R8).

3.1.2 Unitat de mesura inercial: MPU6050

Per a la mesura del moviment del robot s'ha escollit la IMU MPU6050, un sensor de tipus MEMS que combina un acceleròmetre de tres eixos i un giroscopi de tres eixos en un mateix encapsulat, com s'ha descrit al capítol anterior. La seva comunicació es realitza mitjançant el bus I2C, cosa que simplifica el cablejat, i el seu cost reduït i la gran quantitat de documentació disponible la fan idònia per a aquest projecte. La taula 3.2 en resumeix les especificacions principals, segons les dades del fabricant [4].

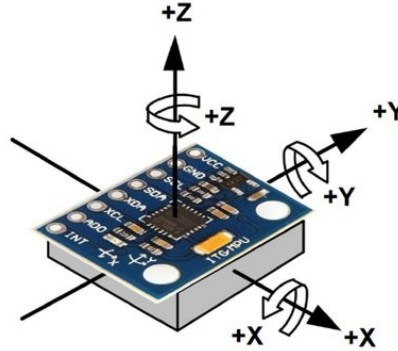


Figura 3.1: Mòdul de la IMU MPU6050 (placa GY-521) amb l'orientació dels eixos de mesura de l'acceleròmetre i el giroscopi.

Paràmetre	Valor
Eixos	3 (acceleròmetre) + 3 (giroscopi)
Rang de l'acceleròmetre	± 2 , ± 4 , ± 8 , ± 16 g (seleccionable)
Rang del giroscopi	± 250 , ± 500 , ± 1000 , ± 2000 °/s (seleccionable)
Interfície	I2C (fins a 400 kHz)
Tensió d'alimentació	2,375 V – 3,46 V
Biaix inicial de l'acceleròmetre	± 50 mg (eixos X, Y)
Convertidor	ADC de 16 bits per a cada canal

Taula 3.2: Especificacions principals de la IMU MPU6050.

3.1.3 Encoders òptics de roda

Per al càlcul de l'odometria s'utilitzen dos encoders òptics de ranura basats en el comparador LM393, amb el seu disc ranurat corresponent. Aquests mòduls disposen de quatre pins (alimentació, terra, sortida digital D0 i sortida analògica A0) i generen un pols digital cada cop que una ranura del disc travessa la forquilla òptica. En aquest projecte s'utilitza únicament la sortida digital D0, que es connecta a una entrada del microcontrolador configurada per interrupció, de manera que el comptatge de polsos no consumeix

temps de processador. La seva elecció es justifica pel baix cost, la facilitat d'integració i la possibilitat d'acoblar el disc directament als motors emprats.

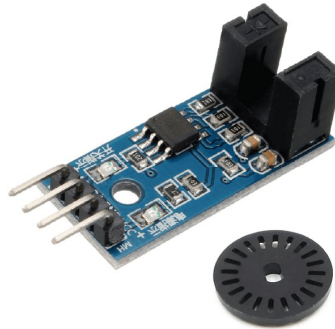


Figura 3.2: Mòdul d'encoder òptic de ranura basat en el comparador LM393. Font: [6].

3.1.4 Sensors d'ultrasons: HC-SR04

Per a la detecció d'obstacles s'han escollit tres sensors d'ultrasons HC-SR04, situats al davant, a l'esquerra i a la dreta del robot. Aquest sensor mesura la distància a un objecte emetent un tren de polsos ultrasònics i mesurant el temps que triga l'eco a tornar. La taula 3.3 en recull les especificacions principals, segons les dades del fabricant [7].

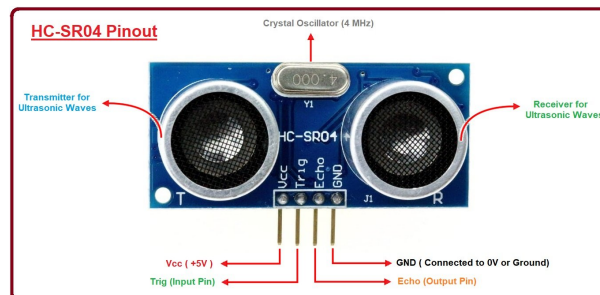


Figura 3.3: Sensor d'ultrasons HC-SR04 i la seva distribució de pins (Vcc, Trig, Echo i GND). Font: [8].

Paràmetre	Valor
Rang de mesura	2 cm – 400 cm
Precisió	± 3 mm
Angle de detecció	15°
Freqüència de treball	40 kHz
Tensió d'alimentació	5 V DC
Corrent de treball	15 mA
Senyal de trigger	Pols TTL de 10 μ s

Taula 3.3: Especificacions principals del sensor d'ultrasons HC-SR04.

Cal tenir en compte que el sensor HC-SR04 funciona a 5 V i, per tant, el seu pin de sortida (Echo) retorna un senyal de 5 V. Com que els pins de l'ESP32-S3 només toleren 3,3 V, ha estat necessari incorporar un divisor de tensió en cada línia Echo per adaptar el nivell, tal com es detallarà a l'apartat de disseny de la PCB.

3.1.5 Sensor de temperatura i humitat: DHT11

Per a la instrumentació ambiental del terreny s'utilitza un sensor DHT11, que mesura la temperatura i la humitat relativa i les transmet mitjançant un protocol digital de bus únic. S'ha escollit per la seva senzillesa, el seu baix cost i el fet que un sol pin de dades n'hi ha prou per llegir ambdues magnituds. La taula 3.4 en resumeix les especificacions, segons les dades del fabricant [9].

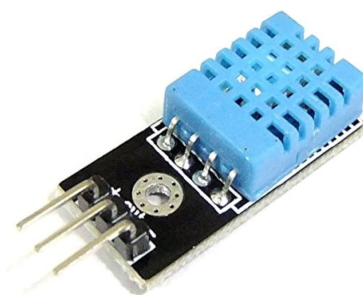


Figura 3.4: Sensor de temperatura i humitat DHT11. Font: [9].

Paràmetre	Valor
Rang de temperatura	0 °C – 50 °C
Precisió de temperatura	±2 °C
Rang d'humitat	20% – 90% RH
Precisió d'humitat	±5% RH
Resolució	1 °C / 1% RH
Tensió d'alimentació	3 V – 5,5 V
Interfície	Bus únic digital
Període de mostreig	≥ 1 s

Taula 3.4: Especificacions principals del sensor DHT11.

3.1.6 Driver de motors: L298N

El control dels motors es fa mitjançant un mòdul driver L298N, basat en el circuit integrat de doble pont en H de STMicroelectronics. Aquest driver permet controlar la velocitat i el sentit de gir de dos canals de motors de corrent continu de manera independent, regulant la velocitat mitjançant modulació per amplada de pols (PWM) sobre els pins d'habilitació

(ENA i ENB), i el sentit de gir mitjançant els pins d'entrada de cada canal (IN1–IN4). El mòdul incorpora un regulador de 5 V integrat que pot alimentar la lògica de control quan la tensió dels motors no supera els 12 V, així com proteccions tèrmiques davant de sobreescalfaments. Aquest regulador s'activa mitjançant un jumper (sovint etiquetat com a 5V-EN): amb el jumper col·locat, el mòdul genera la tensió regulada de 5 V a partir de la tensió de la bateria, que en aquest projecte serveix per alimentar tant la lògica del driver com la unitat de control i els sensors. Per aquest motiu, aquest jumper s'ha de mantenir sempre col·locat; si es retirés, l'alimentació de 5 V deixaria d'estar regulada correctament i podria malmetre l'ESP32. En aquest projecte, els dos motors de cada costat del robot es connecten en paral·lel al mateix canal, de manera que es controlen conjuntament. La taula 3.5 en recull les especificacions principals, segons les dades del fabricant [10].

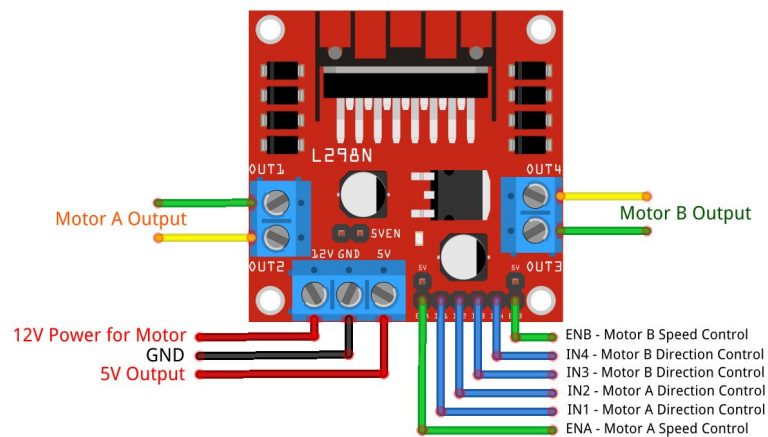


Figura 3.5: Mòdul driver de motors L298N i la seva distribució de pins i borns de connexió. Font: [11].

Paràmetre	Valor
Configuració	Doble pont en H (2 canals independents)
Tensió d'alimentació dels motors	5 V – 35 V
Tensió d'alimentació lògica	5 V
Corrent continu per canal	2 A (4 A en total)
Potència màxima dissipada	25 W
Control de velocitat	PWM (pins ENA i ENB)
Control de sentit	Pins d'entrada IN1–IN4
Caiguda de tensió típica	≈ 2 V

Taula 3.5: Especificacions principals del mòdul driver de motors L298N.

3.1.7 Motors i xassís

Per a la locomoció s'utilitzen quatre motors de corrent continu amb reductora del tipus TT, molt habituals en projectes de robòtica educativa, juntament amb les seves rodes corresponents. Aquests motors funcionen amb tensions de 3 a 6 V i, gràcies a la reductora, ofereixen un parell suficient per moure el xassís del prototip a una velocitat moderada. El conjunt es munta sobre un xassís de dues plantes fabricat a mida, que allotja els motors a la planta inferior i l'electrònica i la bateria a la superior.

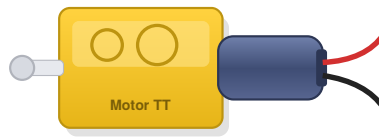


Figura 3.6: Motor de corrent continu amb reductora del tipus TT.

3.1.8 Alimentació

L'alimentació del sistema es proporciona mitjançant bateries de liti recarregables del tipus BRC 18650, amb una tensió nominal de 3,7 V i una capacitat de 4200 mAh cadascuna. Les cel·les es connecten en sèrie per assolir la tensió de treball necessària per alimentar els motors a través del driver L298N, mentre que la unitat de control i els sensors es nodreixen de les sortides regulades de 3,3 V i 5 V de la placa.

3.2 Esquema global

Un cop seleccionats tots els components, el pas següent és definir com s'interconnecten per formar el sistema complet. La figura 3.7 mostra l'esquema global del prototip, on l'ESP32-S3 actua com a unitat central de control i s'encarrega de comunicar-se amb tots els sensors i actuadors, així com d'executar el filtre de Kalman.

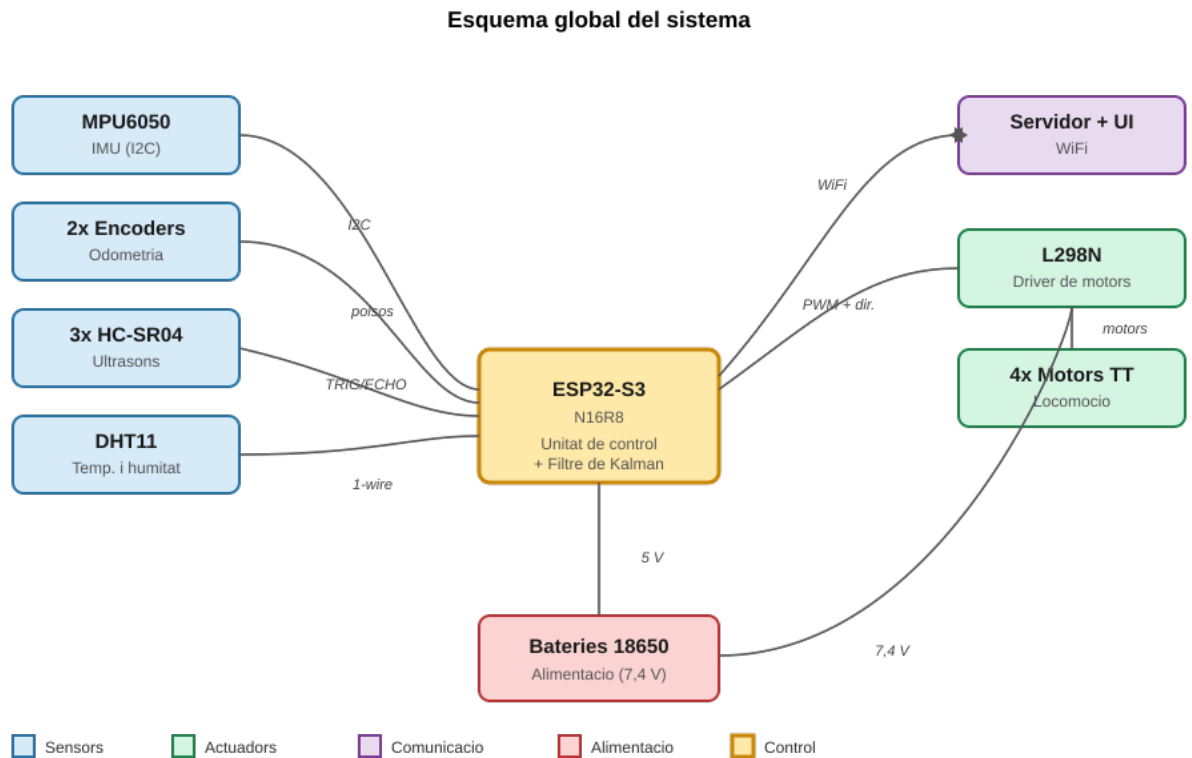


Figura 3.7: Esquema global del sistema al disseny inicial, amb l'ESP32-S3 com a unitat central de control.

Com es pot observar, l'ESP32-S3 rep les dades dels diferents sensors mitjançant diversos protocols de comunicació. La IMU MPU6050 es comunica a través del bus I2C, els encoders envien polsos digitals que es comptabilitzen per interrupció, els sensors d'ultrasons HC-SR04 funcionen amb senyals de trigger i echo, i el sensor DHT11 utilitza el seu protocol de bus únic. Pel que fa als actuadors, l'ESP32-S3 controla el driver L298N mitjançant senyals PWM i de direcció, i aquest al seu torn alimenta els quatre motors. La comunicació amb el servidor i la interfície d'usuari es realitza de manera sense fils a través de WiFi. Finalment, l'alimentació es proporciona mitjançant bateries 18650, que alimenten el driver L298N i, a través del seu regulador de 5 V, els components que ho requereixen. Cal precisar que aquest esquema correspon al disseny inicial, l'alimentació definitiva de la unitat de control, que finalment es va separar de la dels motors, es detalla al capítol 7.

3.3 Disseny de la PCB

Per integrar tota l'electrònica del prototip de manera robusta i ordenada, evitant el cablejat desordenat propi de les protoboards, es va dissenyar una placa de circuit imprès (PCB)

a mida. El disseny de l'esquemàtic i la disposició inicial de la placa es van realitzar amb el programa KiCad, una eina de disseny electrònic de codi obert àmpliament utilitzada tant en l'àmbit professional com acadèmic. Posteriorment, l'arranjament final de l'enrutament i la generació dels fitxers de fabricació es van dur a terme amb Altium Designer, tal com es detalla més endavant en aquest mateix apartat.

3.3.1 Esquemàtic

El primer pas del disseny va ser la creació de l'esquemàtic, on es defineixen tots els components i les seves connexions elèctriques. La figura 3.8 mostra l'esquemàtic complet del sistema dissenyat en KiCad.

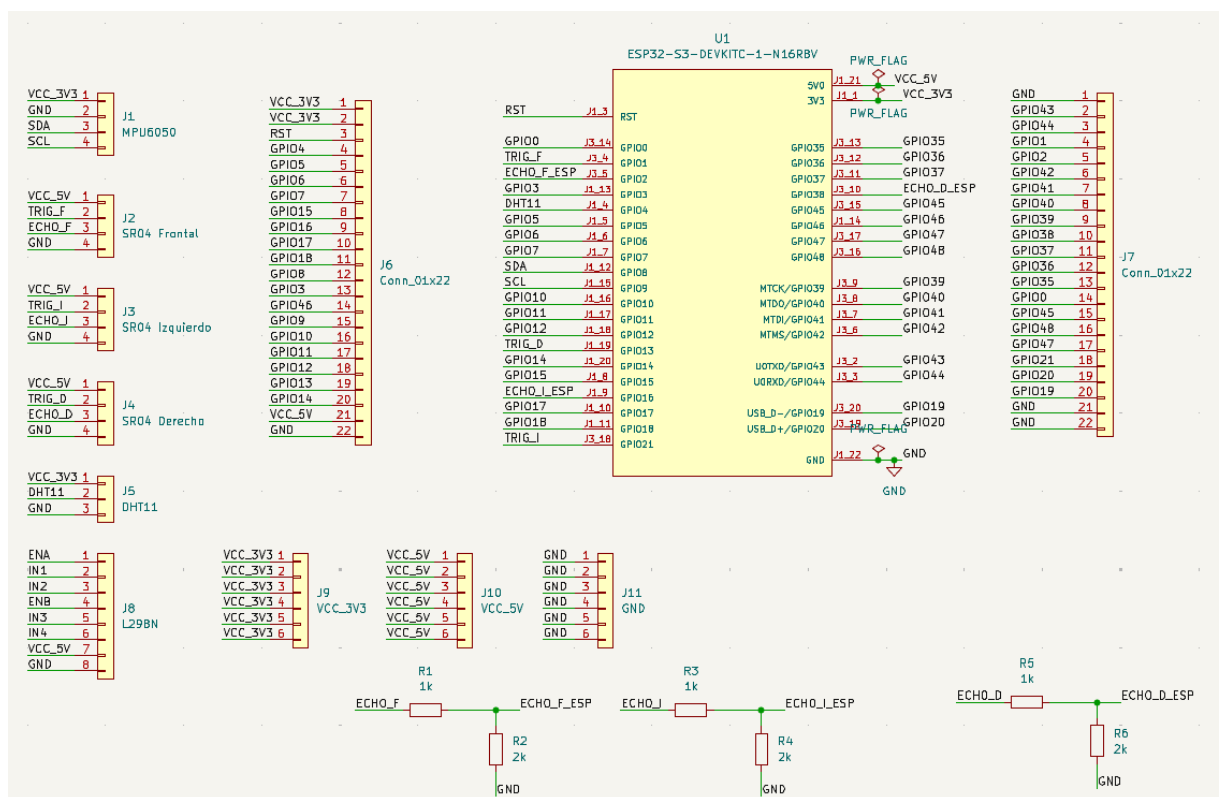


Figura 3.8: Esquemàtic del sistema dissenyat en KiCad.

Al centre de l'esquemàtic es troba el símbol de l'ESP32-S3 [12], al voltant del qual es connecten la resta d'elements mitjançant connectors. Cada sensor i actuador disposa del seu propi connector (J1 per a la MPU6050, J2 a J4 per als sensors d'ultrasons, J5 per al DHT11 i J8 per al driver L298N), de manera que els components es poden connectar i desconnectar còmodament de la placa. També es van incloure connectors per a les línies d'alimentació de 3,3 V i 5 V i per a la massa comuna, per si finalment calguessin més pins per un futur.

Un aspecte important del disseny és l'adaptació dels nivells de tensió dels sensors d'ultrasons. Com s'ha comentat anteriorment, el pin Echo de l'HC-SR04 retorna un senyal de 5 V que podria danyar els pins de l'ESP32-S3, que només toleren 3,3 V. Per solucionar-ho, es van incorporar tres divisors de tensió, un per a cada sensor d'ultrasons, formats cadascun per una resistència de 1 k Ω i una de 2 k Ω , que adapten el senyal de 5 V a un nivell segur de 3,3 V abans d'entrar al microcontrolador. Aquests divisors es poden veure a la part inferior de l'esquemàtic (resistències R1 a R6). La figura 3.9 en mostra l'esquema elèctric: el senyal d'echo de 5 V entra per la resistència R1 i, al punt intermedi entre R1 i R2, se n'obté la tensió reduïda a 3,3 V que s'envia a l'ESP32-S3.

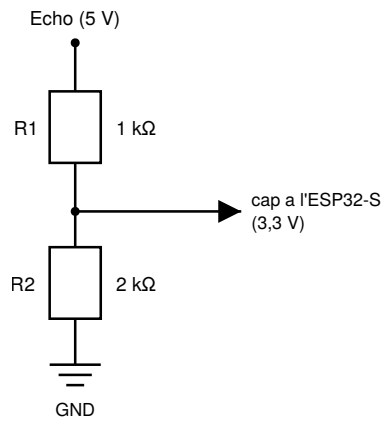


Figura 3.9: Esquema del divisor de tensió a la línia d'echo de cada sensor HC-SR04, amb $R_1 = 1 \text{ k}\Omega$ i $R_2 = 2 \text{ k}\Omega$.

3.3.2 Assignació de pins

La taula 3.6 recull l'assignació final dels pins de l'ESP32-S3 a cadascun dels components del sistema. Aquesta assignació s'ha definit tenint en compte les funcions especials d'alguns pins del microcontrolador i evitant els pins reservats per a la memòria PSRAM interna del mòdul N16R8.

Component	Senyal	Pin ESP32-S3
MPU6050	SDA	IO8
MPU6050	SCL	IO9
DHT11	Dades	IO4
Ultrasò frontal	TRIG / ECHO	IO1 / IO2
Ultrasò esquerre	TRIG / ECHO	IO21 / IO16
Ultrasò dret	TRIG / ECHO	IO13 / IO38
L298N (canal A)	ENA / IN1 / IN2	IO17 / IO5 / IO6
L298N (canal B)	ENB / IN3 / IN4	IO18 / IO7 / IO15
Encoder esquerre	D0	IO48
Encoder dret	D0	IO47

Taula 3.6: Assignació final dels pins de l'ESP32-S3 als components del sistema.

3.3.3 Disseny de la placa i enrutament

Un cop validat l'esquemàtic, es va passar al disseny físic de la placa, situant els components sobre una placa de dimensions 100 x 100 mm i traçant les pistes que connecten els seus pins segons les connexions definides a l'esquemàtic. La figura 3.10 mostra la disposició dels components abans de l'enrutament.

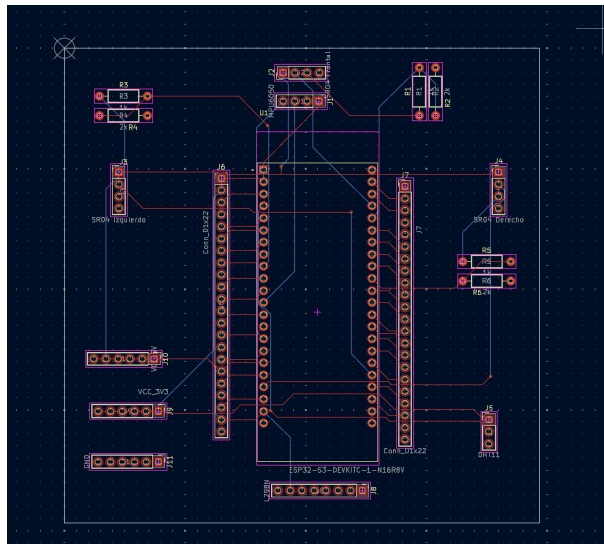


Figura 3.10: Disposició dels components a la PCB en KiCad, abans de l'enrutament.

Per al traçat de les pistes (*routing*) es va utilitzar el connector FreeRouting, un programa d'enrutament automàtic de codi obert que s'integra amb KiCad. Aquesta eina automatitza el procés de traçat, que de manera manual seria molt laboriós, generant un

enrutament que respecta les regles de disseny establertes. La figura 3.11 mostra l'entorn del FreeRouting durant el procés d'enrutament.

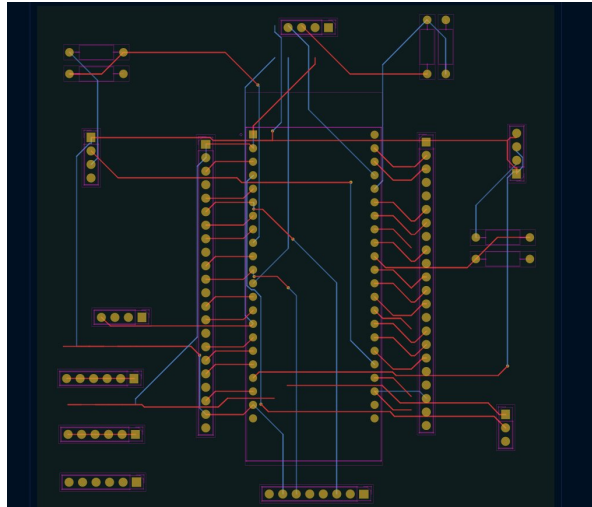


Figura 3.11: Procés d'enrutament automàtic de la placa amb el connector FreeRouting.

Un cop finalitzat l'enrutament i importat de nou a KiCad, el resultat és el que es mostra a la figura 3.12. En aquesta vista ja es poden observar totes les pistes traçades, tot i que s'aprecia que algunes són força fines, un aspecte que més endavant donaria problemes a l'hora de la fabricació.

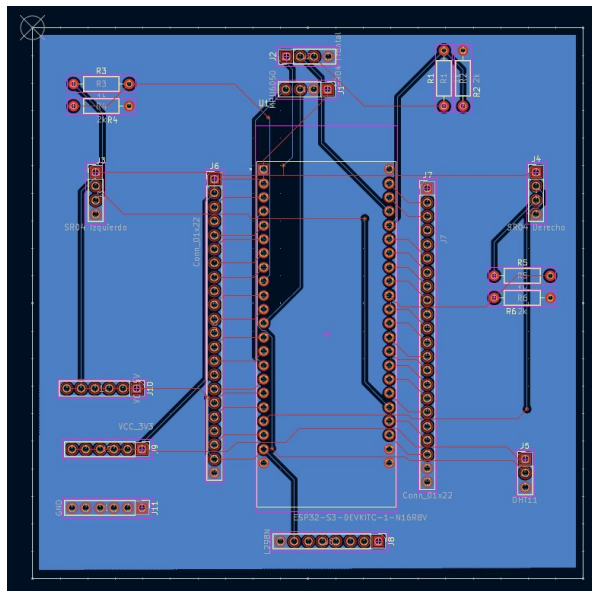


Figura 3.12: Placa enrutada en KiCad després d'aplicar el FreeRouting. S'observen algunes pistes molt fines.

3.3.4 Migració a Altium i fabricació

En arribar a la fase de generació dels fitxers de fabricació (*Gerbers*), va sorgir un primer problema. El KiCad, en combinació amb l'enrutament generat pel FreeRouting, deixava unes pistes massa fines i no generava correctament els fitxers Gerber. Per resoldre-ho, es va decidir migrar el disseny de la placa cap a Altium Designer, una eina de disseny electrònic professional, més potent i adequada per dur a terme aquest tipus de canvis, que permet engruixir les pistes i corregir l'enrutament amb molt més control. Així doncs, tot i que el disseny de l'esquemàtic i de la placa es va fer originalment en KiCad, l'arranjament final del traçat i la generació dels Gerbers es va dur a terme en Altium, aprofitant les seves capacitats més avançades.

Un cop arreglades les pistes en Altium, i després d'una revisió del disseny per part del tècnic del Departament d'Enginyeria Electrònica, en Javier Hellín, per comprovar-ne el correcte funcionament, els Gerbers es van enviar a fabricar a l'empresa JLCPCB. Tanmateix, la primera placa que es va rebre presentava dos problemes deguts a errors en el procés d'exportació de KiCad a Altium.

El primer problema va ser que algunes pistes no s'havien dibuixat. En exportar el disseny d'un programa a l'altre, per algun motiu no es van transferir totes les pistes, de manera que la placa fabricada tenia connexions que faltaven.

El segon problema afectava els pads dels pins. En l'exportació, els forats dels pins van quedar recoberts per resina al seu voltant, de manera que l'única zona on es podia soldar era l'interior del propi forat, i no la corona exterior del pad com és habitual. Això feia que soldar els components fos molt difícil i que en molts casos no s'aconguís un bon contacte elèctric.

Identificar la causa d'aquest segon problema no va ser immediat: després de cercar-ne l'origen, finalment, amb l'ajuda del tècnic Javier Hellín, es va trobar el problema. La configuració de la màscara de soldadura no es transfereix correctament en passar d'un programa a l'altre. La solució va consistir a entrar a les propietats de cada pad i modificar el paràmetre de la màscara de soldadura (*solder mask*) a mode manual, introduint-hi un valor petit i positiu de 0,1 mm, cosa que regenera automàticament la màscara correctament al voltant del pad.

La figura 3.13 mostra la primera versió de la placa fabricada per JLCPCB, en la qual encara es manifestaven aquests problemes.

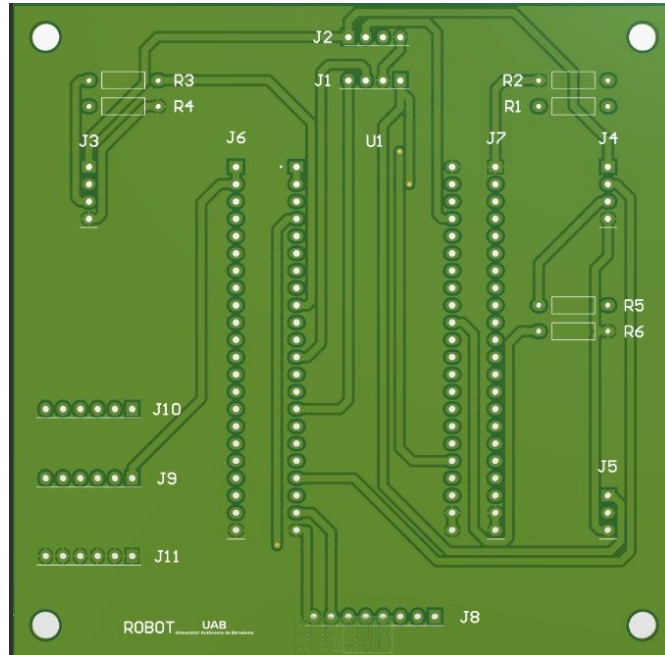


Figura 3.13: Primera versió de la PCB fabricada per JLCPCB, amb pistes que faltaven i problemes a la màscara de soldadura dels pads.

A més dels problemes anteriors, derivats de l'exportació, aquesta primera placa també va patir una dificultat relacionada amb el *footprint* de l'ESP32-S3 N16R8. Ni les llibreries de KiCad ni les d'Altium incloïen el *footprint* exacte d'aquest mòdul, de manera que es va haver de cercar i importar des d'un repositori extern de components. El que no s'esperava és que, tot i tractar-se del *footprint* del mateix mòdul ESP32-S3 N16R8, les mesures no eren correctes: el *footprint* era lleugerament més petit que el mòdul real, amb la separació entre pads desplaçada un pas respecte a la del component físic. Com a conseqüència, en aquesta primera placa els pads d'aquest connector no coincidien amb els pins reals del mòdul.

Després de corregir les pistes que faltaven, la configuració de la màscara de soldadura dels pads i el *footprint* de l'ESP32-S3, col·locant els pads desplaçats un pas i tornant a rutejar-los cap als pins corresponents del microcontrolador, es va tornar a enviar el disseny a fabricar. La segona versió de la placa, que es mostra a la figura 3.14, ja no presentava cap d'aquests problemes i va permetre soldar els components correctament i amb un bon contacte elèctric.

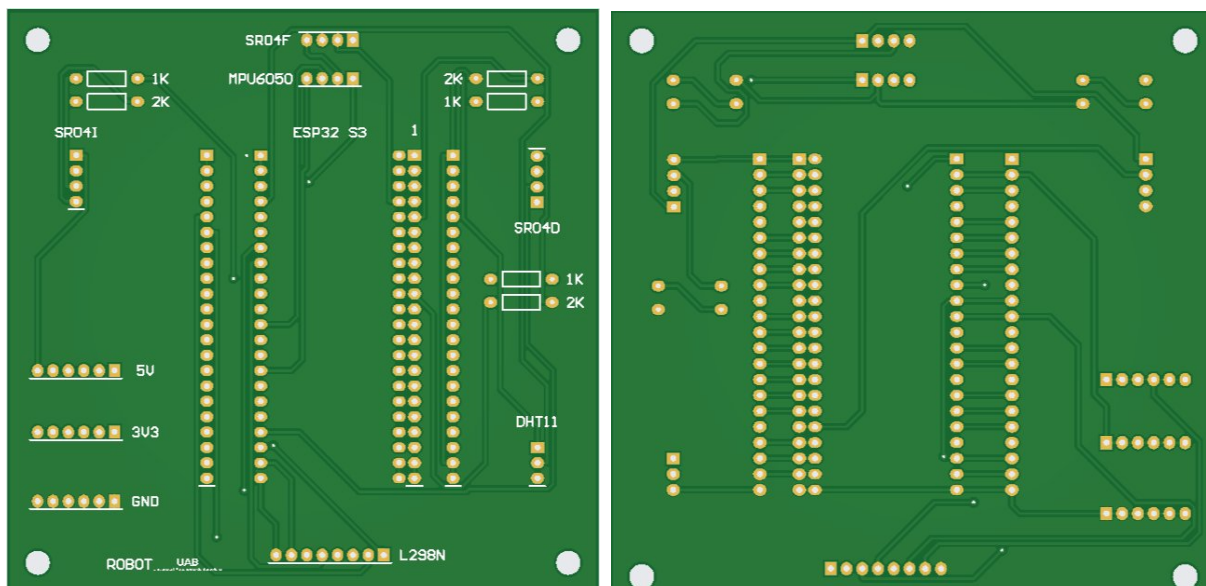
(a) Cara superior (*top*).(b) Cara inferior (*bottom*).

Figura 3.14: Versió final de la PCB fabricada per JLCPCB, amb els problemes resolts i llesta per soldar-hi els components. Es mostren les dues cares de la placa.

4. Modes de moviment i exploració

El moviment del robot el governa el backend, és a dir, el programa escrit en C++ dins de l'entorn Arduino IDE que s'executa a l'ESP32-S3 i que de manera contínua llegeix els sensors, decideix com s'ha de moure el vehicle i actua sobre els motors. Aquest capítol descriu la lògica d'aquest backend, la part que decideix *com es comporta* el robot en cadascun dels seus tres modes de funcionament: manual, automàtic i retorn. Tota la part de comunicació amb l'usuari (l'enviament d'ordres, el format dels missatges i la telemetria) es tracta de manera separada al capítol 6, dedicat a la interfície.

El backend manté en tot moment una variable d'estat que indica quin mode està actiu, definida amb una enumeració amb els valors `MODE_MANUAL`, `MODE_AUTO` i `MODE_RETURN`. A cada iteració del bucle principal s'executa la lògica corresponent al mode seleccionat, de manera que el canvi de mode és immediat i es pot fer en qualsevol moment. Per seguretat, quan es perd la connexió amb el client o se'n connecta un de nou, el robot torna automàticament al mode manual i atura els motors, de manera que el vehicle no pugui quedar mai en moviment fora de control.

4.1 Control dels motors

Els tres modes no actuen directament sobre el maquinari, sinó que comparteixen un conjunt reduït de primitives de moviment que encapsulen el control del driver L298N (taula 3.5): `motorsEndavant`, `motorsEnrere`, `motorsEsquerra`, `motorsDreta` i `motorsParar`. Cada primitiva estableix el sentit de gir de cada motor mitjançant els pins de direcció (IN1–IN4), fixa la velocitat aplicant un valor PWM als pins d'habilitació del driver (ENA i ENB), i actualitza les variables de sentit (`motor_left_dir` i `motor_right_dir`) que utilitzen els encoders per comptar correctament el desplaçament. Aquest darrer detall és important: com que les mateixes primitives les fan servir tots els modes, l'odometria es manté coherent independentment de qui ordeni el moviment. Les assignacions concretes dels pins es recullen a la taula 3.6.

El sistema treballa amb quatre velocitats predefinides. El mode manual utilitza `VEL` per al moviment en línia recta i `VEL_GIR`, més alta, per als girs, ja que un gir sobre l'eix requereix més parell per vèncer la inèrcia i el fregament. El mode automàtic, en canvi, es mou a velocitats més baixes per guanyar seguretat a prop dels obstacles: `VEL_CREUER` quan el camí està lliure i `VEL_PRUDENT` quan s'hi acosta. Com que els dos motors de cada costat es connecten en paral·lel al mateix canal del driver, el robot es comporta com un sistema de tracció diferencial: per avançar, els dos costats giren en el mateix sentit; per girar sobre si mateix, ho fan en sentits oposats. La figura 4.1 il·lustra aquest principi en el

cas d'un gir: quan el costat esquerre avança i el dret recula, el robot gira sobre si mateix cap a la dreta.

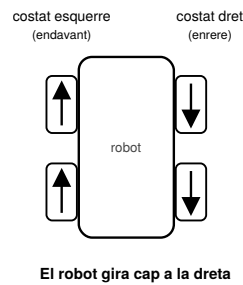


Figura 4.1: Principi de la tracció diferencial en un gir: amb el costat esquerre endavant i el dret enrere, el robot gira cap a la dreta.

4.2 Manual

En el mode manual, el robot es controla de manera remota des de la interfície d'usuari. La lògica del backend en aquest mode és senzilla: per cada ordre de moviment rebuda, el robot llegeix la direcció indicada, crida la primitiva de motor corresponent i en manté l'efecte fins que arriba una ordre nova o l'ordre d'aturada, tal com resumeix la figura 4.2. La descodificació del missatge que arriba de la interfície es detalla al capítol 6.

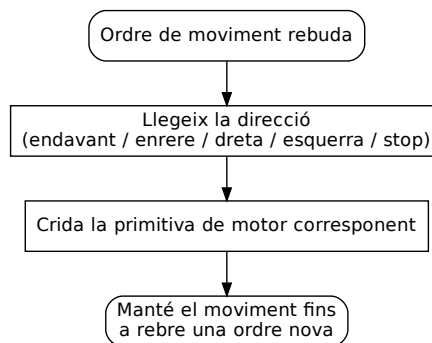


Figura 4.2: Diagrama de flux del mode manual: cada ordre rebuda s'associa a una primitiva de moviment.

A més, qualsevol ordre de moviment fa que el robot torni automàticament a mode manual si es trobava en un altre mode. D'aquesta manera, l'usuari pot recuperar el control immediat del vehicle simplement enviant una ordre, sense haver de canviar el mode de manera explícita.

4.3 Automàtic

En el mode automàtic, el robot es desplaça de manera autònoma evitant els obstacles que troba al seu pas, sense intervenció de l'usuari. Per fer-ho utilitza els tres sensors d'ultrasons HC-SR04 (figura 3.3) situats al davant, a l'esquerra i a la dreta del vehicle. La figura 4.3 mostra la lògica completa del mode.

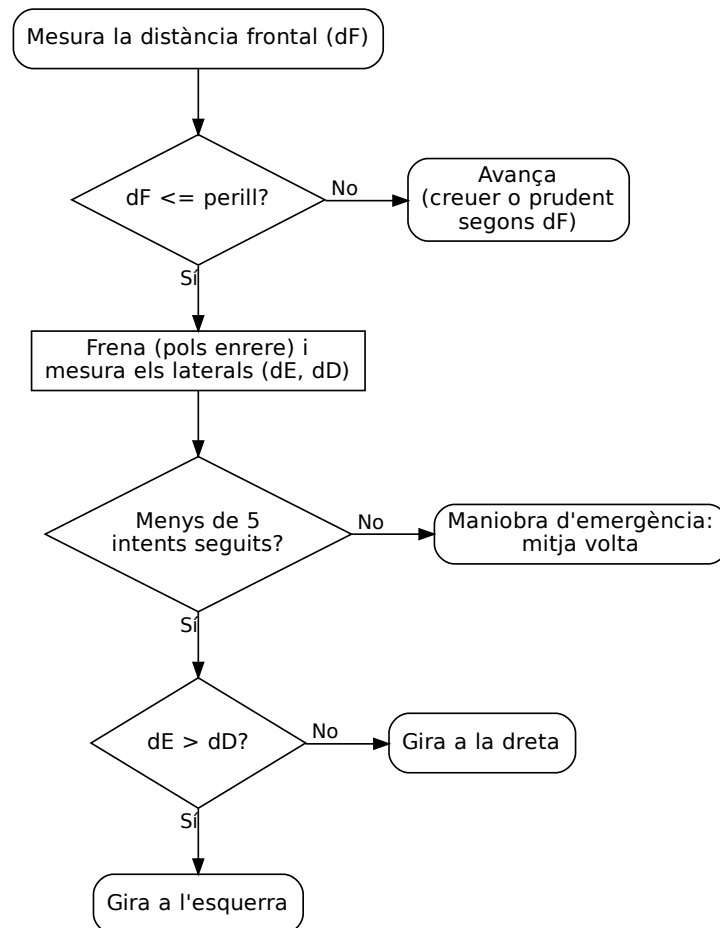


Figura 4.3: Diagrama de flux del mode automàtic d'evitació d'obstacles.

El comportament es regula amb tres llindars de distància. Mentre el sensor frontal mesura una distància superior al llindar de prudència, el robot circula a velocitat de creuer. Quan un obstacle s'acosta i la distància baixa d'aquest llindar, el robot redueix la velocitat per guanyar marge de reacció. Si l'obstacle entra dins del llindar de perill, el robot s'atura i inicia la maniobra d'evitació. Mentre el camí està lliure només es mesura el sensor frontal, amb una sola lectura ràpida, cosa que permet mantenir una freqüència de mostreig alta;

els sensors laterals només es consulten quan cal decidir cap a quin costat girar, i en aquest cas s'utilitza la mediana de tres lectures per obtenir una mesura més estable.

Quan es detecta un obstacle, abans de girar s'aplica un breu pols de frenada (un impuls curt cap enrere) que compensa la inèrcia. A continuació es mesuren els laterals i el robot gira cap al costat amb més espai lliure. Si un obstacle està molt a prop, el robot recula primer per guanyar marge, i si queda bloquejat en totes les direccions durant cinc intents consecutius, executa una maniobra d'emergència consistent en un gir de mitja volta per buscar una sortida.

Un aspecte clau de la implementació és que tota aquesta lògica és no bloquejant. Les pauses entre maniobres no aturen l'execució del programa, sinó que es gestionen amb una funció d'espera activa que durant tota la pausa manté viva la comunicació i la telemetria, i que permet abandonar el mode automàtic a l'instant si l'usuari ho ordena. La línia que ho fa possible és la comprovació que s'executa dins d'aquesta espera:

```
1 if (mode != MODE_AUTO) return false; // avorta la maniobra si es canvia  
    de mode
```

Cal tenir present, finalment, una limitació física d'aquest mode. Els sensors d'ultrasons tenen un con de detecció d'uns 15 graus, de manera que objectes prims com ara potes de cadira o de taula poden no ser detectats de manera fiable. Aquesta limitació és inherent a la tecnologia d'ultrasons i no es pot resoldre completament per programari; la seva mitigació passaria per incorporar sensors complementaris, com ara el sensor muntat sobre servomotor previst com a ampliació futura.

4.4 Retorn

El mode retorn permet que el robot torni de manera autònoma al punt d'origen, és a dir, a la posició (0,0) des de la qual va començar a moure's. Aquest mode es recolza directament en l'estimació de posició i orientació proporcionada per l'odometria i el filtre de Kalman (capítol 2), ja que el robot necessita saber on es troba i cap a on mira per poder calcular com tornar a l'origen.

L'estratègia implementada és el retorn en línia recta, que consta de tres fases, resumides al diagrama de flux de la figura 4.4. En primer lloc, el robot calcula la distància que el separa de l'origen i l'angle cap a aquest. En segon lloc, compara aquest angle objectiu amb la seva orientació actual i gira sobre si mateix fins a quedar encarat cap al punt (0,0), dins d'un marge de tolerància. Finalment, quan ja hi està encarat, avança en línia recta cap a l'origen i s'atura quan la distància és inferior a un petit lllindar, moment en què torna automàticament al mode manual.

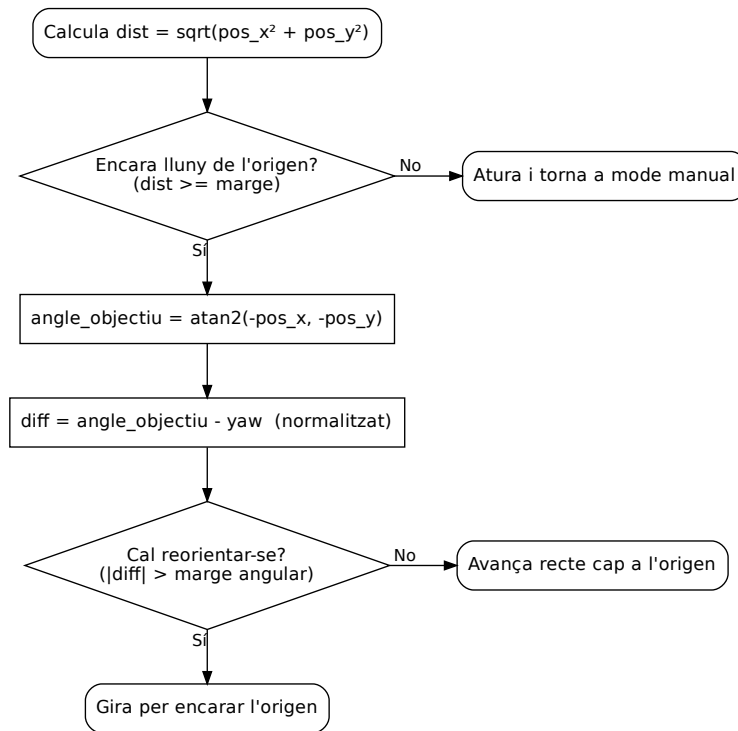


Figura 4.4: Diagrama de flux del mode retorn cap al punt d'origen.

L'angle cap a l'origen es calcula amb la funció trigonomètrica `atan2`, però amb un detall propi d'aquest projecte: l'ordre dels arguments s'ha ajustat a la convenció d'eixos del codi, de manera que la línia exacta és

```
1 float angle_objectiu = atan2f(-pos_x, -pos_y) * RAD2DEG;
```

A partir d'aquí, la diferència entre aquest angle i l'orientació actual es normalitza a l'interval $[-180, 180]$ graus per girar sempre pel camí més curt. Els paràmetres `RETORN_DIST_OK` (10 cm) i `RETORN_ANGLE_OK` (10 graus) defineixen, respectivament, com a prop ha d'arribar el robot de l'origen per considerar que ja hi és i quin marge angular s'admet per donar per bona l'orientació. S'ha optat per aquesta estratègia per la seva senzillesa i robustesa. L'evitació d'obstacles durant el retorn s'ha deixat com a línia de treball futura. Cal remarcar, a més, que la precisió del retorn depèn directament de la qualitat de la calibració de l'odometria: com més fiable sigui l'estimació de posició i orientació, més precís serà el retorn al punt d'origen.

5. Muntatge del prototip

Un cop dissenyats l'electrònica i els modes de funcionament, aquest capítol descriu el muntatge físic del prototip: com s'han distribuït els components sobre el xassís, com s'ha resolt l'alimentació i per què es va passar del muntatge inicial en protoboard a la PCB dissenyada a mida.

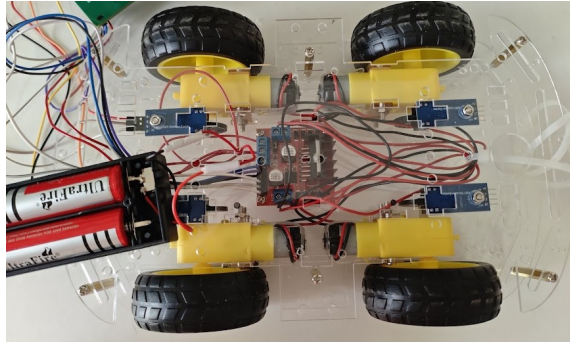
5.1 Disposició dels components al xassís

El prototip es construeix sobre un xassís de dues plantes fabricat a mida. La planta inferior allotja els quatre motors de corrent continu amb reductora del tipus TT amb les seves rodes, mentre que la planta superior queda reservada per a l'electrònica i l'alimentació. Aquesta separació en alçada manté ordenat el conjunt i facilita l'accés tant a la placa com a les bateries.

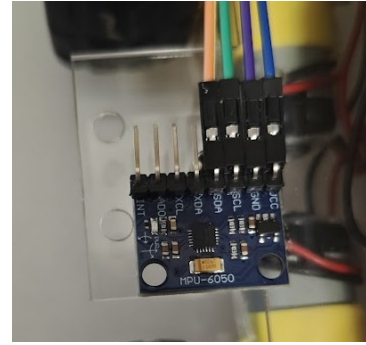
El driver L298N es col·loca a la planta inferior, prop dels motors. Aquesta decisió respon a dos motius. D'una banda, situar a baix un dels components més pesats ajuda a mantenir el centre de gravetat del robot baix, cosa que millora l'estabilitat durant el moviment i els girs. De l'altra, en quedar a tocar dels motors, els cables de potència que van del driver als motors són curts i directes, cosa que redueix el desordre del cablejat i les possibles interferències i caigudes de tensió a les línies de més corrent del sistema (figura 5.1a).

Els sensors es distribueixen segons la seva funció. Els tres sensors d'ultrasons HC-SR04 es munten a la part frontal i als laterals, orientats cap al davant, l'esquerra i la dreta, per cobrir les direccions en què el robot necessita detectar obstacles. Sobre l'eix de la roda posterior de cada costat, és a dir, a la part del darrere del robot, s'hi acobla un disc ranurat solidari amb l'encoder òptic corresponent. El sensor de temperatura i humitat DHT11 es fixa a la planta superior, juntament amb la placa.

La unitat de mesura inercial MPU6050 també es fixa a la planta superior, plana i amb els seus eixos alineats amb els del robot: l'eix X cap endavant, l'eix Y cap a un costat i l'eix Z vertical. Aquesta alineació és important perquè el giroscopi mesuri directament el yaw, el gir del robot sobre l'eix vertical, que és el que fa servir l'odometria.



(a) Driver L298N a la planta inferior, a tocar dels motors.



(b) Posició i orientació de la IMU MPU6050 sobre el xassís.

Figura 5.1: Disposició de components sobre el xassís: (a) el driver L298N a la planta inferior i (b) la IMU MPU6050 a la planta superior, alineada amb els eixos del robot.

5.2 Alimentació i bateries

L'alimentació de potència es proporciona amb bateries de liti 18650. La seva col·locació s'ha pensat buscant un equilibri de pes sobre el xassís i, sobretot, l'accessibilitat: com que durant el desenvolupament i les proves cal extreure-les i recarregar-les sovint, s'han disposat de manera que es puguin treure i tornar a posar còmodament sense haver de desmuntar la resta del robot.

Un dels punts més laboriosos del muntatge va ser, precisament, fer arribar els cables d'alimentació des de la caixa de bateries, situada a la planta superior, fins al driver L298N, situat a la planta inferior. L'estructura del xassís deixava poc marge de moviment per passar-hi el cablejat, i a més els cables que sortien de la caixa de bateries eren massa curts per cobrir aquesta distància. Per allargar-los es van empalmar amb trossos de cable addicionals, unint-los i cargolant-los entre si i recobrint la unió amb cinta aïllant, ja que no es disposava de soldador.

L'esquema d'alimentació definitiu no es va decidir des de l'inici, sinó que és el resultat d'un procés de diagnòstic posterior. En aquest esquema final, l'ESP32-S3 s'alimenta de manera independent per garantir-ne l'estabilitat. Tot aquest procés es descriu en detall al capítol 7, dedicat a les proves i correccions.

5.3 Del protoboard a la PCB

A causa de l'error de la primera versió de la PCB explicat al capítol 3 i del temps que va trigar a arribar la nova placa, el prototip es va haver de muntar inicialment sobre una protoboard, que permet connectar i recol·locar components ràpidament durant les

primeres fases de desenvolupament. Tanmateix, aquest tipus de muntatge va resultar ser una font constant de problemes en un robot que es mou.

La figura 5.2 mostra el prototip en aquesta primera fase, amb tot el cablejat muntat sobre la protoboard. A la part del davant s'hi va col·locar una brida que feia de para-xocs: durant les proves el robot es podia endur algun cop contra parets o objectes, i així aquests impactes els rebia la brida en comptes dels components, reduint al mínim el risc.

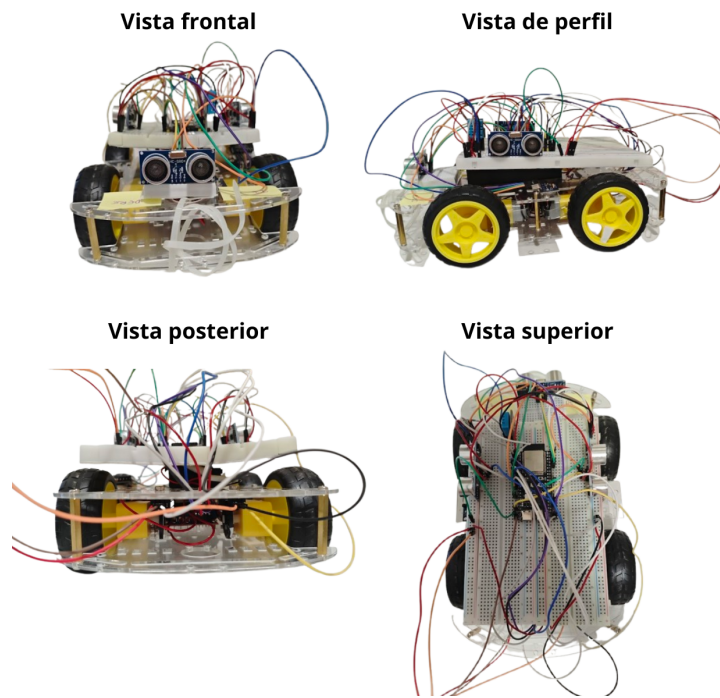


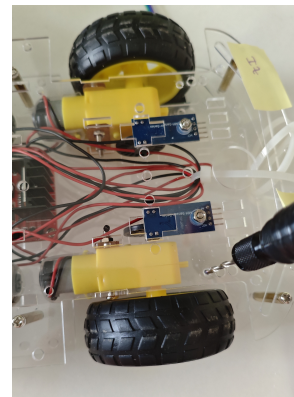
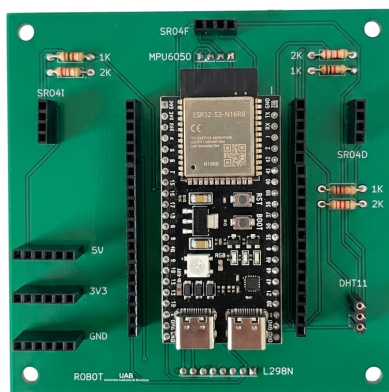
Figura 5.2: Prototip en la fase inicial, muntat sobre protoboard i amb la brida de protecció frontal.

Durant el desenvolupament van aparèixer nombrosos errors intermitents difícils de diagnosticar: la IMU deixava de respondre puntualment, el WiFi requeia sense motiu aparent i els encoders comptaven malament. L'origen de bona part d'aquests problemes no era el programari, sinó la naturalesa de les connexions de la protoboard. Les unions per pressió d'una protoboard presenten una resistència de contacte elevada i variable que, amb la vibració contínua dels motors, canvia constantment i fins i tot arriba a obrir-se de manera momentània. Això introdueix talls i transitoris, és a dir, caigudes i pics de tensió, a les línies d'alimentació i de massa, que es propaguen a la resta del sistema. Com que afecten l'alimentació compartida i les referències de tensió, aquests transitoris es manifesten com a comportaments erràtics de la part digital, difícils de distingir d'un error de programari.

La conclusió és clara: una protoboard en un robot en moviment és intrínsecament poc fiable. Aquesta constatació és el que va motivar el disseny i la fabricació de la PCB descrita al capítol 3. Una placa soldada elimina aquestes unions per pressió, fixa de manera permanent totes les connexions amb una resistència de contacte baixa i estable, i aporta la robustesa necessària perquè res no es desconnecti amb el moviment del robot. A més, una placa de circuit imprès dissenyada a mida aporta professionalitat i elegància a un projecte com aquest TFG, més enllà de la millora purament tècnica.

Un cop rebuda la placa fabricada, es va soldar amb tots els seus components. Per als mòduls principals, com l'ESP32-S3, es van fer servir sòcols, de manera que es poden inserir i extreure sense quedar soldats directament a la placa, cosa que en facilita la substitució i la reparació. Un cop soldada i muntada, la placa va quedar llesta per integrar-se al robot.

Per fixar la PCB al xassís va caldre perforar-lo amb un trepant, ja que no portava forats al lloc on calia subjectar la placa (figura 5.3b). Es va decidir muntar la PCB elevada, de manera que les bateries 18650 quedessin còmodament a sota i fossin fàcils d'accedir per a la càrrega.

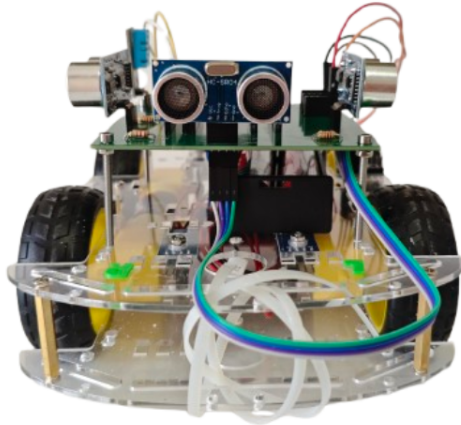


(a) PCB un cop soldada amb tots els sòcols i (b) Perforació del xassís amb un trepant per l'ESP32-S3. fixar la PCB.

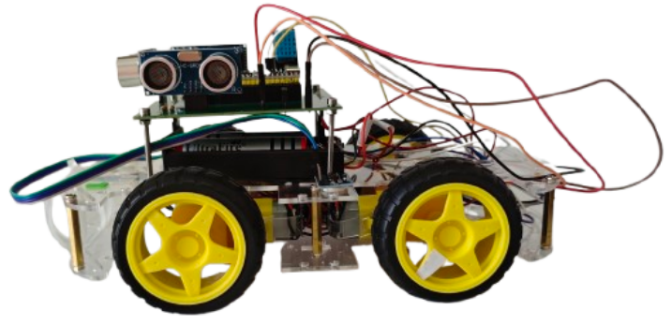
Figura 5.3: Preparació de la PCB i el xassís: (a) la placa soldada amb tots els sòcols i (b) la perforació del xassís per fixar-la.

La figura 5.4 mostra el prototip final, ja amb la PCB integrada al robot en substitució de la protoboard. La MPU6050 es va mantenir en la mateixa posició que en el prototip anterior, ben plana, per garantir una orientació estable i alineada amb els eixos del robot. Per estètica, es van imprimir amb impressora 3D dues lletres que indicaven la dreta i l'esquerra del robot.

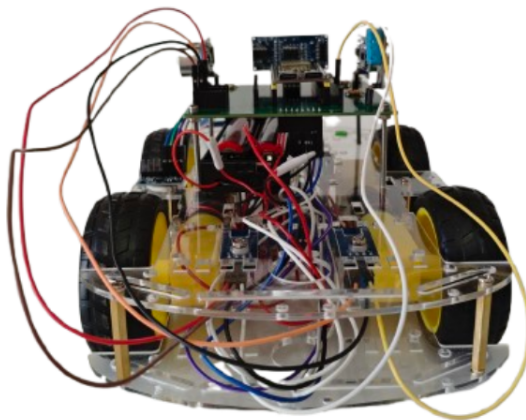
Vista frontal



Vista de perfil



Vista posterior



Vista superior

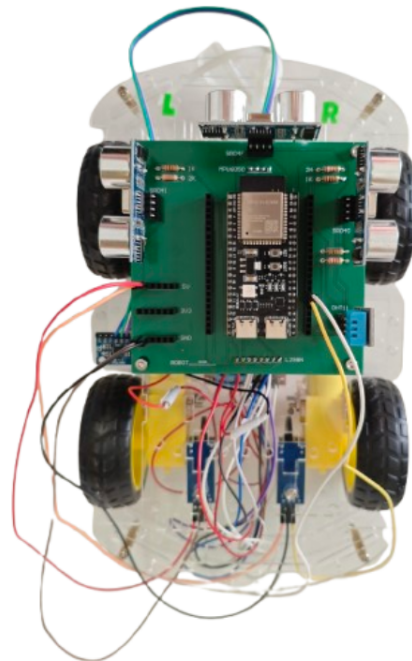


Figura 5.4: Prototip final amb la PCB muntada sobre el robot.

6. Interfície d'usuari i recopilació de dades

Aquest capítol descriu com es comunica el robot amb l'exterior i com l'usuari interacciona amb ell. Es presenta l'arquitectura de comunicació sense fils, el protocol de dades que s'intercanvien el robot i l'ordinador, i la interfície d'usuari que permet controlar el vehicle, visualitzar l'estat dels sensors i comparar el senyal en cru amb el senyal filtrat pel filtre de Kalman.

6.1 Arquitectura de comunicació

L'arquitectura de comunicació entre el robot i l'ordinador es basa en una connexió sense fils a través de WiFi, aprofitant la connectivitat integrada de l'ESP32-S3. El microcontrolador actua com a servidor WebSocket [13], escoltant en el port 81 mitjançant la llibreria `arduinoWebSockets`, mentre que el navegador de l'ordinador, on s'executa una aplicació web desenvolupada amb Next.js [14], hi connecta com a client. La figura 6.1 resumeix aquesta arquitectura.

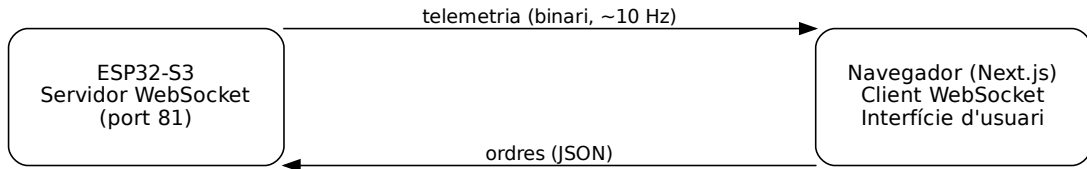


Figura 6.1: Arquitectura de comunicació entre el robot i la interfície d'usuari.

L'ESP32-S3 es connecta en mode estació, és a dir, com un dispositiu client que s'uneix a una xarxa WiFi ja existent, igual que ho faria un telèfon o un ordinador. La xarxa a la qual es connecta és un punt d'accès (un *hotspot*) que comparteix el mateix ordinador, i és sobre aquesta xarxa que actua com a servidor WebSocket al port 81. El navegador s'hi connecta indicant l'adreça IP que el robot rep dins d'aquesta xarxa. Aquesta arquitectura directa, només entre l'ordinador i el robot, sense cap encaminador ni altres dispositius pel mig, dona una latència baixa i evita qualsevol saturació: a 10 Hz amb trames de 96 bytes, el tràfic és d'uns 960 bytes per segon, una xifra menyspreable. Com a contrapartida, el robot ha de romandre dins de la zona de cobertura del *hotspot* de l'ordinador; si en surt, perd la connexió. Un cop establerta la connexió WebSocket, el canal queda obert de manera permanent i permet l'intercanvi de missatges en totes dues direccions, cosa

imprescindible per a una telemetria en temps real i un control fluid. Per evitar que el mode d'estalvi d'energia del WiFi introdueixi retards, aquest es desactiva explícitament al backend amb una sola instrucció:

```
1 WiFi.setSleep(false); // desactiva l'estalvi d'energia del WiFi
```

Tot i que l'amplada de banda necessària és molt baixa, a la pràctica, amb el prototip sobre la PCB, de tant en tant apareixen petits problemes de connectivitat o moments en què la interfície es satura momentàniament a causa de l'arribada contínua de dades. Es tracta d'incidències puntuals que no impedeixen el funcionament normal del sistema, però que caldria depurar en una versió més acabada, per exemple afinant encara més la freqüència de telemetria o la gestió del refresc al navegador.

6.2 Protocol de dades

Sobre la connexió WebSocket s'intercanvien dos tipus de missatges, cadascun amb un format adequat al seu propòsit: trames binàries per a la telemetria i missatges de text en format JSON per al control.

Telemetria: trames binàries

Les dades dels sensors s'envien del robot a la interfície en forma de trames binàries en *little-endian* (és a dir, amb el byte menys significatiu de cada valor numèric col·locat primer, que és l'ordenació nativa del processador de l'ESP32), a una freqüència de 10 Hz. Aquesta freqüència es va reduir des dels 100 Hz inicials, que resultaven innecessaris i sobrecarregaven el sistema, fins als 10 Hz actuals, suficients per a una visualització fluida. S'ha triat un format binari perquè, en transmetre's moltes vegades per segon, és molt més eficient que el text: cada camp ocupa exactament la seva mida i no cal codificar ni interpretar cap estructura textual. Cada trama ocupa 96 bytes i conté la marca de temps, les lectures crues de la IMU (acceleració en m/s^2 i velocitat angular en rad/s), els resultats dels dos filtres de Kalman, incloent-hi per a cadascun l'angle de l'acceleròmetre o dels encoders, l'angle integrat del giroscopi, l'estimació filtrada i el biaix del giroscopi estimat pel propi filtre, els comptadors dels dos encoders, la posició estimada (x, y) , la temperatura i la humitat, i les distàncies dels tres sensors d'ultrasons.

La taula 6.1 en detalla l'estructura camp a camp, amb el desplaçament en bytes des de l'inici de la trama, el nom del camp i el seu tipus de dada.

Desplaçament	Camp	Tipus	Mida
0	Marca de temps	uint32	4 B
4	Acceleració X	float32	4 B
8	Acceleració Y	float32	4 B
12	Acceleració Z	float32	4 B
16	Velocitat angular X	float32	4 B
20	Velocitat angular Y	float32	4 B
24	Velocitat angular Z	float32	4 B
28	Pitch (acceleròmetre)	float32	4 B
32	Pitch (giroscopi)	float32	4 B
36	Pitch (Kalman)	float32	4 B
40	Biaix del giroscopi (pitch)	float32	4 B
44	Yaw (giroscopi)	float32	4 B
48	Yaw (encoders)	float32	4 B
52	Yaw (Kalman)	float32	4 B
56	Biaix del giroscopi (yaw)	float32	4 B
60	Comptador encoder esquerre	int32	4 B
64	Comptador encoder dret	int32	4 B
68	Posició X	float32	4 B
72	Posició Y	float32	4 B
76	Temperatura	float32	4 B
80	Humitat	float32	4 B
84	Distància frontal	int32	4 B
88	Distància esquerra	int32	4 B
92	Distància dreta	int32	4 B

Taula 6.1: Estructura de la trama binària de telemetria (96 bytes, little-endian).

Control: ordres JSON

Les ordres que l'usuari envia al robot són poc freqüents i es transmeten com a text en format JSON, més còmode de llegir i d'ampliar. Cada ordre inclou un camp `cmd` que n'indica el tipus. La taula 6.2 en recull les principals.

Ordre (cmd)	Paràmetres	Funció
move	direction	Mou el robot (forward, backward, left, right, stop) i activa el mode manual.
mode	value	Canvia de mode (manual, auto, return).
telemetry	rate_hz	Ajusta la freqüència de la telemetria.
reset	angle	Reinicia els angles estimats i els filtres.
reset	position	Reinicia els encoders i la posició a l'origen.

Taula 6.2: Ordres de control que la interfície envia al robot.

Per exemple, per fer avançar el robot o per canviar-lo a mode automàtic, la interfície envia missatges com els següents:

```
1 {"cmd": "move", "direction": "forward"}
2 {"cmd": "mode", "value": "auto"}
3 {"cmd": "telemetry", "rate_hz": 10}
```

A més, el robot informa la interfície del mode en què es troba enviant un missatge de text JSON cada cop que aquest canvia, de manera que la interfície sempre en mostra l'estat correcte.

6.3 Interfície d'usuari

Disseny previ de la interfície

Abans de començar a programar, es va dissenyar l'aspecte que es volia que tingués la interfície. Amb l'eina web Figma es va elaborar una primera maqueta visual que definia com s'havien de distribuir els diferents elements (els controls, les gràfiques i el mapa) i quin havia de ser l'aspecte general de l'aplicació. Aquesta maqueta va servir de guia durant la implementació posterior, de manera que la programació es va poder centrar a reproduir un disseny ja pensat prèviament en comptes d'improvisar-lo sobre la marxa.

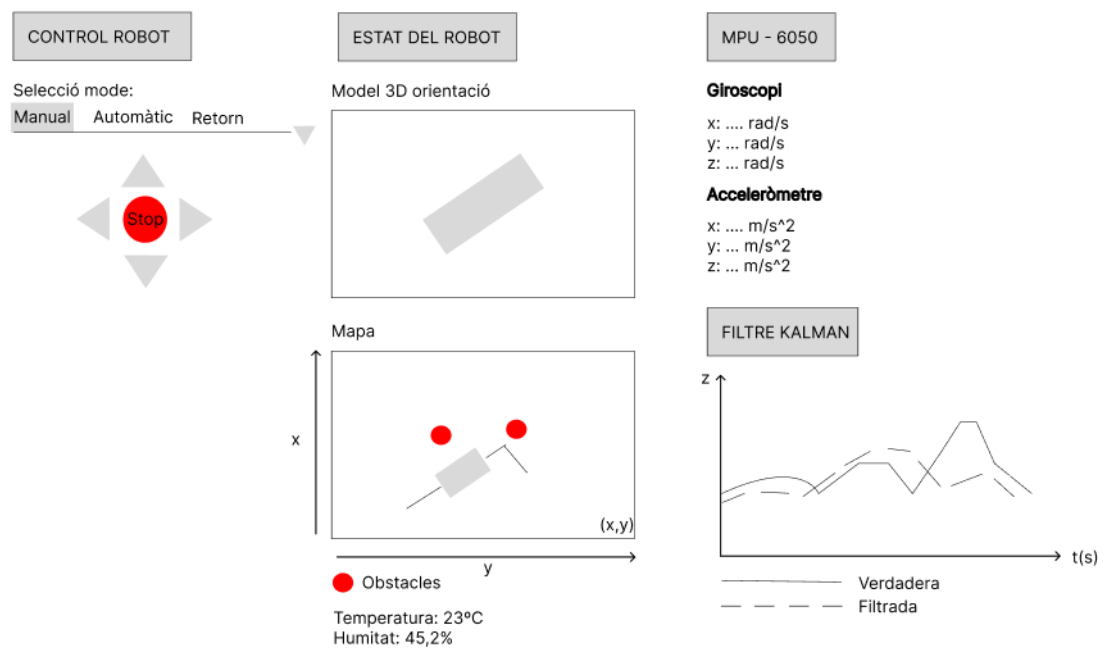


Figura 6.2: Disseny previ de la interfície amb el mode manual seleccionat.

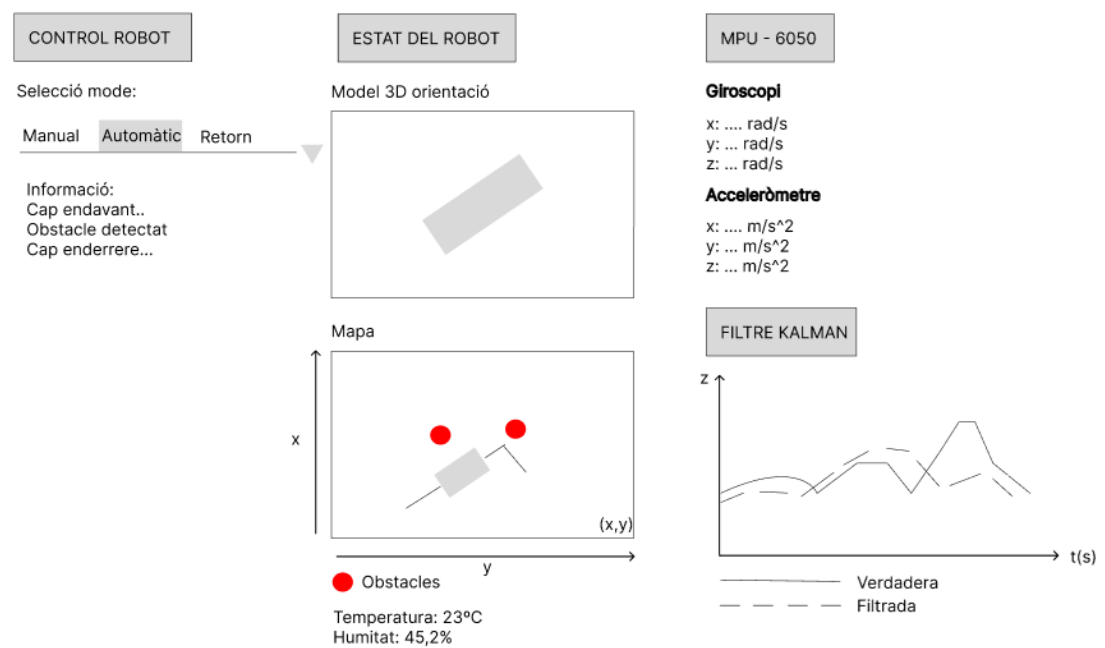


Figura 6.3: Disseny previ de la interfície amb el mode automàtic seleccionat.

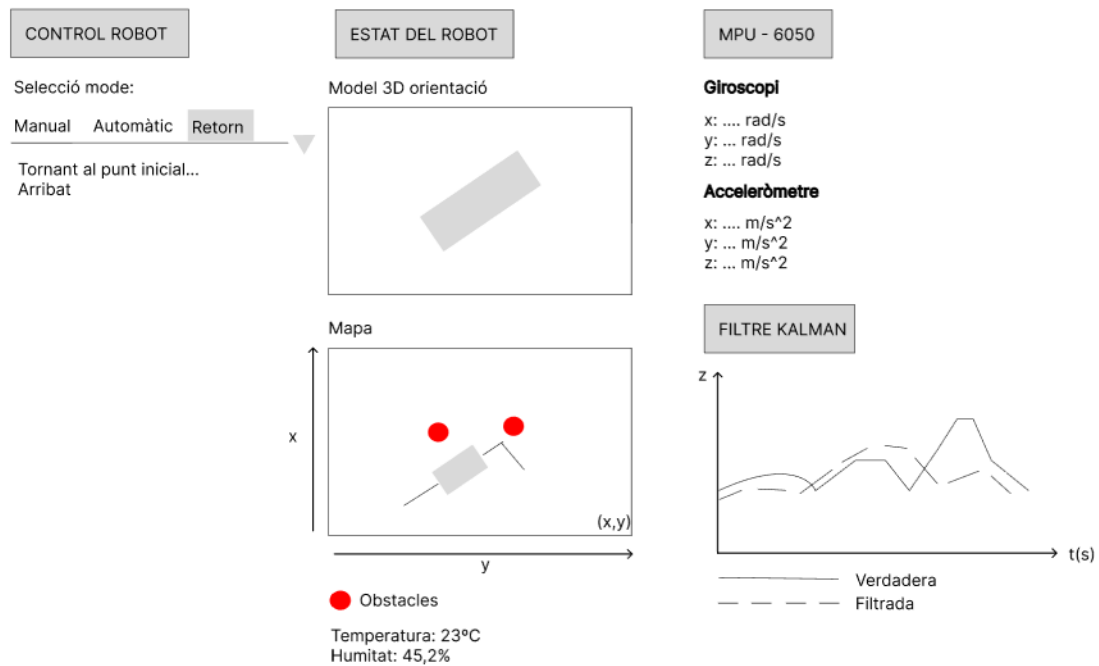


Figura 6.4: Disseny previ de la interfície amb el mode retorn seleccionat.

Implementació

La interfície d'usuari és una aplicació web desenvolupada amb Next.js i TypeScript (React) que s'executa a l'ordinador i es connecta al robot com a client WebSocket. L'estil visual es construeix amb Tailwind CSS [15], la gràfica del filtre de Kalman es dibuixa amb la llibreria Recharts, i el mapa d'odometria es renderitza sobre un *canvas* HTML. La recepció de dades es fa decodificant les trames binàries mitjançant un objecte **DataView** , que llegeix cada camp de la trama de 96 bytes a partir del seu desplaçament i tipus. En rebre cada trama, l'aplicació la descodifica camp a camp i n'actualitza la visualització; per no saturar el ritme de refresc del navegador amb la freqüència de la telemetria, l'actualització de les gràfiques es limita i es gestiona de manera eficient.

El desenvolupament d'aquesta interfície va suposar un repte important, ja que era la primera vegada que es programava una aplicació d'aquest tipus. Per això, abans de començar, es van valorar diferents opcions per construir la part web. Una primera possibilitat era fer servir JavaScript pur amb HTML i CSS, però aquesta via obliga a gestionar manualment l'estructura del projecte, l'actualització de la interfície i la reactivitat de les dades, cosa que es torna feixuga a mesura que l'aplicació creix. Una segona opció era utilitzar una biblioteca com React [16], que facilita molt la construcció d'interfícies reactives basades en components, però que per si sola no aporta una estructura de projecte ni eines de desenvolupament integrades. Finalment es va triar Next.js [14], un entorn de

treball construït sobre React que hi afegeix una estructura de projecte clara, un servidor de desenvolupament, suport natiu per a TypeScript i un conjunt d'eines ja integrades. És, doncs, una solució alhora més completa, perquè inclou de sèrie allò que amb React s'hauria de configurar a part, i més senzilla de posar en marxa, motiu pel qual es va escollir per a aquest projecte.

La figura 6.5 mostra l'aspecte general de la interfície un cop implementada. A la captura el robot apareix desconnectat, de manera que els panells es veuen sense dades en temps real, cosa que permet apreciar-ne amb claredat l'estructura i la distribució.

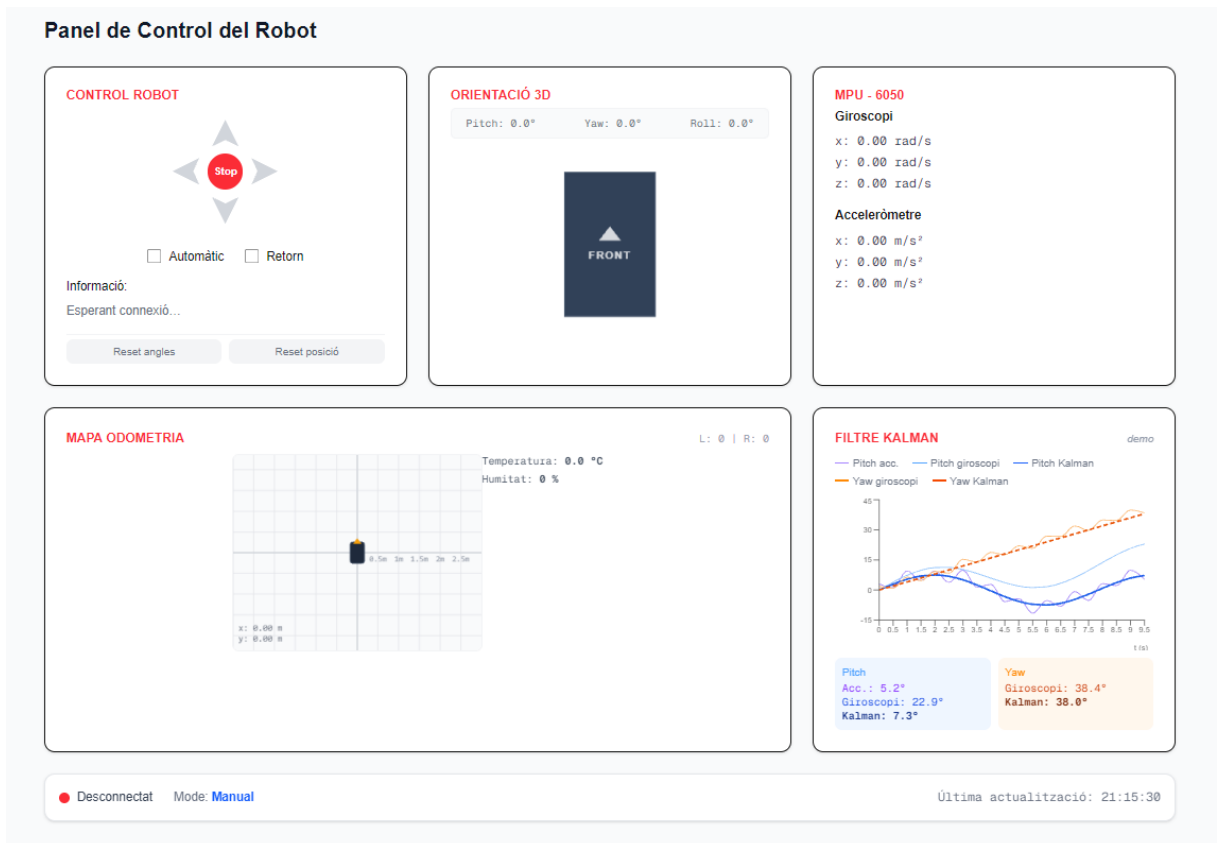


Figura 6.5: Aspecte general de la interfície d'usuari, amb el robot desconnectat.

La interfície s'organitza en dues parts. D'una banda, un panell de control permet teleoperar el robot i canviar de mode; cada acció es tradueix en una de les ordres JSON descrites a l'apartat anterior. De l'altra, diversos panells mostren en temps real l'estat del vehicle:

- **Sensor inercial.** Els valors de l'acceleròmetre (en m/s^2) i del giroscopi (en rad/s), és a dir, les lectures crues de la IMU.
- **Comparació del filtre de Kalman.** Una gràfica en temps real que superposa el senyal en cru i el filtrat, detallada més avall.

- **Orientació 3D.** Una representació tridimensional del robot que es mou segons l'orientació estimada a partir de la IMU, per visualitzar de manera intuïtiva cap a on apunta.
- **Encoders i mapa d'odometria.** Els comptadors de polsos de cada roda i un mapa, dibuixat sobre un *canvas*, que mostra la posició estimada (x, y) del robot respecte a l'origen i el rastre del seu recorregut. Sobre aquest mapa s'hi representen també els obstacles: les lectures dels tres sensors d'ultrasons es dibuixen com a rajos que surten del robot cap a on detecta cada obstacle, a la distància mesurada i només quan hi ha alguna cosa a menys d'un metre, girant amb l'orientació del robot.
- **Entorn.** Les lectures de temperatura i humitat del DHT11, que es mostren sota el mapa d'odometria.

La visualització del filtre de Kalman

Un dels elements centrals de la interfície, i una de les demostracions principals del treball, és la representació gràfica del filtre de Kalman. Per entendre-la cal explicar primer una decisió de disseny important: el sistema implementa **dos filtres de Kalman independents**, construïts sobre una mateixa estructura reutilitzable instanciada dues vegades. Cadascun fusiona fonts diferents perquè respon a una necessitat diferent.

El primer, el filtre del **yaw**, és el que fa funcionar de veritat l'odometria. El yaw és l'angle d'orientació del robot al pla, és a dir, cap a on apunta, i és el que s'utilitza per actualitzar la posició (x, y) (capítol 2). Aquest filtre fusiona la velocitat angular del giroscopi (l'eix g_z) amb l'orientació deduïda dels encoders. Tots dos observen el mateix gir però amb errors de naturalesa diferent: el giroscopi és precís a curt termini però deriva amb el temps, mentre que els encoders donen el gir directament però fallen quan una roda llisca. El filtre de Kalman els combina i n'aprofita el millor de cadascun.

Per què el yaw no es fusiona amb l'acceleròmetre? Aquí apareix la limitació clau de la IMU emprada. La MPU6050 és una IMU de sis eixos, sense magnetòmetre. L'acceleròmetre mesura la gravetat, cosa que proporciona una referència absoluta per a la inclinació (el roll i el pitch), però en girar el robot sobre l'eix vertical la projecció de la gravetat no canvia gens. Per tant, l'acceleròmetre no pot observar ni corregir la deriva del yaw, i és per això que el yaw es fusiona amb els encoders, tal com és l'enfocament habitual en robots diferencials amb una IMU de sis eixos.

El segon filtre, el del **pitch**, té una finalitat purament demostrativa. El pitch és la inclinació endavant i enrere del robot i, a diferència del yaw, sí que es pot corregir amb l'acceleròmetre, perquè la gravetat hi actua com a referència absoluta. Aquesta propietat

és precisament la que el fa idoni per ensenyar el filtre de Kalman en acció, i per això és el filtre que es mostra de manera destacada a la gràfica.

Amb aquest disseny, la visualització té sentit. Per al pitch, la gràfica superposa **tres senyals** en temps real: la referència de l'acceleròmetre, l'angle obtingut integrant únicament el giroscopi (que deriva) i el resultat filtrat pel filtre de Kalman. Les tres línies es distingeixen per color i estil, de manera que s'aprecia a cop d'ull com la del giroscopi se separa progressivament de la referència mentre que la del filtre s'hi manté enganxada: queda demostrat que el filtre corregeix alhora el soroll de l'acceleròmetre i la deriva del giroscopi. Per al yaw, en canvi, la gràfica només pot mostrar **dues línies**, la del giroscopi i la del filtre, perquè no hi ha cap referència absoluta de l'acceleròmetre amb què comparar-les.

La gràfica s'actualitza en temps real mantenint una finestra dels darrers segons, i el reinici dels angles també reinicia la gràfica. La demostració és més neta just després d'aquest reinici, ja que la deriva del giroscopi creix sense límit amb el temps i, si es deixa córrer molta estona, acaba comprimint l'escala de la gràfica. En resum, el doble filtre permet tenir alhora una estimació d'orientació correcta i defensable per a la posició (el yaw amb els encoders) i una demostració visual honesta i didàctica del filtre de Kalman (el pitch amb l'acceleròmetre), sense caure en l'error d'intentar corregir el yaw amb l'acceleròmetre, cosa que amb aquesta IMU no seria possible.

El mapa d'odometria i la visualització dels obstacles

L'altre element destacat de la interfície és el mapa d'odometria, que mostra en temps real on és el robot i per on ha passat. Es dibuixa íntegrament sobre un *canvas* HTML amb gràfics vectorials: una graella de referència, els eixos, el robot i el rastre del seu recorregut, tot pintat per codi a base de línies, punts i arcs. S'ha optat deliberadament per aquest estil de dibuix tècnic, i no per icones o imatges, perquè el conjunt tingui l'aspecte d'un instrument de mesura coherent amb la resta del quadre de comandament. El mapa treballa en metres, de manera que la posició (x, y) que arriba de l'odometria s'hi representa directament i el rastre va acumulant el camí recorregut.

Sobre aquest mateix mapa s'hi superposen els obstacles detectats pels tres sensors d'ultrasons. Cada lectura es dibuixa com un raig que surt del robot en la direcció del sensor corresponent (el frontal cap on mira el robot i els laterals a 90 graus a banda i banda) i que arriba fins a la distància mesurada. Com que els sensors mesuren en centímetres i el mapa treballa en metres, la distància es converteix abans de dibuixar-la. A més, els rajos giren amb el robot: la seva direcció es calcula a partir del yaw amb les funcions sinus i cosinus, de manera que sempre apunten cap a la posició real de l'obstacle segons cap a on està orientat el robot en cada moment.

Un detall important del disseny és que un raig només es dibuixa quan la distància mesurada és inferior a un metre. Aquesta decisió té un doble motiu. D'una banda, manté el mapa net: en un espai lliure no apareix cap raig, i només quan el robot s'acosta a una paret o a un objecte apareix el raig del costat corresponent. De l'altra, evita mostrar dades antigues, ja que en el mode automàtic els sensors laterals només es mesuren quan hi ha un obstacle al davant i la resta del temps les seves lectures podrien quedar congelades; el llindar d'un metre fa que aquestes lectures velles, que ja no corresponen a la realitat, no es dibuixin.

Cal tenir present que la interfície mostra totes aquestes dades en temps real però no les emmagatzema: en desconnectar el robot només en queda l'últim valor rebut, sense cap historial. L'única excepció és la gràfica del filtre de Kalman, que manté una finestra amb els darrers segons de senyal i permet veure'n l'evolució recent. Guardar un registre persistent de les dades en una base de dades es planteja com a treball futur al capítol 9.

7. Proves, correccions i validacions

Aquest capítol recull les proves realitzades per validar el funcionament del prototip, els problemes detectats durant aquest procés i les correccions aplicades per resoldre'ls, així com les validacions finals del sistema complet. La metodologia ha estat incremental: validar cada subsistema de manera independent abans d'integrar-lo, i bona part d'aquestes proves es van fer directament sobre el robot, abans de connectar-lo a la interfície.

7.1 Proves

Proves prèvies a la connexió amb la interfície

Abans de connectar l'ESP32-S3 a la interfície, es va comprovar el funcionament dels subsistemes bàsics treballant directament amb el robot.

La primera prova va ser la del mode automàtic, i abans calia posar a punt els sensors d'ultrasons, que van donar força feina. Els HC-SR04 funcionen a 5 V, però llavors el seu pin Echo retorna un senyal de 5 V que pot malmetre els pins de l'ESP32-S3, que només toleren 3,3 V. Com que en la fase de protoboard encara no es disposava del divisor de tensió, per precaució es van alimentar provisionalment a 3,3 V per no arriscar-se a fer malbé el microcontrolador amb el senyal d'Echo. A aquesta tensió, però, el sensor no és fiable i sovint donava una lectura constant de 999 cm, és a dir, no detectava res. També va caldre assignar correctament els pins de *trigger* i *echo* evitant els pins especials de l'ESP32-S3 (els de *strapping*, els de la memòria flash i els del USB). La solució definitiva va arribar amb el divisor de tensió integrat a la PCB (capítol 3), que permet alimentar els sensors a 5 V mantenint el senyal d'Echo en un nivell segur; amb això es van poder ajustar els llindars de detecció i validar l'algorisme d'evitació d'obstacles.

A continuació es va caracteritzar el funcionament dels encoders, també abans de la interfície. Una prova important va ser comptar els polsos generats en una volta sencera de roda. Per saber on començava i acabava una volta mentre la roda girava, es va enganxar un petit tros de cinta aïllant al costat d'una ranura, que feia de marca de referència. D'aquesta manera es va comprovar que una volta sencera no generava els 20 polsos que es podien esperar pel nombre de ranures del disc, sinó 40, ja que el sensor detecta tant la vora d'entrada com la de sortida de cada ranura. Aquest valor es va incorporar al càlcul de l'odometria.

Amb el nombre de polsos per volta ja determinat, es van fixar la resta de paràmetres mecànics necessaris per al càlcul de l'odometria de les equacions de la secció 2.2.3. El radi de les rodes és de 33 mm, de manera que la circumferència completa és de $2\pi \cdot 33 \approx 207$

mm; repartida entre els 40 polsos d'una volta, cada pols equival a un desplaçament lineal d'aproximadament 5,18 mm. Aquest valor, que anomenem metres per pols, és el que converteix el comptatge de polsos de cada encoder en el desplaçament real de cada costat del robot, és a dir, els valors Δs_e i Δs_d de l'equació (2.17). La distància entre les rodes esquerra i dreta, el paràmetre b de l'equació (2.18), és de 130 mm, i és la que permet derivar la variació d'orientació $\Delta\theta$ a partir de la diferència de desplaçament entre tots dos costats. Amb aquests tres paràmetres, el radi de roda, els metres per pols i la distància entre rodes, queda completament definit el càlcul d'odometria que alimenta la predicció del filtre de Kalman.

Tot seguit es van fer dues proves de moviment per validar l'odometria. En el moviment en línia recta, els dos encoders compten de manera pràcticament simètrica: en una de les proves, l'esquerre va comptar 61 polsos i el dret 60, amb una posició resultant coherent (un avanç net en l'eix x sense desviació apreciable en y). Això confirma que l'odometria en línia recta és fiable. En canvi, en el gir sobre el mateix lloc apareix una gran asimetria: en una prova de gir cap a la dreta, l'esquerre va comptar 78 polsos i el dret només 6. El motiu és que la roda que ha d'anar cap enrere amb prou feines gira (llisca, perquè el gir sobre l'eix és molt exigent), de manera que el seu encoder gairebé no compta. La conclusió és que l'odometria de gir basada només en els encoders no és fiable, fet que justifica directament la fusió amb el giroscopi mitjançant el filtre de Kalman, ja que el giroscopi mesura la rotació real del robot encara que les rodes patinin.

Proves un cop connectat a la interfície

Amb els subsistemes bàsics validats, es va connectar el robot a la interfície i es van comprovar la resta de funcions. Es va verificar que el mode manual responia correctament, és a dir, que el robot es teleoperava sense retards a partir de les ordres enviades des de la interfície, i es va comprovar que la unitat de mesura inercial donava els resultats esperats, tant en repòs com en moviment, amb les magnituds i els signes correctes.

7.2 Correccions

Augment de les velocitats

Una de les primeres correccions va ser augmentar les velocitats dels motors. Amb els valors inicials, el robot es movia amb dificultat i, sobretot, li costava molt girar sobre si mateix. Hi van contribuir dos factors. D'una banda, el driver L298N introdueix una caiguda de tensió notable (de l'ordre d'un parell de volts), de manera que, a valors de PWM baixos, als motors no els arribava prou tensió efectiva per vèncer la inèrcia i el fregament. De l'altra,

mentre l'ESP32 compartia l'alimentació amb els motors, el consum del microcontrolador enviant dades de manera contínua reduïa encara més la tensió disponible per als motors, que quedaven massa fluixos. La solució va ser apujar els valors de PWM, especialment el de gir, fins als valors actuals, de manera que els motors disposessin de prou parell per moure's i girar amb fiabilitat.

Alimentació: separació de l'ESP32 i els motors

El problema d'alimentació anterior va portar a una correcció més de fons. Es van diagnosticar una sèrie de fallades intermitents (caigudes de WiFi, la MPU donant lectures incoherents, el moviment que s'encallava) que finalment es van atribuir a l'alimentació. L'origen és un fenomen elèctric ben conegut: els motors de corrent continu demanen un pic de corrent molt elevat en l'instant d'arrencar i en canviar de sentit, força superior al del seu règim normal. Quan l'ESP32-S3 compartia la mateixa font que els motors, alimentant-se de les bateries a través del regulador del L298N, aquests pics de corrent provocaven una caiguda brusca de la tensió a la línia comuna.

L'ESP32 disposa d'un circuit de protecció anomenat detector de *brownout*. Un *brownout* és una baixada de la tensió d'alimentació per sota del nivell mínim de funcionament; el detector vigila aquesta tensió i, quan cau per sota d'un cert llindar, reinicia el microcontrolador per evitar que operi de manera erràtica amb una alimentació insuficient. Cada pic de corrent dels motors feia caure la tensió per sota d'aquest llindar i disparava el reinici, cosa que es manifestava com a caigudes de WiFi, pèrdua de connexió i fallades en la lectura de la IMU.

La solució va ser separar els dominis d'alimentació. L'ESP32-S3 s'alimenta ara de manera independent amb una bateria portàtil (*power bank*) de 5 V connectada per USB, que li garanteix una tensió estable al marge del consum dels motors. Els motors continuen alimentant-se amb les bateries 18650 a través del L298N. Tots dos circuits comparteixen la massa (GND comú), un detall imprescindible perquè les referències de tensió de les dues fonts siguin coherents i els senyals, com ara els polsos dels encoders, s'interpretin correctament. El jumper del regulador de 5 V del L298N es manté col·locat i els encoders s'alimenten des d'aquesta sortida de 5 V del driver. Amb aquesta separació, els pics de corrent dels motors ja no afecten l'alimentació de la part de control, i els reinicis i les caigudes de connexió van desaparèixer.

Sobreescalfament del L298N

El driver L298N s'escalfa molt durant el funcionament, i aquest escalfament afecta tots els modes de moviment, ja que en qualsevol d'ells els motors demanen corrent de manera continuada. On es fa més crític, però, és en el mode retorn: pel fet de moure's a tirons,

alternant constantment girs i avanços amb molts arrencaments, el driver demana corrent gairebé sense pausa i s'escalfa encara més. En arribar a una certa temperatura, la protecció tèrmica del driver actua i el robot s'atura abans d'arribar al seu destí. El comportament es va aconseguir millorar, però una solució completa encara està pendent.

Tècnicament, l'arrel del problema és que el L298N és un pont en H basat en transistors bipolars, amb una caiguda de tensió interna elevada; aquesta caiguda es dissipa en forma de calor, i en una maniobra exigent i contínua com el retorn el driver acaba acumulant massa temperatura. Hi ha diverses vies per resoldre-ho del tot. La primera és reduir la generació de calor, suavitzant el control perquè el moviment sigui més continu i amb menys arrencaments bruscos, i abaixant la velocitat de gir durant el retorn per limitar el corrent. La segona és millorar la dissipació, afegint un dissipador o una petita ventilació al L298N. I la més efectiva seria substituir el L298N per un driver basat en transistors MOSFET (com el TB6612FNG o el DRV8833), que tenen unes pèrdues i un escalfament molt menors per al mateix corrent.

Correcció de la deriva del giroscopi

Per millorar l'estimació d'orientació es va implementar la calibració del biaix del giroscopi en arrencar. Un giroscopi en repòs no marca exactament zero, sinó que presenta un petit biaix constant que, en integrar-se al llarg del temps, és la causa principal de la deriva. El mètode aplicat aprofita que el robot està quiet en encendre's per promitjar unes 500 lectures del giroscopi en repòs i obtenir així el biaix de cada eix, que després es resta a totes les lectures posteriors. Aquesta calibració complementa el filtre de Kalman, que ja estima i corregeix el biaix de manera adaptativa amb una de les seves variables d'estat: en partir d'un valor ja calibrat, el filtre no ha d'aprendre el biaix derivant durant els primers instants. A més, es va ajustar el paràmetre de soroll de la mesura (R) del filtre del yaw perquè confiés més en el giroscopi que en els encoders durant els girs, justament quan aquests últims són menys fiables.

7.3 Validacions

Un cop aplicades les correccions, es va validar el funcionament del sistema complet comprovant que totes les parts treballaven correctament i de manera conjunta. Per a cada comprovació es va contrastar el comportament real del robot amb el que mostrava la interfície en el mateix instant.

En primer lloc es va comprovar l'estat general del sistema amb el robot connectat i en repòs, sense rebre cap ordre, per verificar que tots els panells rebien dades i es mostraven correctament (figura 7.1).

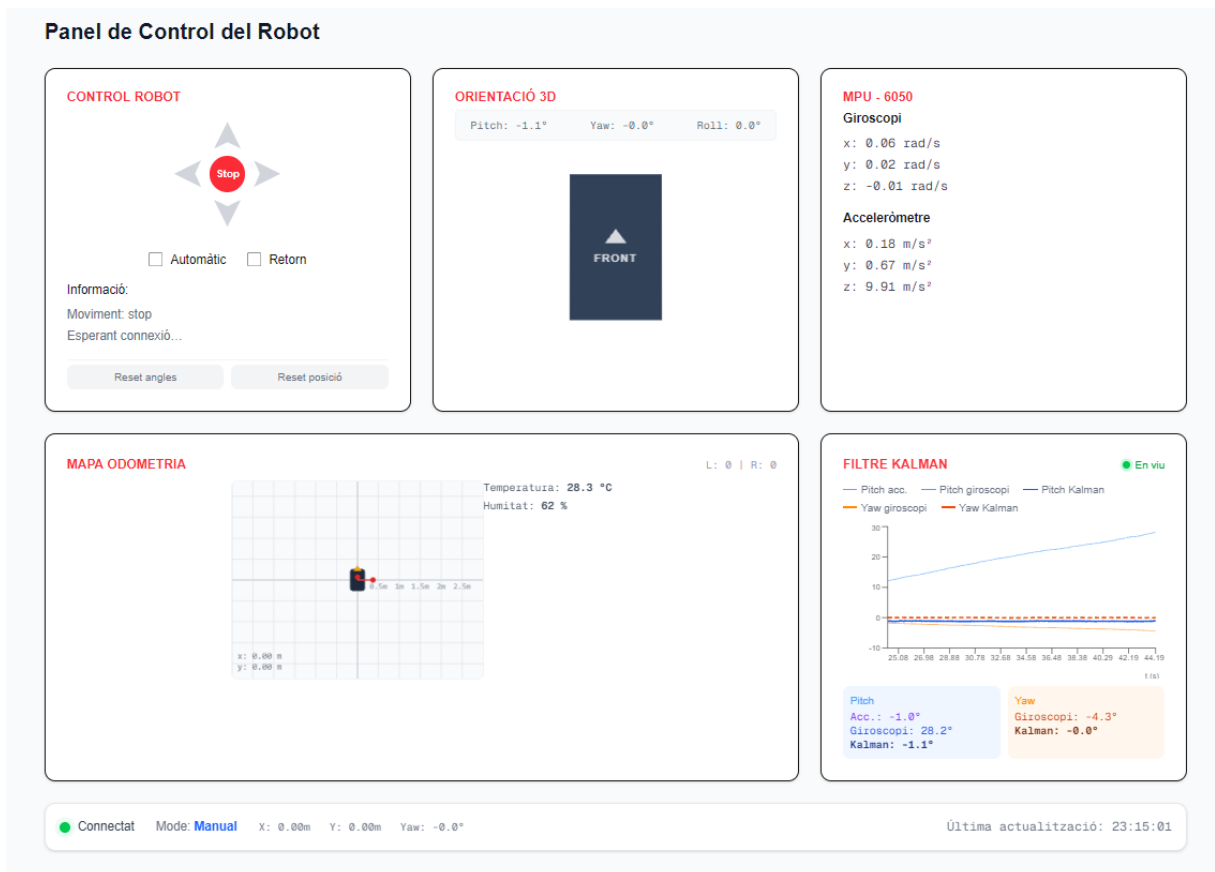


Figura 7.1: Estat general de la interfície amb el robot connectat i en repòs.

D'aquesta captura ja se'n poden extreure diverses validacions. L'acceleròmetre, amb el robot pla i en repòs, marca uns $9,91 \text{ m/s}^2$ en l'eix z i valors propers a zero en x i y , per tant, està mesurant correctament l'acceleració de la gravetat en la direcció vertical, tal com s'espera. El giroscopi, també en repòs, dona valors molt petits però no exactament nuls (de l'ordre de centèsimes de rad/s), i és precisament aquest petit biaix residual el que, integrat en el temps, provoca la deriva. Aquesta deriva es fa evident al panell del filtre de Kalman: l'angle de pitch calculat només amb el giroscopi havia derivat fins als $28,2^\circ$, mentre que el filtrat es mantenia a $-1,1^\circ$, coherent amb la referència de l'acceleròmetre. Finalment, el sensor DHT11 mostrava una temperatura de $28,3^\circ\text{C}$ i una humitat del 62 %, valors coherents amb les condicions ambientals, i tant la posició (x, y) com els comptadors dels encoders es mantien a zero perquè el robot encara no s'havia mogut.

A continuació es va validar l'orientació. Inclinant el robot cap amunt i cap avall es va comprovar que el model 3D va reflectir la inclinació real, amb un pitch de $25,5^\circ$ (figura 7.2). La gràfica permet veure l'acció del filtre: mentre el robot es movia, els senyals en cru del pitch (l'acceleròmetre, $31,8^\circ$, i el giroscopi, $41,4^\circ$) oscil·laven i diferien entre ells, mentre que el filtre en donava una estimació neta i estable ($25,5^\circ$). L'efecte es va veure encara

més clar en el yaw: tot i que el robot no havia girat, el giroscopi havia derivat fins als $-30,3^\circ$, mentre que el filtre el mantenia pràcticament a zero ($-1,4^\circ$). Això va confirmar que el filtre de Kalman corregeix la deriva del giroscopi i que les lectures de la IMU eren coherents.

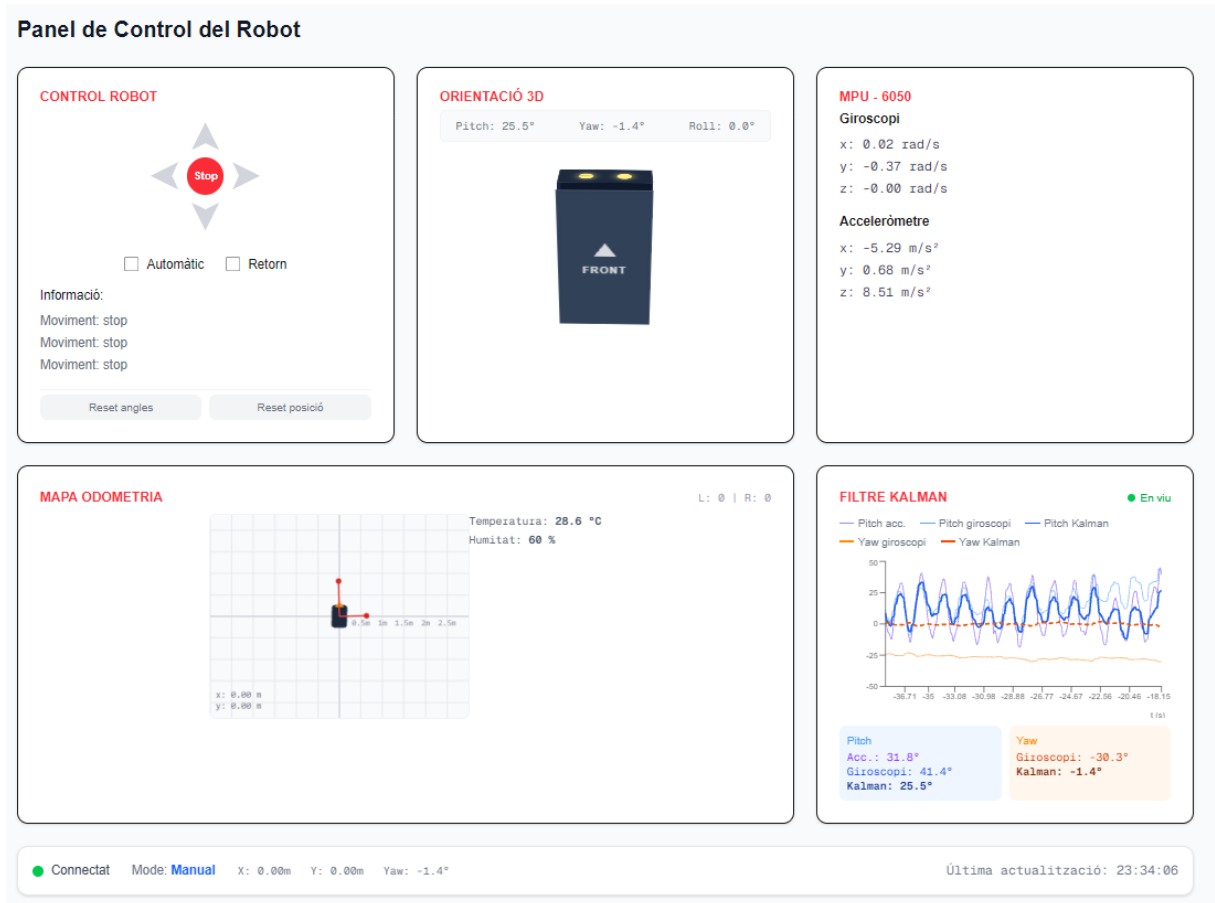


Figura 7.2: Orientació: model 3D i gràfica del filtre de Kalman.

Posteriorment, es va validar l'odometria del sistema. Es va col·locar un metre estès al terra des del punt de partida del robot per marcar una distància coneguda d'un metre (figura 7.3). A continuació es va fer avançar el robot en mode manual al llarg d'aquest recorregut i es va comprovar que la posició estimada al mapa indicava un desplaçament aproximadament equivalent al recorregut real, fet que confirma la coherència dels càlculs d'odometria i de la representació de la posició al mapa (figura 7.4).



Figura 7.3: Robot al punt de partida, amb el metre marcant la distància d'un metre a recórrer.

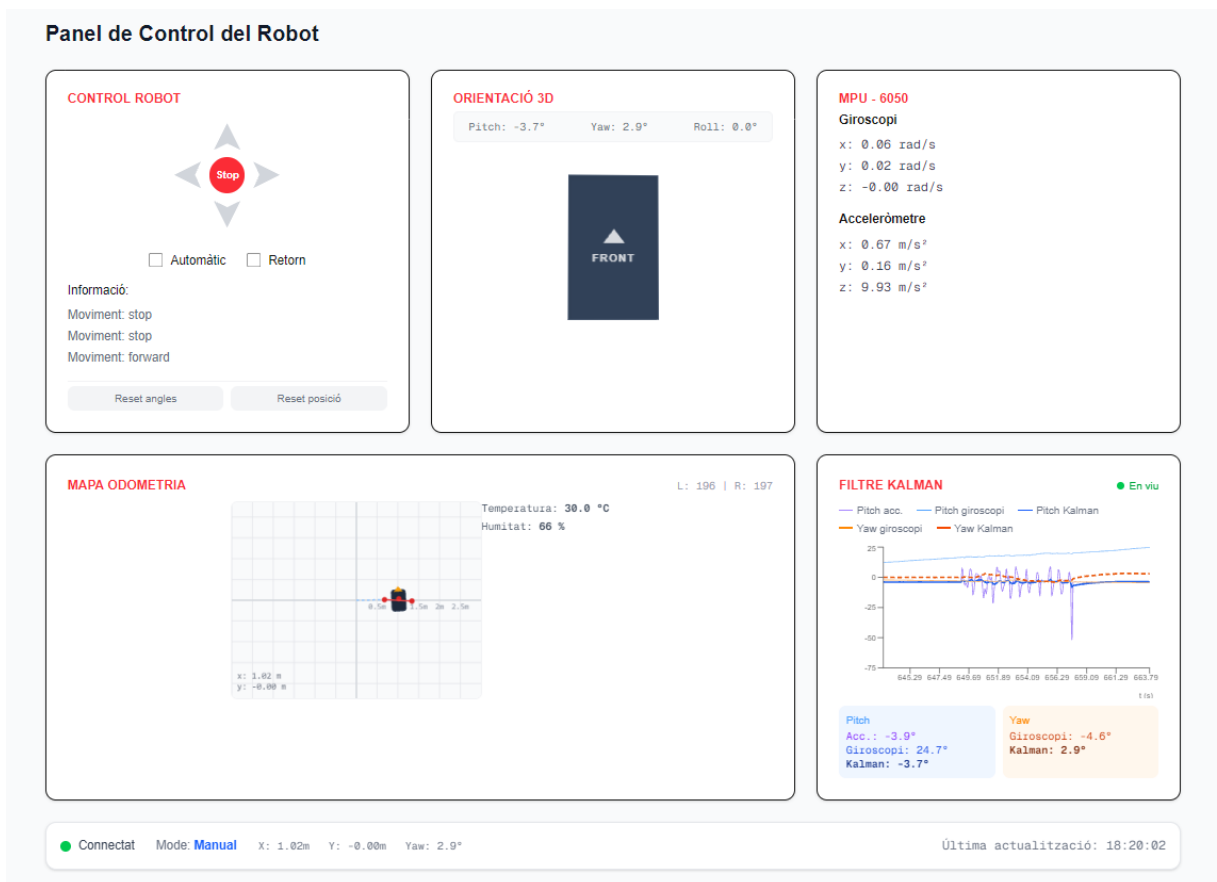


Figura 7.4: Mode manual: avanç d'un metre i posició al mapa.

Tot seguit, es va validar el funcionament del mode automàtic en un entorn amb diversos obstacles. Es va observar que el robot era capaç de detectar-los i modificar la seva trajectòria de manera autònoma per evitar col·lisions. Paral·lelament, la interfície representava en temps real els rajos associats a les mesures dels sensors d'ultrasons i actualitzava la posició estimada del robot al mapa. Aquesta prova va confirmar el correcte funciona-

ment del sistema de detecció d'obstacles, de l'algorisme d'evitació i de la representació gràfica de l'entorn. (figura 7.5).

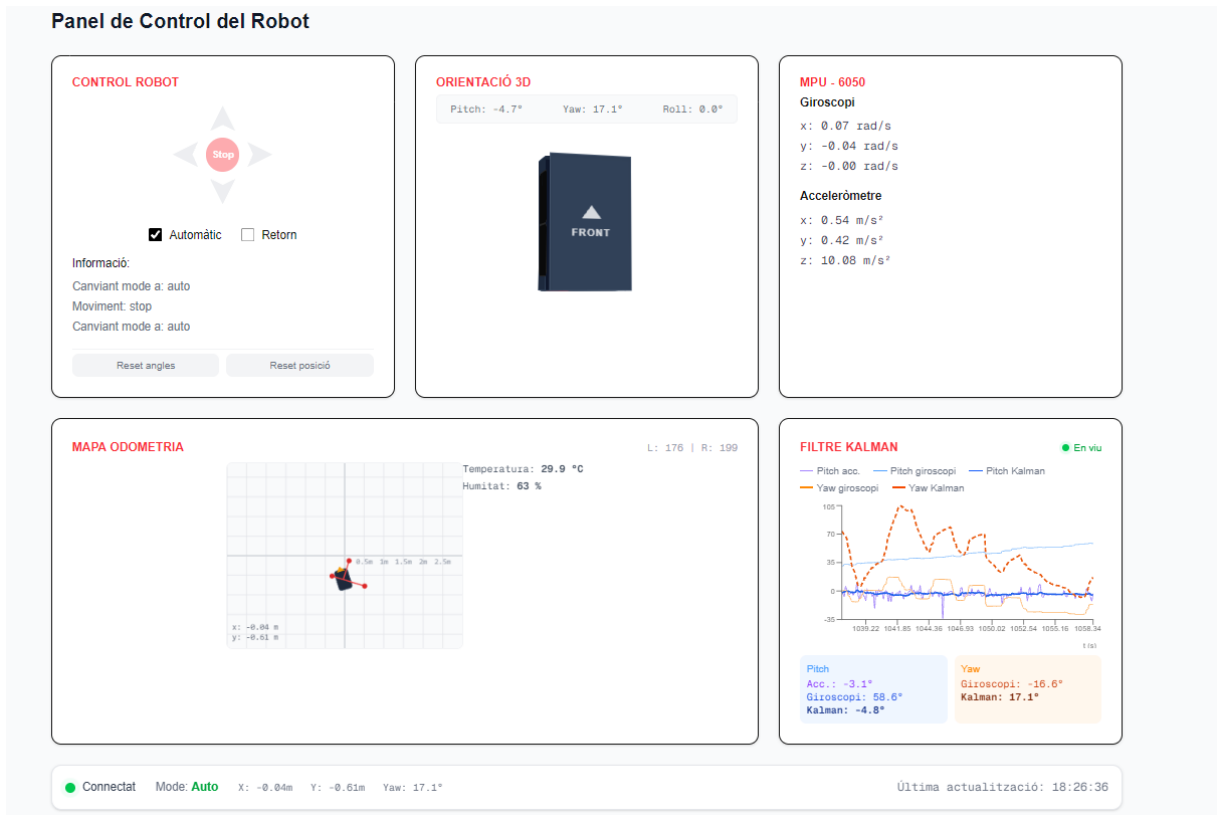


Figura 7.5: Mode automàtic: detecció i esquiva d'un obstacle.

Finalment, per validar el mode de retorn, es va allunyar prèviament el robot del punt d'origen i se'n va activar la funcionalitat. Tal com s'observa a la figura 7.6, el sistema va ser capaç de calcular la trajectòria de tornada i desplaçar-se fins a les proximitats de la posició inicial. La manca d'una coincidència exacta amb el punt de partida posa de manifest les limitacions inherents de l'odometria, un comportament esperable atès que qualsevol error en l'estimació de la posició es propaga i s'acumula al llarg del recorregut. Malgrat aquesta deriva, el mode de retorn es va considerar validat, ja que les correccions aplicades atorguen al prototip la precisió suficient per mantenir un comportament coherent i navegar de manera autònoma fins al seu entorn d'origen.

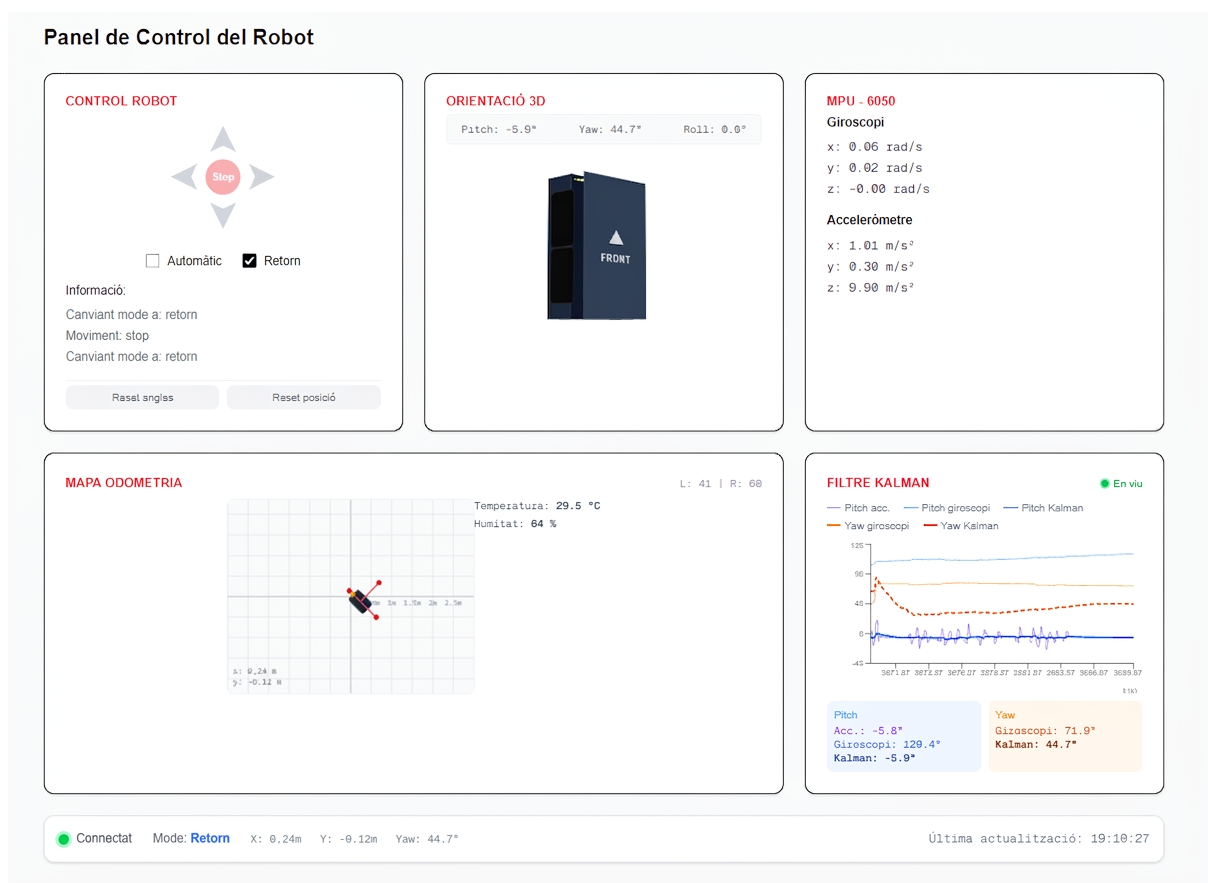


Figura 7.6: Mode retorn: tornada cap al punt d'origen.

La figura 7.7 mostra on va quedar el robot un cop finalitzat el retorn.

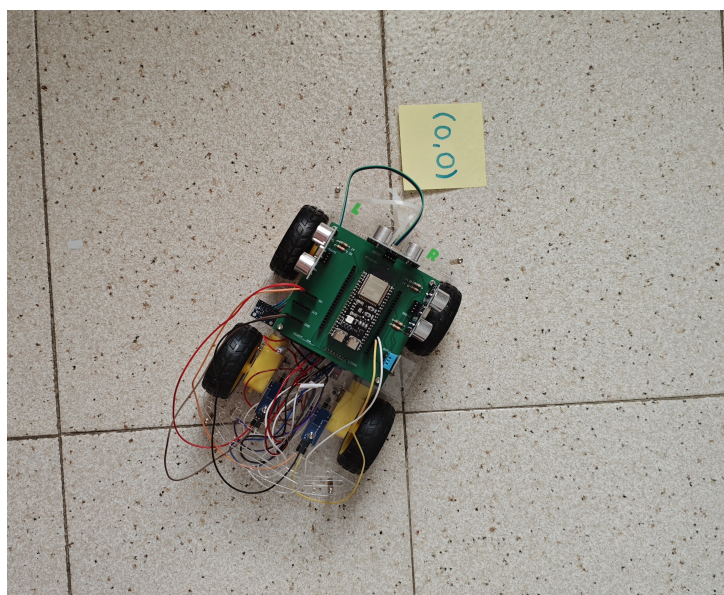


Figura 7.7: Posició final del robot un cop completat el mode retorn.

8. Conclusions

En aquest treball s'ha concebut, dissenyat i implementat un prototip de robot mòbil capaç d'estimar la seva pròpia posició i orientació sense dependre de cap sistema de posicionament de localització ni de connectivitat, un escenari representatiu de l'exploració planetària o d'entorns interiors. A la vista dels resultats de les proves i validacions del capítol anterior, es pot afirmar que el prototip compleix els objectius principals que es van marcar a la introducció: s'ha dissenyat un robot mòbil basat en l'ESP32-S3 que opera sense cap sistema de posicionament de localització ni connectivitat externa, s'ha integrat la IMU MPU6050 i s'ha implementat un filtre de Kalman que en fusiona les dades amb les dels encoders, s'ha dotat el sistema d'instrumentació ambiental i de detecció d'obstacles, s'han desenvolupat els modes manual, automàtic i de retorn, s'ha dissenyat una PCB pròpia i un xassís a mida, i s'ha establert la comunicació sense fils amb una interfície d'usuari que permet visualitzar i comparar el senyal en cru amb el filtrat.

Per una banda, s'ha construït un robot de tracció diferencial basat en l'ESP32-S3 que integra una IMU MPU6050 per I2C, dos encoders òptics per al càlcul d'odometria, tres sensors d'ultrasons per a la detecció d'obstacles i un sensor DHT11 per a la instrumentació ambiental. Sobre aquest maquinari s'ha implementat l'element central del treball: una estimació de l'orientació i la posició basada en el filtre de Kalman. En concret, s'han implementat dos filtres independents, un per al yaw, que fusiona el giroscopi amb l'odometria dels encoders i és el que sosté l'estimació real de la posició, i un per al pitch, de caràcter demostratiu, que fusiona el giroscopi amb l'acceleròmetre per il·lustrar visualment com el filtre corregeix alhora el soroll de la mesura i la deriva de la integració. Les validacions van mostrar clarament aquest efecte: amb el robot en repòs, l'angle calculat només amb el giroscopi derivava de manera progressiva mentre que l'estimació filtrada es mantenia estable i lligada a la referència física.

S'han implementat també els tres modes de funcionament previstos com a imprescindibles, manual, automàtic i retorn, accessibles des d'una interfície d'usuari desenvolupada amb Next.js que es comunica amb el robot per WiFi mitjançant WebSocket i que permet teleoperar el vehicle, monitoritzar els sensors en temps real i comparar el senyal en cru amb el senyal filtrat. A més, s'ha dissenyat i fabricat una PCB pròpia que ha substituït el muntatge inicial sobre protoboard, aportant la robustesa necessària perquè les connexions no fallin amb la vibració del moviment, i s'ha integrat tot el conjunt sobre un xassís a mida.

Pel que fa a la precisió de l'estimació, les proves van confirmar el comportament esperat de cada font. L'odometria en línia recta es va mostrar fiable, amb un desplaçament estimat coherent amb la distància real recorreguda. En els girs sobre el mateix eix, en canvi, els

encoders sols no són de fiar a causa del lliscament de les rodes, i és precisament aquesta limitació la que justifica la fusió amb el giroscopi mitjançant el filtre de Kalman. El mode retorn va aconseguir portar el robot fins a les proximitats del punt d'origen, tot i que sense una coincidència exacta: la diferència posa de manifest que qualsevol error en l'estimació de la posició s'acumula al llarg del recorregut, una limitació inherent a la navegació per estimació sense cap referència externa de correcció.

El treball també ha permès identificar amb claredat les limitacions del prototip. Algunes són físiques, com el con de detecció d'uns 15 graus dels sensors d'ultrasons, que impedeix detectar de manera fiable objectes prims, o l'escalfament del driver L298N en maniobres exigents i continuades com el retorn, que pot arribar a activar-ne la protecció tèrmica i fer que el robot s'aturi abans d'arribar al seu destí. Altres aspectes queden pendents de millora, com la calibració fina de l'odometria o l'emmagatzematge persistent de les dades, que es va plantejar com a requisit recomanable però que finalment no es va implementar i es recull com a línia de treball futur al capítol següent.

En conjunt, el resultat és un prototip funcional que serveix com a prova de concepte d'un vehicle explorador a petita escala, capaç d'estimar de manera raonablement robusta la seva posició i orientació a partir exclusivament de sensors embarcats. Més enllà del resultat tècnic, el projecte ha permès integrar en un únic sistema bona part dels coneixements adquirits al llarg del grau, des del disseny de circuits i fabricació de la PCB fins a l'adquisició i el tractament de senyal, la fusió de la capa de sensat amb el filtre de Kalman, la programació del microcontrolador i, per primera vegada, el desenvolupament d'una aplicació web per al control i la visualització en temps real.

9. Treball futur

El prototip desenvolupat compleix els objectius principals del treball, però obre la porta a diverses ampliacions que en millorarien les prestacions i que, per limitacions de temps, s'han deixat com a línies de treball futur. Algunes corresponen als requisits classificats com a *could have* en la metodologia MoSCoW del capítol 1.

La primera ampliació natural és la calibració fina de l'odometria, comparant els metres reals recorreguts amb la posició estimada per ajustar els paràmetres mecànics. Una odometria més precisa és la base sobre la qual es recolzen totes les altres millores relacionades amb el posicionament.

Pel que fa als modes de moviment, es preveu afegir un mode d'anar a coordenades, en què l'usuari indiqui un punt de destí (x, y) i el robot s'hi desplaci de manera autònoma a partir de la seva posició estimada, tal com es contemplava als requisits *could have* del projecte. En la mateixa línia, el mode retorn es podria millorar incorporant l'evitació d'obstacles durant el trajecte, combinant el comportament d'anar cap a l'objectiu amb el d'esquivar, en lloc del retorn en línia recta actual.

Una altra ampliació prevista és la incorporació d'un sensor d'ultrasons muntat sobre un servomotor, que permetria escanejar l'entorn en diferents direccions i construir un mapa de l'espai explorat, anant més enllà de la simple detecció d'obstacles. Aquesta millora depèn directament de disposar d'una odometria fiable, ja que les dades de distància només són útils si es poden situar correctament respecte a la posició del robot.

Una millora de caràcter més constructiu seria el disseny d'una carcassa a mida, impresa amb impressora 3D, que cobrís el cablejat que actualment queda a la vista. Tot i que bona part de les connexions s'han ordenat amb la PCB, encara hi ha cables inevitables, com els que alimenten els motors o els que arriben als encoders, que queden exposats. Una carcassa tancada milloraria l'acabat i la robustesa del prototip, alhora que en protegiria l'electrònica de cops i de pols durant l'exploració.

Quant a la interfície i la gestió de dades, es planteja afegir una base de dades al mateix ordinador per emmagatzemar de manera persistent les dades recollides, com ara les de la gràfica del filtre de Kalman, de manera que es puguin recuperar i analitzar després de cada sessió. També es podria enriquir la representació tridimensional de l'orientació del robot a partir de les dades de la IMU.

En conjunt, aquestes línies de treball comparteixen un mateix fil conductor: a partir d'un posicionament robust sense cap sistema de localització ni connectivitat externa, que és el nucli d'aquest treball, ampliar les capacitats del robot cap a una exploració i un mapatge cada cop més autònoms i complets.

Referències

- [1] Espressif Systems, “ESP32-S3 Series Datasheet,” Espressif Systems, inf. tèc., 2024. cons. 11 de febr. de 2026. adr.: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf
- [2] Espressif Systems. “ESP32-S3-DevKitC-1 v1.1 User Guide,” Espressif Systems, cons. 11 de febr. de 2026. adr.: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/hw-reference/esp32s3/user-guide-devkitc-1.html>
- [3] Wikipedia contributors. “Pitch, yaw, and roll,” Simple English Wikipedia. adr.: https://simple.wikipedia.org/wiki/Pitch,_yaw,_and_roll
- [4] InvenSense Inc., “MPU-6000 and MPU-6050 Product Specification, Revision 3.4,” InvenSense Inc., inf. tèc. PS-MPU-6000A-00, ag. de 2013. cons. 11 de febr. de 2026. adr.: https://www.cdiweb.com/datasheets/invensense/mpu-6050_datasheet_v3%204.pdf
- [5] J. Borenstein, H. R. Everett i L. Feng, *Where am I? Sensors and Methods for Mobile Robot Positioning*. University of Michigan, 1996. adr.: https://www.cs.cmu.edu/~motionplanning/papers/sbp_papers/integrated1/borenstein_robot_positioning.pdf
- [6] Texas Instruments, “LM393B, LM2903B, LM193, LM293, LM393 and LM2903 Dual Comparators Datasheet,” Texas Instruments, inf. tèc. SLCS005AH, Rev. AH, abr. de 2025, Comparador en què es basa el mòdul d’encoder òptic de ranura emprat al prototip. cons. 11 de febr. de 2026. adr.: <https://www.ti.com/lit/ds/symlink/lm393.pdf>
- [7] ELEC Freaks, *Ultrasonic Ranging Module HC-SR04 Datasheet*, 2011. cons. 11 de febr. de 2026. adr.: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [8] The Engineering Projects. “Introduction to HC-SR04 (Ultrasonic Sensor).” adr.: <https://www.theengineeringprojects.com/2018/10/introduction-to-hc-sr04-ultrasonic-sensor.html>
- [9] Aosong Electronics Co., Ltd., *DHT11 Humidity & Temperature Sensor Datasheet*, 2010. cons. 11 de febr. de 2026. adr.: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- [10] STMicroelectronics, *L298 Dual Full-Bridge Driver Datasheet*, Document de referència del driver de motors emprat al prototip, 2000. cons. 11 de febr. de 2026. adr.: <https://www.st.com/resource/en/datasheet/l298.pdf>
- [11] D. Nedelkovski. “Arduino DC Motor Control Tutorial – L298N | PWM | H-Bridge,” HowToMechatronics. adr.: <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>
- [12] SnapEDA. “ESP32-S3-DEVKITC-1-N16R8V: símbol, footprint i model 3D,” SnapEDA. adr.: <https://www.snapeda.com/parts/ESP32-S3-DEVKITC-1-N16R8V/Espressif%20Systems/view-part/>

- [13] I. Fette i A. Melnikov, “The WebSocket Protocol,” Internet Engineering Task Force (IETF), inf. tèc. RFC 6455, des. de 2011. DOI: [10.17487/RFC6455](https://doi.org/10.17487/RFC6455) cons. 3 d’abr. de 2026. adr.: <https://datatracker.ietf.org/doc/html/rfc6455>
- [14] Vercel. “Next.js Documentation,” Vercel Inc., cons. 3 d’abr. de 2026. adr.: <https://nextjs.org/docs>
- [15] Tailwind Labs. “Tailwind CSS Documentation: Styling with utility classes,” Tailwind Labs Inc., cons. 3 d’abr. de 2026. adr.: <https://tailwindcss.com/docs/styling-with-utility-classes>
- [16] Meta Open Source. “React: The library for web and native user interfaces,” Meta Platforms, Inc., cons. 3 d’abr. de 2026. adr.: <https://react.dev/>
- [17] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, núm. 1, pàg. 35-45, 1960. DOI: [10.1115/1.3662552](https://doi.org/10.1115/1.3662552)
- [18] R. Siegwart, I. R. Nourbakhsh i D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2a ed. Cambridge, Massachusetts: The MIT Press, 2011.
- [19] MathWorks. “Filtro de Kalman,” MathWorks. adr.: <https://es.mathworks.com/discovery/kalman-filter.html>
- [20] MathWorks. “Fundamentos de los filtros de Kalman (sèrie de vídeos),” MathWorks. adr.: <https://es.mathworks.com/videos/series/understanding-kalman-filters.html>
- [21] MathWorks. “Kalman Filtering (Control System Toolbox),” MathWorks. adr.: <https://es.mathworks.com/help/control/ug/kalman-filtering.html>
- [22] MathWorks. “Modelo IMU, GPS e INS/GPS (Navigation Toolbox),” MathWorks. adr.: <https://es.mathworks.com/help/nav/ug/model-imu-gps-and-insgps.html>
- [23] Naylamp Mechatronics. “Tutorial MPU6050, Acelerómetro y Giroscopio,” Naylamp Mechatronics. adr.: https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html