

Nil Munté Guerrero

**DETECCIÓ D'OCUPACIÓ DE SEIENTS I DETECCIÓ DE
GESTOS AMB LA MÀ MITJANÇANT EL RADAR FMCW A
77 GHz**

TREBALL DE FI DE GRAU

dirigit per Antonio Ramón Lázaro Guillén

Grau de Sistemes i Serveis de Telecomunicacions



UNIVERSITAT ROVIRA I VIRGILI

**Tarragona
2021**

AGRAÏMENTS

Amb la finalització d'aquest Treball de Fi de Grau s'acaba la meua etapa com estudiant de l'ETSE (URV). La realització d'aquest projecte no hagués estat possible sense l'ajuda i col·laboració d'un equip de professionals.

En primer lloc, m'agradaria donar les gràcies al tutor d'aquest Treball de Fi de Grau, l'Antonio Ramón Lázaro Guillén. Ell ha estat qui ha confiat amb mi per la realització d'aquest projecte, ha estat un referent a l'hora de treballar i qui m'ha guiat i encaminat en l'execució d'aquest treball. Gràcies al seu assessorament he pogut dur-lo a terme i, sobretot, introduir-me en el món de la investigació.

En segon lloc, a tots els components del grup d'investigació NEPHOS. Gràcies al seu recolzament i ajuda en el moment que ha estat necessari he pogut tirar endavant amb la realització del projecte.

També m'agradaria donar les gràcies a l'URV per la seva proximitat i per haver-me facilitat els mitjans tècnics i logístics necessaris pel desenvolupament del TFG.

Per acabar, a la meua família per donar-me ànims i suport moral en tot moment.

RESUM EXECUTIU

(Català) Aquest Treball de Fi de Grau es basa en l'estudi i la utilització del RADAR FMCW (*Frequency Modulated Continuous Wave*) AWR1642 de *Texas Instruments* pel desenvolupament de dues aplicacions en la banda mil·limètrica de 77 GHz. Les dues aplicacions desenvolupades han estat: (i) detecció del nombre d'ocupants a l'interior d'un vehicle de quatre i cinc places i determinació de la localització d'aquests usuaris, (ii) reconeixement de gestos amb la mà mitjançant una xarxa neuronal convolucional (CNN). Ambdues aplicacions s'han realitzat amb el programa *Matlab* i, en concret, l'aplicació (i) s'ha optimitzat també per *Raspberry Pi* a través de la programació amb el llenguatge *Python*.

(Español) Este Trabajo de Fin de Grado se basa en el estudio y la utilización del RADAR FMCW (*Frequency Modulated Continuous Wave*) AWR1642 de *Texas Instruments* para el desarrollo de dos aplicaciones en la banda milimétrica de 77 GHz. Las dos aplicaciones desarrolladas han sido: (i) detección del número de ocupantes en el interior de un vehículo de cuatro y cinco plazas y determinación de la localización de estos usuarios, (ii) reconocimiento de gestos con la mano mediante una red neuronal convolucional (CNN). Ambas aplicaciones se han realizado con el programa *Matlab* y, en concreto, la aplicación (i) se ha optimizado también para *Raspberry Pi* mediante la programación con el lenguaje *Python*.

(English) This Final Degree Project is based on the study and use of the FMCW (*Frequency Modulated Continuous Wave*) AWR1642 RADAR of *Texas Instruments* for the development of two applications in the 77 GHz millimeter band. The two applications developed have been: (i) detection of the number of occupants inside a four and five-seat vehicle and determination of the location of these users, (ii) hand gesture recognition using a Convolutional Neural Network (CNN). Both applications have been carried out with the program *Matlab* and, in particular, the application (i) has also been optimized for *Raspberry Pi* by programming with *Python*.

PARAULES CLAU

mmWave RADAR. Dispositiu radar que emet a la banda mil·limètrica (*millimeter-wave* o *mmWave band*). Tot i que aquesta banda té un rang dels 30 GHz als 300 GHz, aproximadament, els RADARS que operen en aquesta banda treballen als 24 GHz i 77 GHz (l'utilitzat en aquest projecte treballa en la darrera freqüència).

FMCW. Un radar FMCW (*Frequency Modulated Continuous Wave*) és un tipus de sensor radar que emet una senyal de transmissió contínua (CW, *Continuous Wave*) i que pot canviar la seva freqüència de funcionament durant la mesura, és a dir, que la senyal transmesa està modulada en freqüència.

Xarxa Neuronal Convolucional (CNN). Una Xarxa Neuronal Convolucional (de l'anglès, *Convolutional Neural Network*, CNN) és un tipus de xarxa neuronal artificial que té un ús molt comú en aplicacions d'anàlisi d'imatges. Aquest anàlisi inclou una fase d'entrenament i una fase de *testeig* de la xarxa per poder establir un reconeixement artificial mitjançant la CNN.

Detecció d'ocupació. En aquest projecte, és un sistema basat en la detecció del nombre d'ocupants a l'interior d'un vehicle i la detecció de la seva posició mitjançant el RADAR FMCW.

Reconeixement de gestos. En aquest projecte, és un sistema basat en el reconeixement de gestos amb la mà mitjançant el RADAR FMCW i una CNN.

LLISTA DE SÍMBOLS

λ	Longitud d'ona
Φ_0	Fase inicial de la senyal IF
τ	Temps de propagació d'una ona
B	Ample de banda
c	Velocitat de la llum
f_c	Freqüència inicial de la senyal <i>chirp</i>
F_s	Freqüència de mostreig
R	Rang o distància fins un objecte
R_{max}	Rang màxim detectable per un radar
S	Pendent de la freqüència del senyal <i>chirp</i>
T	Període d'una ona
T_f	Període de trama o <i>frame</i>

T_c	Període de <i>chirp</i>
v, V	Velocitat d'un l'objecte
v_{so}	Velocitat del so
l	Distància entre antenes
θ_s	Angle de moviment de la font <i>Doppler</i>
θ_o	Angle de moviment de l'observador <i>Doppler</i>

LLISTA D'ACRÒNIMS

RADAR RADio Detection and Ranging

LIDAR Laser Imaging Detection And Ranging

FMWC Frequency Modulated Continuous Wave

CLI Command Line Interface

HF High Frequency

Tx Transmissió

Rx Recepció

RF Radiofreqüència

ADC Convertidor Analògic-Digital

EM Electromagètic

LoS Line of Sight

EKF Extended Kalman Filter

AAC Control de Creuer Adaptatiu

DNN Deep Neural Networks

CNN Convolutional Neural Network

DSP Digital Signal Processor

FFT Fast Fourier Transform

ISM Industrial, Scientific and Mechanical

UWB Ultra Wide Band

IR-UWB Impulse Radio Ultra-WideBand

FOV Field Of View

TLV Tag Length Value

FCC Federal Communications Commission

DBF Digital BeamForming

IC Integrated circuit

2D-FFT Bidimensional Fast Fourier Transform

FDTW Fusion Dynamic Time Warping

HCI Human-Computer Interaction

HGR Human Gesture Recognition

CW Continuous-Wave

SFCW Single-Frequency Continuous-Wave

ADAS Advanced Driving Assistant System

ETSE Escola Tècnica Superior d'Enginyeria

URV Universitat Rovira i Virgili

INTRODUCCIÓ.....	1
1.1. Marc de treball.	1
1.2. Motivació del projecte.....	1
1.3. Objectius del projecte.....	1
1.4. Estructura del document.	2
EL RADAR FMCW	3
2.1. Principis bàsics de l'efecte doppler.	3
2.1.1. Efecte Doppler.....	3
2.1.2. Efecte Doppler relatiu i dilatació de temps	4
2.1.3. Efecte Doppler en RADARs	7
2.2. La tecnologia RADAR.....	10
2.2.1. El RADAR en el sector d'automoció.....	12
2.3. Fonaments teòrics dels sensors FMCW.....	16
2.3.1. Paràmetres de chirp.	18
2.3.1.1. Rang màxim.....	18
2.3.1.2. Resolució de rang.	19
2.3.1.3. Mesura de la velocitat.....	19
2.3.1.4. Resolució de velocitat.....	20
2.3.1.5. Detecció d'angle.....	20
2.3.1.6. Camp de visió angular màxim.	21
2.4. RADAR FMCW mmWave AWR1642 de Texas Instruments.	22
2.4.1. Descripció del hardware.	22
2.4.2. Comunicació i configuració dels paràmetres d'entrada.	25
2.4.3. Format de les dades de sortida.....	25
2.4.4. Distribució dels eixos en el RADAR.....	26
2.4.5. mmWave Demo Visualizer.	27
2.4.5.1. Paràmetres del Fitxer de Configuració.	32
2.5. Adquisició i representació de dades amb Matlab.....	39
2.5.1. Programa RADARSetup18XX.m.....	40
2.5.2. Programa readAndParseData18XX.m.....	40

2.5.3. Programa objectComponentsExists.m.....	42
2.5.4. Programa principal.	42
2.5.5. Representació de les dades	43
2.6. Referències.	46
DETECCIÓ DE L'OCUPACIÓ EN VEHICLES	47
3.1. Estat de l'art.	47
3.2. Processat del senyal.	49
3.3. Anàlisi dels resultats obtinguts.....	50
3.3.1. Definició de les zones del vehicle	51
3.3.2. Càlcul de la resposta amb Matlab.....	55
3.3.3. Presentació i anàlisi dels resultats obtinguts.	56
3.3.3.1. Programa recognition.m.	57
3.3.3.2. Programes dadesFuncioTemps.m i dadesFuncioTempsReal.m.	65
3.3.3.3. Programa createConfusionMatrix.m.	69
3.4. App interfície gràfica.	74
3.5. Processat amb RaspberryPi i Python.	76
3.6. Referències	77
SISTEMA DE DETECCIÓ DE GESTOS AMB LA MÀ BASAT EN RADAR FMCW EN LA BANDA MIL·LIMÈTRICA.....	78
4.1. Estat de l'art.....	78
4.2. Sistema de detecció de gestos utilitzant xarxes neuronals convolucionals.	81
4.2.1. Fase d'adquisició.....	87
4.2.2. Fase d'entrenament i prova	90
4.2.2.1. Càrrega de les imatges.....	91
4.2.2.2. Definició de l'arquitectura de xarxa.....	92
4.2.2.3. Especificació de les opcions de training.....	94
4.2.2.4. Entrenament i obtenció de la xarxa.	94
4.2.2.5. Testeig de la xarxa.....	95
4.2.3. Correlació de les imatges dels gestos realitzats.....	98
4.3. Referències.	100

CONCLUSIONS I LÍNIES FUTURES	101
5.1. Detecció de l'ocupació en vehicles.	101
5.2. Sistema de detecció de gestos amb la mà basat en RADAR FMCW en la banda mil·limètrica.	102
5.3. Conclusions generals i línies futures.	102
5.4. Referències.	103

CAPÍTOL I

INTRODUCCIÓ

En la introducció d'aquest projecte es presentarà breument el marc i la motivació del projecte, així com la seva justificació i abast, per concloure amb l'estructura del document.

1.1. Marc de treball.

El projecte s'ha dut a terme amb el grup de recerca NEPHOS, dins del Departament d'Enginyeria Electrònica, Elèctrica i Automàtica de l'Escola Tècnica Superior d'Enginyeria (ETSE) de la Universitat Rovira i Virgili (URV).

El projecte rep el codi RTI2018-096019-B-C31 (MCIN/AEI/FEDER), amb el nom "*Nuevas Soluciones para la próxima generación de dispositivos y sensores IOT*", el qual està enfocat en el desenvolupament de software d'adquisició i processament de dades a partir d'un RADAR FMCW de banda mil·limètrica en aplicacions *wireless* en l'interior de vehicles. Es pretén realitzar dues aplicacions. La primera consisteix en un sistema d'ocupació i detecció d'usuaris a l'interior d'un vehicle de quatre i cinc places. La segona es tracta d'un sistema de detecció i reconeixement de gestos que també pot ser útil en vehicles per realitzar diverses accions sobre aquests, per així millorar la seguretat i comoditat dels usuaris i que el vehicle pugui actuar en concordança a les dades rebudes pel RADAR.

1.2. Motivació del projecte.

El cas d'ús descrit en l'apartat anterior requereix una solució per la monitorització d'usuaris robusta i intel·ligent i, a la vegada, ràpida i econòmica, per tal que el projecte tingui viabilitat davant a l'actual solució implementada. En aquest projecte es proposa un sistema RADAR intel·ligent capaç d'adquirir, processar i obtenir una resposta en un temps instantani.

El paper principal del RADAR en el sistema de detecció d'objectes, tant persones com vehicles, és la robustesa, sent aquest un dispositiu que no és capaç de donar informació més enllà de la posició, velocitat i acceleració, però amb un grau de confiabilitat superior al d'una càmera o, fins i tot, al d'un *Laser Imaging Detection and Ranging* (LIDAR) i amb una variabilitat en els seus resultats molt menor davant a condicions meteorològiques adverses.

Un altre avantatge del dispositiu és el seu baix consum i la seva autonomia a l'integrar capacitat de processament.

1.3. Objectius del projecte.

El projecte té com a principals objectius els següents:

- Estudiar l'adequada configuració dels paràmetres del RADAR FMCW per adaptar-lo al perímetre i tipus d'obstacles (usuaris, mà) que es pretén detectar.
- Desenvolupament de firmware per RADAR de banda mil·limètrica.
- Processament de senyal obtingut a partir del RADAR FMCW.
- Desenvolupament d'una aplicació de detecció i ocupació d'usuaris a l'interior d'un vehicle per millorar el sistema implementat en vehicles actual.

- Desenvolupament d'una aplicació de detecció de gestos per interactuar i realitzar diverses accions sobre el vehicle, tot millorant la comoditat del conductor del vehicle i evitar-ne la seva distracció.

1.4. Estructura del document.

La memòria d'aquest projecte es divideix en 5 capítols. L'estructura que es desenvoluparà és la següent:

Capítol I. Introducció. Es defineix el marc, justificació i objectius del projecte a desenvolupar.

Capítol II. RADAR FMCW. Exposició dels conceptes teòrics necessaris per la correcta comprensió del problema abordat. Descripció del hardware i software sobre el què es realitza la implementació (*Texas Instruments AWR1642*). Explicació sobre com es realitza l'adquisició i representació de dades mitjançant el RADAR.

Capítol III. Detecció de l'ocupació en vehicles. Aborda l'estat actual de les investigacions sobre aquesta aplicació (Estat de l'art). Explicació de l'aplicació sobre la detecció i reconeixement d'usuaris a l'interior del vehicle mitjançant el RADAR, *Matlab*, *Python* i *Raspberry Pi*. Es detallen els passos seguits en l'adaptació del codi del RADAR i es descriu el codi implementat. Presentació dels resultats obtinguts.

Capítol IV. Reconeixement de gestos amb la mà. Aborda l'estat actual de les investigacions sobre aquesta aplicació (Estat de l'art). Explicació de l'aplicació sobre la detecció i reconeixement de gestos amb la mà mitjançant el RADAR i *Matlab*. Es detallen els passos seguits en l'adaptació del codi del RADAR i es descriu el codi implementat. Presentació dels resultats obtinguts.

Capítol V. Conclusions i línies futures. Anàlisi de les proves realitzades, resultats i viabilitat pel projecte. Definició de les possibles línies de treball futures.

CAPÍTOL II

EL RADAR FMCW

En aquest capítol, inicialment, s'exposaran els principis bàsics del sistema RADAR i de l'efecte doppler, així com algunes aplicacions recurrents i l'impacte del RADAR en el mercat actual. Seguidament, es presentaran els fonaments teòrics dels sensors *FMWC* i en concret es descriurà el sistema hardware i configuració software del RADAR utilitzat, el RADAR *AWR1642* de *Texas Instruments*. Finalment, es detallarà la manera d'obtenir les dades d'aquest RADAR i poder-les tractar amb programes de tractament de dades, com ara *Matlab*.

2.1. Principis bàsics de l'efecte doppler.

2.1.1. Efecte Doppler.

En un RADAR monoestàtic, on el transmissor i receptor estan situats en el mateix lloc, el RADAR transmet un senyal electromagnètic (EM) a un objecte i rep una senyal reflectida de l'objecte. El RADAR pot mesurar el rang de l'objecte a partir del retard de temps del senyal rebut. Si l'objecte està en moviment, la freqüència del senyal rebut es desplaçarà de la freqüència de la senyal transmesa, el que es coneix com efecte Doppler [1]. La *Doppler shifted frequency* es determina a partir de la velocitat radial de l'objecte en moviment, és a dir, la component de velocitat en la direcció de la *Line of Sight del RADAR* (LoS). Basat en el desplaçament de freqüència Doppler del senyal rebut, el RADAR pot mesurar la velocitat radial de l'objecte en moviment.

L'efecte *Doppler* va ser descobert pel matemàtic i físic Christian Johann Doppler i descriu les següents fórmules a la *Biographical Encyclopedia of Scientists*.

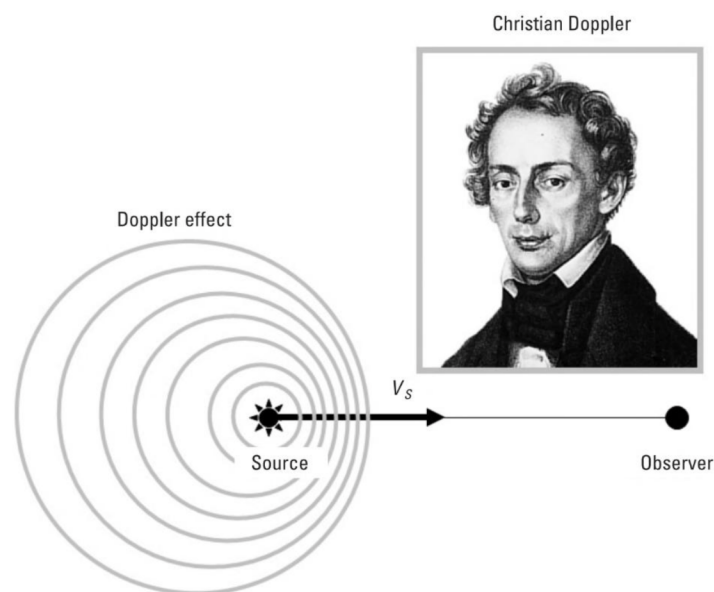


Figura 2.1. L'any 1842, Christian Doppler va descobrir el fenomen on el color aparent d'una llum d'una font canviava pel seu moviment, conegut com l'efecte Doppler.

La longitud d'ona d'una font de so es defineix com:

$$\lambda = \frac{v_{so}}{f} \quad (2.1)$$

On v_{so} és la velocitat de propagació de l'ona de so en un medi donat i f és la freqüència de la font de so. Si la font es mou a una velocitat v , relativa al medi, la freqüència percebuda per l'observador és:

$$f' = \frac{v_{so}}{v_{so} \mp v} \cdot f = \frac{1}{1 \mp \frac{v}{v_{so}}} \cdot f \quad (2.2)$$

Si $v/v_{so} \ll 1$, la *Doppler shifted frequency* percebuda per l'observador és, aproximadament:

$$f' = \frac{1}{1 \mp \frac{v}{v_{so}}} \cdot f \cong \left(1 \pm \frac{v}{v_{so}}\right) \cdot f \quad (2.3)$$

Quan la font s'apropi a l'observador s'agafarà el signe de dalt, mentre que si la font s'allunya de l'observador s'agafarà el signe de baix.

Si la font és estacionària i l'observador es mou amb una velocitat v_0 relativa al medi, la freqüència percebuda per l'observador és:

$$f' = \frac{v_{so} \pm v_0}{v_{so}} \cdot f = \left(1 \pm \frac{v_0}{v_{so}}\right) \cdot f \quad (2.4)$$

Quan l'observador s'apropi a la font s'agafarà el signe de dalt, mentre que si l'observador s'allunya de la font s'agafarà el signe de baix.

Si tant la font com l'observador s'estan movent, la freqüència percebuda per l'observador és:

$$f' = \frac{v_{so} \pm v_0}{v_{so} \mp v} \cdot f = \frac{1 \pm \frac{v_0}{v_{so}}}{1 \mp \frac{v}{v_{so}}} \cdot f \quad (2.5)$$

Quan la font i l'observador es mouen l'un cap a l'altre, s'agafa el signe de dalt de la fórmula anterior (2.5). En cas contrari, quan es mouen en sentits oposats, s'agafa el signe de sota.

2.1.2. Efecte Doppler relatiu i dilatació de temps.

Comparat amb ones de so, no hi ha medi involucrat en la llum o l'ona de propagació EM. La velocitat de propagació de la llum o de les ones EM, c , vist des de la font i de l'observador és la mateixa constant.

L'efecte relatiu Doppler és diferent del clàssic efecte Doppler perquè inclou l'efecte de la dilatació del temps de la relativitat espacial i no involucra el medi de l'ona de propagació com a punt de referència. Quan la llum o la font EM a una freqüència f es mou a una velocitat v_s i angle θ_s , relatiu a la direcció des de la font S a l'observador O tal com es pot veure a la Figura 2.2, l'interval de temps entre dues crestes successives de l'ona emesa a t_1 i t_2 es determina per:

$$\Delta t_s = t_2 - t_1 = \frac{\gamma}{f} \quad (2.6)$$

On:

$$\gamma = \frac{1}{\sqrt{1 - \frac{v_s^2}{c^2}}} \quad (2.7)$$

És un factor que representa la dilatació de temps relativa i c és la velocitat de propagació de la llum o de l'ona EM.

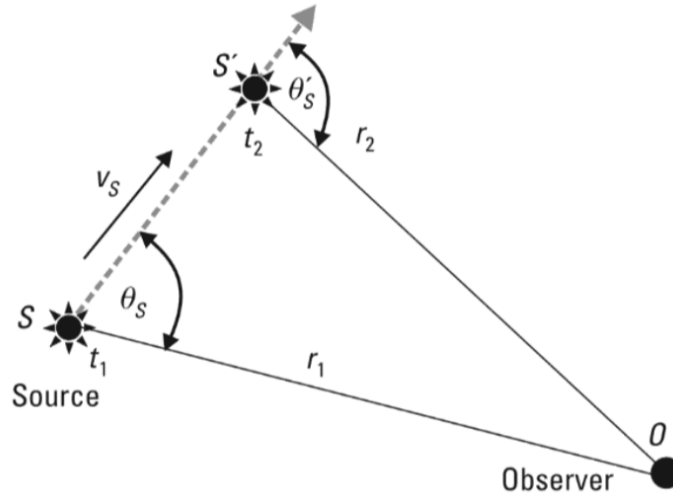


Figura 2.2. L'efecte Doppler en el cas que l'única font S es mogui amb una velocitat v_s i angle θ_s relatiu a la direcció des de la font S a l'observador O .

D'aquesta manera, l'interval de temps entre l'arribada de dos crestes d'ona successives a l'observador és:

$$\Delta t_o = \left(t_2 + \frac{r_2}{c}\right) - \left(t_1 + \frac{r_1}{c}\right) = \frac{\gamma}{f} \left(1 - \frac{v_s \cdot \cos \theta_s}{c}\right) \quad (2.8)$$

I la freqüència observada és:

$$f' = \frac{1}{\Delta t_o} = \frac{1}{\gamma} \cdot \frac{f}{1 - \frac{v_s \cdot \cos \theta_s}{c}} \quad (2.9)$$

Si tant l'observador com la font es mouen com es pot observar en la Figura en el cas 2-D, la freqüència observada a l'instant de temps quan l'ona és emesa és:

$$f' = \frac{1}{\gamma} \cdot \frac{1 \pm \frac{v_o \cdot \cos \theta_o}{c}}{1 \mp \frac{v_s \cdot \cos \theta_s}{c}} \quad (2.10)$$

On θ_s i θ_o , tal com es pot veure en la Figura 2.3, són l'angle de moviment de la font i l'angle de moviment de l'observador en el temps que l'ona s'emet, respectivament.

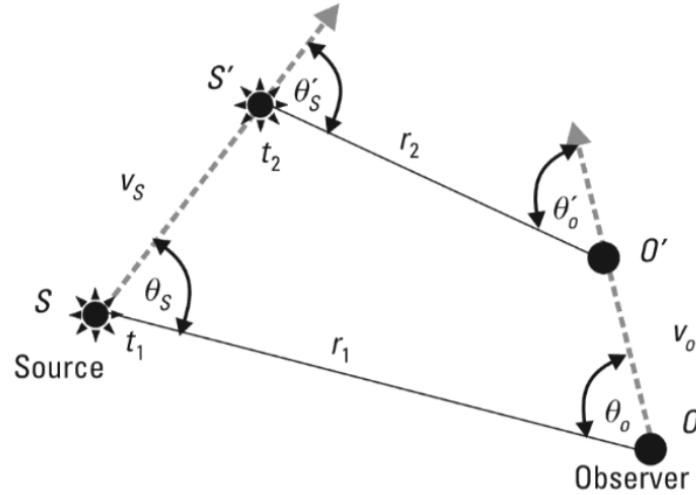


Figura 2.3. L'efecte Doppler en el cas que tant la font com l'observador es moguin

En general, donat el moviment relatiu entre la font i l'observador v , quan la font i l'observador es mouen l'un cap a l'altre, l'observada *Doppler shifted frequency* es pot reescriure com:

$$f' = \frac{1}{\gamma} \cdot \frac{f}{1 - \frac{v}{c}} = \sqrt{1 - \left(\frac{v}{c}\right)^2} \cdot \frac{f}{1 - \frac{v}{c}} = \sqrt{\frac{1 + \frac{v}{c}}{1 - \frac{v}{c}}} \cdot f \quad (2.11)$$

Si la font s'allunya de l'observador, la freqüència observada és:

$$f' = \sqrt{\frac{1 - \frac{v}{c}}{1 + \frac{v}{c}}} \cdot f \quad (2.12)$$

Si la velocitat v és molt més petita que la velocitat de propagació c de l'ona EM, és a dir, $v \ll c$, o $v/c \approx 0$, la freqüència Doppler relativa és la mateixa que la freqüència Doppler clàssica. D'acord amb la sèrie de McLaurin:

$$\sqrt{\frac{1 - \frac{v}{c}}{1 + \frac{v}{c}}} = 1 - \frac{v}{c} + \frac{\left(\frac{v}{c}\right)^2}{2} - \dots \quad (2.13)$$

Quan la font i l'observador s'allunyen l'un de l'altre, la *Doppler shifted frequency* es pot aproximar a:

$$f' \cong \left(1 - \frac{v}{c}\right) \cdot f \quad (2.14)$$

Que és la mateixa que la clàssica *Doppler frequency shift*. A més, la *Doppler frequency shift* entre la freqüència emesa de la font i la freqüència percebuda per l'observador és:

$$f_D \cong f' - f = -\frac{v}{c} \cdot f \quad (2.15)$$

La *Doppler frequency shift* és proporcional a la freqüència f emesa per la font d'ona i la velocitat relativa v entre la font i l'observador.

2.1.3. Efecte Doppler en RADARS.

En un RADAR, la velocitat de l'objectiu, v , és normalment molt més lenta que la velocitat de propagació de l'ona EM, c , és a dir, que $v \ll c$ o $\beta = v/c \approx 0$. En sistemes RADAR monoestàtics, la distància d'anada i tornada (*round-trip distance*) recorreguda per l'ona EM és el doble de la distància entre el transmissor i l'objectiu. En aquest cas, el moviment de l'ona consisteix en dos segments: viatjar del transmissor a l'objectiu, que produeix un *Doppler shift* ($-fv/c$), i viatjar de l'objectiu al receptor, que produeix un altre *Doppler shift* ($-fv/c$), on f és la freqüència de transmissió. Així doncs, la total *Doppler shift* acaba sent:

$$f_D = -f \cdot \frac{2v}{c} \quad (2.16)$$

Si el RADAR és estacionari, v serà la velocitat radial de l'objectiu al voltant de la LoS del RADAR. La velocitat es defineix com a positiva quan l'objectiu s'allunya del RADAR. Com a conseqüència, la *Doppler shift* és negativa.

En un sistema RADAR biestàtic, tal com es pot veure en la Figura 2.4, el transmissor i receptor estan separats per una línia de base L , que és comparable amb el rang màxim de l'objectiu respecte el transmissor i receptor. El rang del transmissor al receptor és donat per un vector, r_T , i el rang del receptor a l'objectiu és donat per un altre vector, r_R . L'angle biestàtic β és definit per l'angle entre la línia de transmissió-objectiu i la línia de recepció-objectiu. L'angle de visió de transmissió és α_T i pel receptor és α_R .

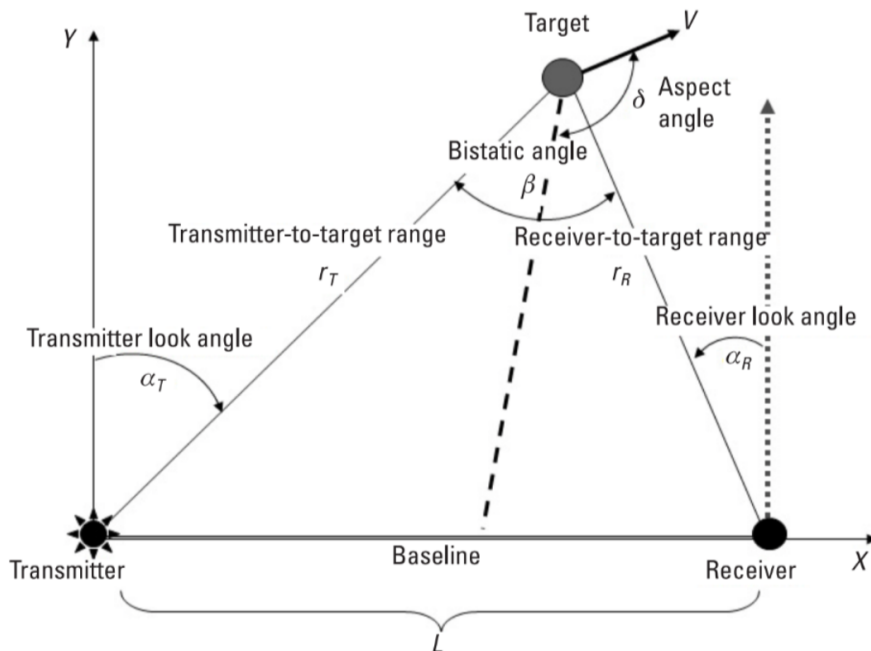


Figura 2.4. La configuració 2-D del sistema RADAR biestàtic.

L'angle de visió es defineix per l'angle des del corresponent vector de referència perpendicular a la línia base del transmissor-receptor a la LoS del vector de l'objectiu. L'angle positiu es defineix en sentit antihorari.

D'aquí, l'angle biestàtic és:

$$\beta = \alpha_R - \alpha_T \quad (2.17)$$

Si la distància des del transmissor a l'objectiu és coneix com $r_T = |r_T|$, la distància des del receptor a l'objectiu és:

$$r_R = |r_R| = \sqrt{L^2 + r_T^2 - 2r_T L \sin \alpha_T} \quad (2.18)$$

I l'angle vist des de la recepció és:

$$\alpha_R = \tan^{-1} \left(\frac{L - r_T \cdot \sin \alpha_T}{r_T \cdot \cos \alpha_T} \right) \quad (2.19)$$

Quan l'objectiu s'està movent a una velocitat de vector V , la component al voltant de la direcció LoS des del transmissor a l'objectiu és:

$$v_T = V \cdot \frac{r_T}{|r_T|} \quad (2.20)$$

I la component al voltant de la direcció LoS des del receptor a l'objectiu és:

$$v_R = V \cdot \frac{r_R}{|r_R|} \quad (2.21)$$

Llavors, degut al moviment de l'objectiu, el rang des del transmissor a l'objectiu es defineix en funció del temps:

$$r_T(t) = r_T(t=0) + v_T t \quad (2.22)$$

I el rang des del receptor a l'objectiu també és en funció del temps:

$$r_R(t) = r_R(t=0) + v_R t \quad (2.23)$$

El canvi de fase entre el senyal transmès i rebut és en funció de la longitud d'ona del RADAR $\lambda = c/f$, la distància des del transmissor a l'objectiu $r_T(t)$, i la distància des de l'objectiu al receptor $r_R(t)$:

$$\Delta\phi(t) = \frac{r_T(t) + r_R(t)}{\lambda} \quad (2.24)$$

Llavors, la *Doppler frequency shift* és mesurada per la taxa del canvi de fase. Prenent el temps derivatiu del canvi de fase (2.24), la *bistatic Doppler frequency shift* és:

$$f_{D_{Bi}} = \frac{1}{2\pi} \cdot \frac{d}{dt} \Delta\phi(t) = \frac{1}{2\pi} \cdot \frac{1}{\lambda} \left[\frac{d}{dt} r_T(t) + \frac{d}{dt} r_R(t) \right] = \frac{1}{2\pi} \cdot \frac{1}{\lambda} \cdot (v_T + v_R) \quad (2.25)$$

Per seguir el canvi de fase en el temps, la fase del senyal transmès s'ha de conèixer. Per això, un sistema totalment coherent es requereix per preservar i seguir el canvi de fase en el senyal rebut.

En un sistema RADAR biestàtic, el *Doppler shift* depèn en tres factors:

(i) *Doppler shift* màxim. Si l'objectiu s'està movent a una velocitat V , és:

$$f_{D_{\max}} = \frac{2f}{c} |V| \quad (2.26)$$

(ii) Aquest factor està relacionat al factor de triangulació biestàtica:

$$D = \cos\left(\frac{\alpha_R - \alpha_T}{2}\right) = \cos\left(\frac{\beta}{2}\right) \quad (2.27)$$

(iii) Aquest factor està relacionat amb l'angle δ entre la direcció de moviment de l'objectiu i la direcció del bisector: $C = \cos\delta$. Per això, el *Doppler shift* d'un sistema RADAR biestàtic es pot representar per:

$$f_{D_{Bi}} = f_{D_{\max}} \cdot D \cdot C = \frac{2f}{c} |V| \cos\left(\frac{\beta}{2}\right) \cos\delta \quad (2.28)$$

Si els dos objectius estan separats en rang i velocitat, poden ser resolts per la resolució en rang i *Doppler* del sistema RADAR. En un sistema RADAR monoestàtic, si la resolució de rang és coneguda com Δr_{Mono} i la resolució *Doppler* com $\Delta f_{D_{Mono}}$, llavors la resolució del rang i *Doppler* del sistema RADAR biestàtic es pot determinar per la resolució monoestàtica de rang corresponent i la resolució *Doppler* escalada per una funció de l'angle biestàtic β . Per això, la resolució de rang biestàtica és:

$$\Delta r_{Bi} = \frac{1}{\cos\left(\frac{\beta}{2}\right)} \Delta r_{Mono} \quad (2.29)$$

I la resolució *Doppler* biestàtica és:

$$\Delta f_{D_{Bi}} = \cos\left(\frac{\beta}{2}\right) \Delta f_{D_{Mono}} \quad (2.30)$$

2.2. La tecnologia RADAR.

El sistema RADAR (*Radio Detection and Ranging*) és un dispositiu que opera radiant energia electromagnètica i detectant l'eco retornat pels objectes reflectants de radiació en un rang determinat. L'anàlisi de la senyal retornada proporciona informació sobre l'objectiu a detectar. El rang, o distància a l'objectiu, es calcula a partir del temps que tarda l'ona electromagnètica a arribar fins l'objectiu i tornar. La posició angular de l'objectiu es determina gràcies a una antena direccional d'ample de banda estret que és capaç de percebre l'angle d'arribada de la senyal rebuda. Quan l'objectiu està en moviment, el RADAR pot calcular la seva trajectòria i, mitjançant algorismes com l'EKF, predir la seva posició [2].

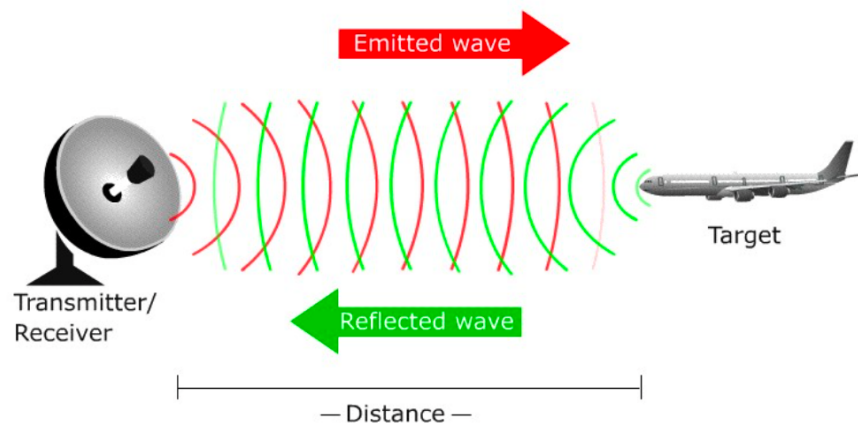


Figura 2.5. Esquema emissor-receptor d'un sistema RADAR.

Quan la resolució és suficient, el RADAR pot proporcionar informació suficient per predir la forma i mida d'un objectiu. Mitjançant aquesta tecnologia es poden detectar objectius relativament petits en rangs de distància molt variables, des dels mil·límetres de resolució als quilòmetres.

El principi de la tecnologia RADAR es pot aplicar a freqüències des dels MHz (regió HF de l'espectre electromagnètic) fins els bilions d'Hz (*Laser RADAR*). Segons les freqüències utilitzades varia el rang i, per tant, l'aplicació (Taula 2.1).

Designació	Rang de freqüències nominal
HF	3 MHz - 30 MHz
VHF	30 MHz - 300 MHz
UHF	300 MHz - 1000 MHz
L	1000 MHz - 2000 MHz
S	2000 MHz - 4000 MHz
C	4000 MHz - 8000 MHz
X	8000 MHz - 12000 MHz
K _u	12 GHz - 18 GHz
K	18 GHz - 27 GHz
K _a	27 GHz - 40 GHz
V	40 GHz - 75 GHz
W	75 GHz - 110 GHz
mm	30 GHz - 300 GHz

Taula 2.1. Bandes de freqüència del RADAR.

La regulació de l'espectre en RADARS queda tal com es pot veure en la Figura 2.6.

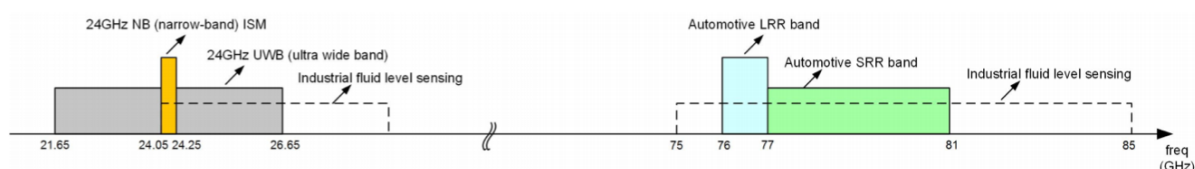


Figura 2.6. Regulació de l'espectre en el sistema RADAR.

Hi ha dues bandes de freqüència per aplicacions de RADAR: la de 24 GHz i la de 77 GHz. Les limitacions de hardware a la banda de 77 GHz van fer que inicialment fos complicat el desenvolupament de noves aplicacions en aquesta banda i la majoria de sistemes es duïen a terme a la banda baixa. La banda de 24 GHz és una banda sense llicència i a l'inici estava dividida en la banda ISM (*Industrial, Scientific and Mechanical*) de 250 MHz d'ample de banda, des dels 24 GHz als 24.25 GHz, i la UWB (*Ultra Wide Band*) dels 21.65 GHz als 26.65 GHz. Tot i que inicialment estava dedicada al RADAR, degut al desenvolupament de les tecnologies 5G la UWB serà eliminada per la ITU en RADARs, amb data màxima 2022 ('*sunset date*') [7].

En aplicacions que es requereix de més resolució, que és inversament proporcional a l'ample de banda, com és el cas de la mesura de nivell de líquids en tancs, es permet un major ample de banda. En aquest cas, cal tenir en compte que no es preveuen interferències amb altres

sistemes ja que s'utilitzen en aplicacions en tancs metàl·lics que eviten la propagació fora de l'entorn.

2.2.1. El RADAR en el sector d'automoció.

Actualment, el RADAR juga un paper molt important en el sector de l'automoció. A finals dels anys 90, la necessitat d'un canvi d'enfocament militar del RADAR cap al sector comercial va portar a les companyies a posar el focus d'atenció en RADARs de curt abast per automoció. D'aquesta manera, els sistemes *millimeter-wave (mm-wave)*, RADARs operatius en el rang dels 24 GHz i 77 GHz, van guanyar molt d'interès [2].

Les primeres aplicacions del RADAR en automoció van ser els sistemes d'ajuda en aparcament, advertència de col·lisió, i AAC que, a més de mantenir una velocitat de creuer, mantenen la distància de seguretat amb els vehicles de davant.

Originalment, s'utilitzaven sistemes polsats a 24 GHz. Això actualment ha canviat degut als RADARs FMCW i a l'existència de la banda de 77 GHz. A banda de raons tècniques (major ample de banda), la banda de 24 GHz deixarà de ser utilitzada en RADARs a partir del 2022 degut a regulacions legislatives [5]. Amb l'eliminació de la UWB, la banda ISM no és de massa interès per aplicacions del RADAR d'automoció (poc ample de banda). Aquest canvi de freqüència suposa una millora de quasi un factor 20 en termes de resolució en rang, ja que és proporcional a l'escombrat d'ample de banda, i una millora en un factor 3 en resolució de velocitat, ja que és inversament proporcional a la freqüència [7]. La tendència futura és anar cap a freqüències superiors a 100 GHz [5].

Un altre desenvolupament en relació al disseny de RADARs d'automoció són l'antena i el *beam forming* (múltiples antenes per transmetre la mateixa senyal). Inicialment, s'utilitzaven antenes orientades mecànicament. L'actual generació de RADARs apliquen *patch arrays* amb *beam steering* electrònics o *beam forming* digitals.

Els RADARs moderns utilitzen una sofisticada gamma de formes d'ona, receptors i algorismes de processament de senyal per adquirir dades de forma fiable i d'alta resolució en ambients amb multitud d'ones EM i un gran nombre d'interferències. Actualment els RADARs en automoció utilitzen formes d'ona FMCW amb amplituds de banda d'entre 250 MHz i 1 GHz per sensors de llarg abast i fins a 4 GHz per sensors de curt abast.

Complementant-se amb els sistemes sensorials com ara LIDAR, ultrasons o càmeres de vídeo, el RADAR segueix essent en l'actualitat una part important dels sistemes de sensors en automoció per la seva funcionalitat optimitzada, robustesa, confiabilitat i poca sensibilitat a les condicions climàtiques adverses. En aquest sector, juntament amb l'industrial, s'estan utilitzant de manera massiva i, per tant, estan disponibles a preus molt raonables.

La majoria de prototips amb automatització de la conducció es basen en un sistema de visió amb càmera i un sensor LIDAR giratori muntat al sostre del vehicle. Aquesta solució proveeix al sistema un núvol de punts molt precís que permet implementar funcions d'alt nivell en els vehicles autònoms, com la planificació de ruta o detecció d'objectes. Tot i així, els sistemes LIDAR no són perfectes i tenen inconvenients com ara el seu baix rendiment en condicions de baixa llum o el seu elevat cost. En la Taula 2.2 i la Taula 2.3 s'introdueix una comparativa dels sensors més implementats en aquests sistemes sota diferents condicions de funcionament.

Percepció sensorial	RADAR	LIDAR	Visió (Càmera)
Rang	++	+	++
Resolució de Rang	+	++	0
Resolució angular	0	++	+
Funcionament en males condicions meteorològiques	++	0	-
Funcionament en la foscor	++	++	--
Funcionament en molta llum	++	+	+
Color/Contrast	--	--	++
Velocitat radial	++	0	-

Taula 2.2. Comparació del rendiment de sensors d'automoció

Sensor	Fortaleses	Debilitats
Infraroig (IR)	Baix cost. Ample FOV. Curt rang designat.	Afecció per la pluja i pols. No mesura de velocitat
Ultrasons (US)	Baix cost. Curt rang designat.	No resolució angular. Afecció pel soroll i vent.
Làser	Alta directivitat.	Afecció per pluja i boira.
Càmera	Bon FOV. Rang màxim mitjà.	No dades de velocitat.
RADAR	Bo sota qualsevol condició meteorològica. Rang fins 250 m. Cost correcte.	No classificació d'objectes.
LIDAR	Millor resolució de velocitat i rang.	Alt cost.

Taula 2.3. Comparació entre diferents tecnologies sensorials disponibles [7].

El gran avantatge del RADAR respecte la resta de sensors és la seva robustesa sota condicions meteorològiques adverses i la possibilitat de mesurar directament la velocitat angular dels objectius mòbils detectats gràcies a l'efecte Doppler.

En l'actualitat, per tal de reduir accidents cada vegada es disposen de més sistemes de seguretat tant a l'interior com exterior del vehicle. El sistema ADAS, molt destacat, no pot prevenir accidents completament, però poden protegir el vehicle i als seus usuaris de factors humans i aquest error humà és la causa de la majoria d'accidents de trànsit [3]. Aquest sistema pot actuar en múltiples aplicacions (com ara *collision warning and avoidance*, *adaptive cruise control*, *lane keeping*, *autonomous parking*, etc.), percebent amb molta precisió (detecció sensorial, enteniment i modelatge) tot l'entorn del vehicle tot actuant en la presa de decisions a temps real [4].

Tot i així, aquesta tasca requereix d'una gran complexitat ja que els entorns són altament dinàmics i els recursos són limitats en cada vehicle. Arran de la necessitat d'una alta fiabilitat,

s'ha considerat optar per la fusió multisensorial de dades. Així doncs, per minimitzar les debilitats de cadascun dels sensors esmenats a la Taula, és necessari implementar una fusió sensorial que, encara normalment es fa de forma redundant, combina la informació de cadascun dels sensors. Cap tecnologia és capaç de cobrir tots els requeriments; la fusió dels sensors és considera un element pivot que porta cap a la total garantia de seguretat en la vida real.

El RADAR i la càmera tenen característiques complementàries que porten cap a la desitjada fusió sensorial de forma natural. Poden abordar limitacions inherents de cada sensor que, si no fos per la seva combinació, quedarien limitats. Les dades dels diferents sensors utilitzats per diferents tasques del vehicle es poden combinar, verificar i fusionar per tal d'aconseguir una gran millora en precisió i fiabilitat. La Taula 2.4 recull un resum dels avantatges de la fusió sensorial [4].

Percepció sensorial	Càmera	RADAR	Fusió
Distància	0	+	+
Angle	+	0	+
Velocitat radial	-	+	+
Velocitat lateral	0	-	+
Frontera	+	-	+
Obstacle	0	+	+
Classificació	+	0	+
Temps/Lluminositat/Pols	-	+	+

Taula 2.4. Avantatge de la fusió en diferents tasques.

Un altre gran camp d'aplicació actual per la millora en la seguretat dels vehicles són els sistemes de detecció de l'ocupació en el seu interior i els sistemes de detecció de gestos per reduir les distraccions humanes.

Els sistemes *Hand-gesture-based user interfaces* (UIs) en vehicles poden reduir les distraccions visuals i cognitives, tot millorant la seguretat i confort dels usuaris del vehicle. A més, poden ser configurats segons les preferències dels usuaris i, en el futur, es poden ampliar per incloure funcionalitats de monitoratge en la conducció. Fins i tot, s'està treballant per poder estandarditzar les interfícies vehiculars de gestos [6].

En comparació a sensors de color i profunditat, els RADARS són robustos a les condicions ambientals d'il·luminació, tenen un cost més baix, menor complexitat computacional i utilitzen menys energia. A més, el senyal RADAR pot penetrar materials opacs com el plàstic. Per tant, són ideals per l'interior d'un vehicle. Tot i així, és necessari combinar-se amb aquests d'altres sensors, tal com s'ha comentat anteriorment, juntament amb xarxes convolucionals DNNs (*Deep Neural Networks*) per fusionar les dades de tots els sensors i classificar-ne els gestos amb la mà.

2.3. Fonaments teòrics dels sensors FMCW.

Un sistema RADAR *mmWave* està compost per antenes de Tx i Rx i components d'RF, així com components analògics (com l'oscil·lador de rellotge) i components digitals, com ara convertidors ADC i, en el nostre cas, microprocessadors i acceleradors hardware (DSPs). En la Figura 2.7 es pot veure l'esquema de blocs d'un RADAR FMCW [2] [7].

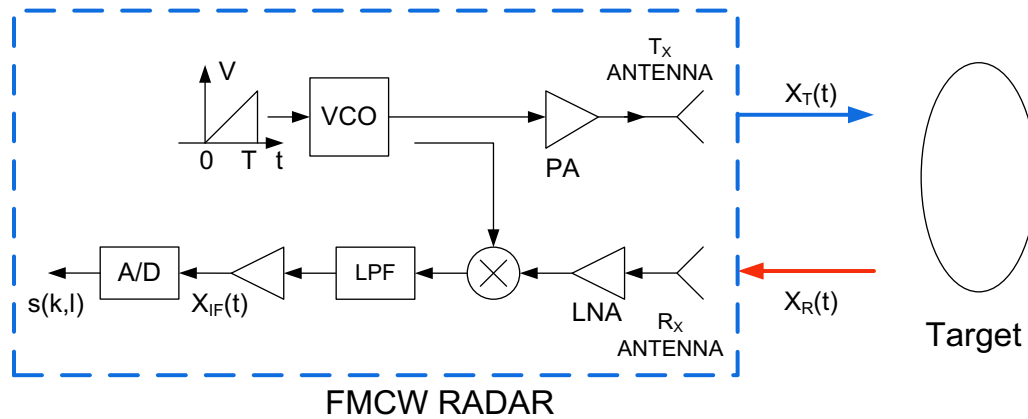


Figura 2.7. Diagrama de blocs generalitzat d'un RADAR FMCW.

Un RADAR FCMW emet una senyal de microones contínua, modulada en freqüència amb una ona de baixa freqüència, normalment en forma d'ona de serra i període T . Aquesta senyal se la coneix amb el nom de *chirp*.

Per simplificar l'anàlisi s'assumeix que la senyal rebuda de la reflexió d'un objecte estacionari és una còpia de la senyal transmesa i endarrerida en un temps de propagació τ :

$$\tau = \frac{2R}{c} \quad (2.31)$$

On R és el rang o distància fins l'objecte, i c és la velocitat de la llum.

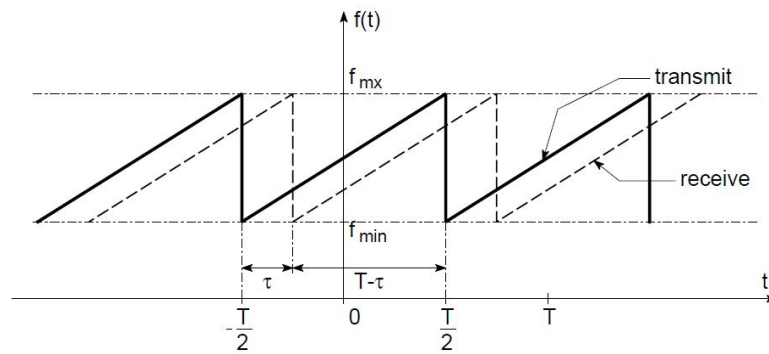
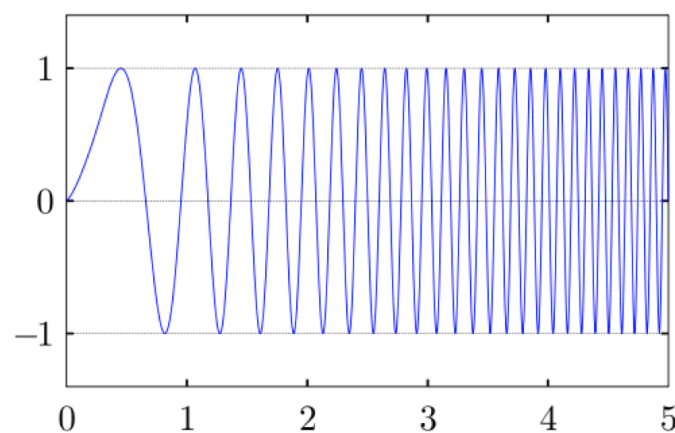


Figura 2.8. Freqüència de les senyals transmeses i rebudes.

Figura 2.9. Senyal de tipus *chirp* representada en el domini del temps.

Al restar les freqüències de la senyal transmesa i la senyal rebuda mitjançant un mesclador de freqüències, obtenim la senyal de freqüència intermèdia (IF). Com que la senyal rebuda té la mateixa forma d'ona que la senyal transmesa però endarrerida en el temps, la senyal IF obtinguda a partir de la resta ha de ser una senyal amb freqüència constant. Llavors, la sortida del mesclador de senyals en el domini del temps haurà de tenir forma d'ona sinusoidal, ja que té freqüència de valor constant tal com es pot veure en la Figura 2.10.

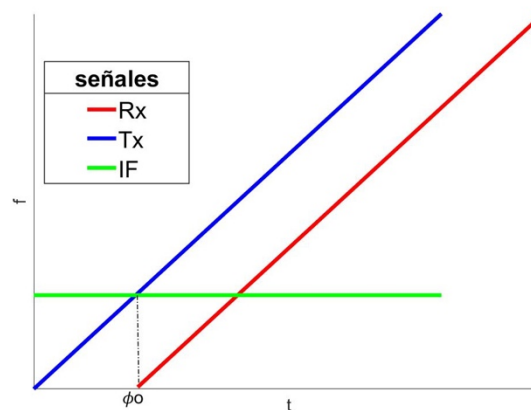


Figura 2.10. Senyal IF en funció de Tx i Rx.

Anomenarem Φ_0 a la fase inicial de la senyal IF, que és la diferència de la fase de la senyal *chirp* Tx i la senyal *chirp* Rx en el moment corresponent al començament de la senyal IF:

$$\phi_0 = 2\pi f_c \tau \quad (2.32)$$

On τ és el temps de propagació i f_c és la freqüència inicial de la senyal *chirp*. Prenent com aproximació que la pendent de la variació del *chirp* i la distància són suficientment petites, es pot derivar a la següent equació:

$$\phi_0 = \frac{4\pi R}{\lambda} \quad (2.33)$$

Amb λ com a longitud d'ona. Així doncs, per un objecte detectat a una distància R del RADAR, la senyal IF serà una ona sinusoidal amb la forma:

$$A \sin(2\pi f_0 t + \phi_0) \quad (2.34)$$

On:

$$f_0 = \frac{S2R}{c} \quad (2.35)$$

Essent S la pendent de la freqüència del senyal *chirp*. La darrera equació, 2.35, permet trobar la distància fins un objecte a partir de l'anàlisi freqüencial de la senyal IF.

En el cas de la detecció de varis objectes, cadascun generarà una senyal d'Rx diferent que, a la vegada, generaran una senyal d'IF amb varis valors de freqüència. Mitjançant l'anàlisi de *Fourier* es poden separar freqüencialment i d'aquesta manera poder trobar la presència de diferents objectes.

2.3.1. Paràmetres de chirp.

En aquest subapartat es detallaran els càlculs necessaris per la correcta configuració dels paràmetres d'interès d'un RADAR FMCW [2] [7].

2.3.1.1. Rang màxim.

De l'equació 2.35 pot obtenir-se el rang d'un objecte detectat pel RADAR en funció de la freqüència del senyal IF. Tenint en compte que la senyal IF és típicament digitalitzada mitjançant un ADC pel seu processament en DSP, l'ample de banda de la senyal està limitat pel temps de mostreig:

$$F_s \geq \frac{S2 \cdot R_{m\grave{a}x}}{c} \quad (2.36)$$

On F_s és la freqüència de mostreig i $R_{m\grave{a}x}$ és el rang màxim detectable pel RADAR. Aïllant de l'equació 2.36, es pot obtenir el rang màxim:

$$R_{m\grave{a}x} = \frac{F_s c}{2S} \quad (2.37)$$

2.3.1.2. Resolució de rang.

La resolució de rang permet distingir d'entre dos o més objectes propers. Quan dos objectes s'apropen, el RADAR deixa de diferenciar-los en algun punt com objectes separats i els identifica com un d'únic. Incrementant la longitud de la senyal IF, aquesta resolució pot incrementar-se. Per fer-ho, l'ample de banda B ha d'augmentar proporcionalment. Així, es separen els pics de freqüència en l'espectre de la senyal IF. Dos senyals IF es poden diferenciar sempre que:

$$\Delta f > \frac{1}{T} \quad (2.38)$$

Ja que una finestra d'observació (T) no pot diferenciar entre continguts freqüencials separats més d' $1/T$ Hz. Diferenciant l'equació 2.35, i sabent que $B = ST$, es pot expressar l'equació 2.38 com:

$$\Delta R = \frac{c}{2S} \Delta f = \frac{c}{2ST} = \frac{c}{2B} \quad (2.39)$$

Per tant, s'obté que la resolució de rang només depèn de l'ample de banda del *chirp*:

$$\Delta R = \frac{c}{2B} \quad (2.40)$$

2.3.1.3. Mesura de la velocitat.

Per mesurar la velocitat, un RADAR FMCW transmet 2 senyals *chirp* en un interval de temps T_c . Les senyals *chirp* rebudes són processades amb un algorisme FFT per extreure'n el rang. La senyal de rang de les senyals reflectides tindrà un pic en la mateixa freqüència, però de diferent fase. A partir d'aquesta fase es pot caracteritzar el moviment de l'objecte, sabent que $R=vT_c$. Utilitzant l'equació 2.33:

$$\Delta\phi = \frac{4\pi v T_c}{\lambda} \quad (2.41)$$

Aïllant v i imposant que $|\Delta\phi| < \pi$, s'obté:

$$v < \frac{\lambda}{4T_c} \quad (2.42)$$

De la darrera equació 2.42 es pot trobar la velocitat màxima mesurable per dues senyals *chirp* separades un temps T_c . Per augmentar aquesta velocitat màxima és necessari disminuir els temps de transmissió entre senyals T_c .

$$v_{\max} = \frac{\lambda}{4T_c} \quad (2.43)$$

La mesura de velocitat de múltiples objectes en el mateix rang té un desenvolupament similar a la mesura del rang de diversos objectes.

2.3.1.4. Resolució de velocitat.

Dos freqüències discretes, ω_1 i ω_2 , es poden diferenciar si $\Delta\omega = \omega_2 - \omega_1 > 2\pi/N$ radians. Aïllant de l'equació 2.41:

$$v > v_{res} = \frac{\lambda}{4T_f} \quad (2.44)$$

On $T_f = NT_c$ és el període de trama o frame. La darrera equació 2.44 ens proporciona la velocitat mínima que un RADAR FMCW és capaç de detectar.

2.3.1.5. Detecció d'angle.

Un RADAR FMCW pot estimar l'angle amb el que incideix una senyal reflectida en el pla horitzontal, tal com es pot veure en la Figura 2.11.

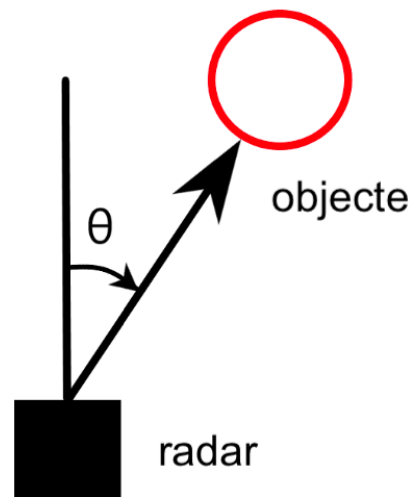


Figura 2.11. Angle d'arribada.

A partir dels canvis de fase en el pic de la transformada FFT, es pot estimar l'angle a un objecte, utilitzant dues o més antenes Rx, com es pot veure en la Figura 2.12.

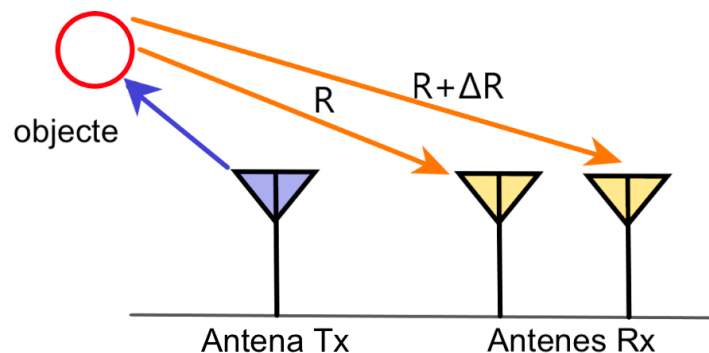


Figura 2.12. Estimació de l'angle d'arribada

Es pot expressar el canvi de fase com:

$$\Delta\phi = \frac{2\pi\Delta R}{\lambda} \quad (2.45)$$

Assumint que es treballa en el pla horitzontal, podem expressar $\Delta R = l\sin(\theta)$, essent l la distància entre antenes. Així doncs, es pot calcular l'angle d'arribada θ com:

$$\theta = \sin^{-1}\left(\frac{\lambda\Delta\phi}{2\pi l}\right) \quad (2.46)$$

S'ha de tenir en compte que $\Delta\phi$ depèn de $\sin(\theta)$, donant lloc a una dependència no lineal. Quan l'angle θ és petit, és possible fer l'aproximació $\sin(\theta) \sim \theta$, d'on es pot entendre que la precisió de l'estimació angular depèn de l'angle d'arribada i és més precisa a menor θ .

2.3.1.6. Camp de visió angular màxim.

Es defineix el camp de visió angular màxim com el l'angle màxim d'arribada que el RADAR pot estimar (Figura 2.13).

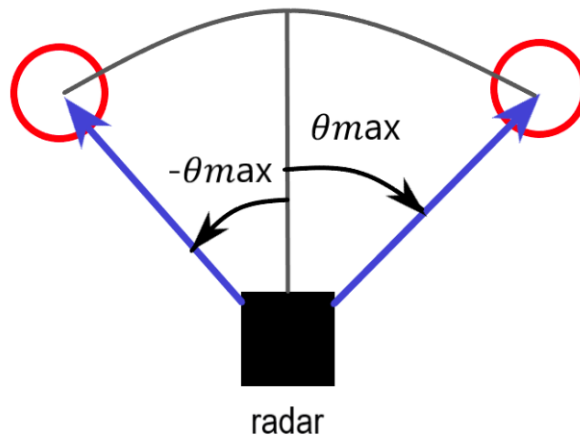


Figura 2.13. Màxim camp de visió angular.

Imposant que $|\Delta\omega| < \pi$, s'obté:

$$\frac{2\pi l \sin(\theta)}{\lambda} < \pi \quad (2.47)$$

Per tant, el camp de visió màxim que pot percebre un RADAR amb dues antenes separades entre sí una distància l , és:

$$\theta_{\max} = \sin^{-1}\left(\frac{\lambda}{2l}\right) \quad (2.48)$$

Així doncs, amb un espai entre antenes de $l = \lambda/2$, s'obté el major camp de visió angular possible ($\pm 90^\circ$). A la pràctica, els RADAR FMCW moderns utilitzen agrupacions virtuals d'antenes per tal de millorar la resolució angular utilitzant tècniques més avançades de *beamforming* i d'estimació d'angle d'arribada (DoA).

2.4. RADAR FMCW mmWave AWR1642 de Texas Instruments.

El RADAR AWR1642 és un dispositiu RADAR single-chip altament integrat que inclou dins la mateixa placa els components de radiofreqüència, processador i DSP que doten el dispositiu de capacitat de càlcul i anàlisis, ports de comunicació i E/S. El dispositiu basa el seu funcionament en la tecnologia RADAR FMWC, que consisteix en l'emissió de microones modulades en freqüència per així obtenir-ne el rang i velocitat.

2.4.1. Descripció del hardware.

El RADAR *mmWave* AWR1642 és un dispositiu RADAR single-chip de 77 GHz altament integrat que inclou 2 antenes de transmissió Tx i 4 antenes de recepció Rx, així com un DSP C674x programable a 600 MHz, que s'ocupa del processament de les senyals RADAR, i un processador ARM Cortex-R4F programable a 200 MHz, dedicat a la configuració, control i calibratge del RADAR. El dispositiu cobreix les bandes de freqüència de 76-77 GHz i 77-81 GHz [2].

L'AWR1642boost ve integrat en un únic mòdul, que inclou ports de comunicació i control externs. Té l'aparença física que es pot veure en les Figures 2.14, 2.15 i 2.16.

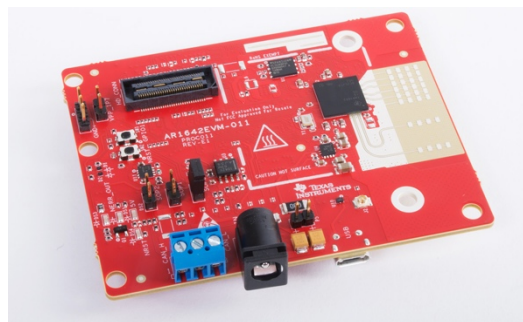


Figura 2.14. Placa AWR1642boost.

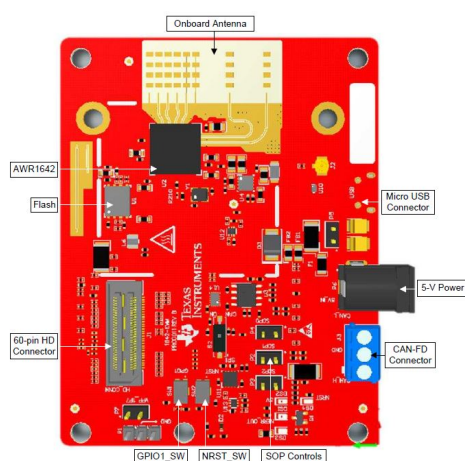


Figura 2.15. Part del davant de la Placa AWR1642boost.

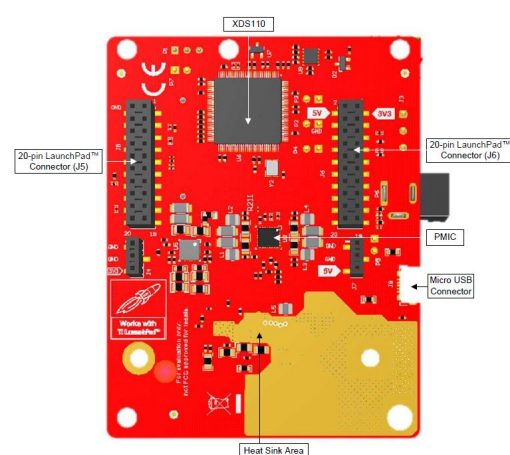


Figura 2.16. Part del darrere de la Placa AWR1642boost.

El RADAR s'ha col·locat sobre una placa de suport d'un gruix d'1 cm impresa en 3D i de dimensions com es pot veure en la Figura 2.17. La Figura 2.18 mostra com s'ha situat el RADAR en el vehicle per l'aplicació de detecció d'ocupació, amb l'ajuda d'un suport amb ventosa per situar-lo just sota el retrovisor. La Figura 2.19 mostra com s'ha utilitzat el RADAR per l'aplicació de reconeixement de gestos amb la mà tot situant-lo en paral·lel a una superfície plana, com ara un escriptori.

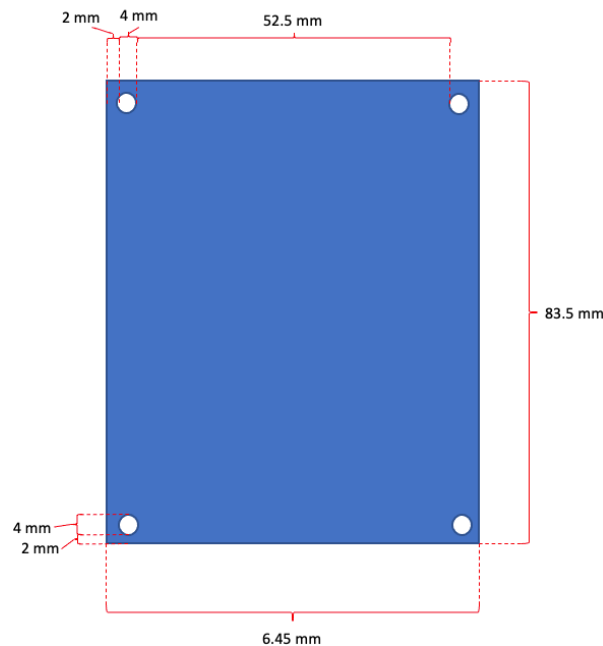


Figura 2.17. Esquema dimensional de la placa de suport al RADAR impresa en 3-D.



Figura 2.18. Il·lustració de la situació del RADAR per l'aplicació de detecció de l'ocupació a l'interior d'un vehicle.

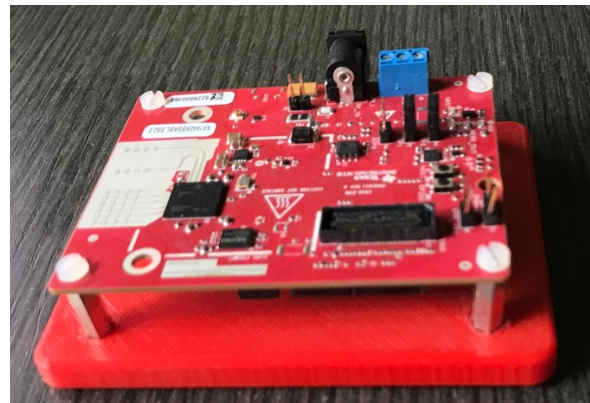


Figura 2.19. Il·lustració de la situació del RADAR per l'aplicació de detecció de gestos amb la mà.

En la Figura 2.20 s'hi pot veure el diagrama de blocs funcional del RADAR en qüestió [8].

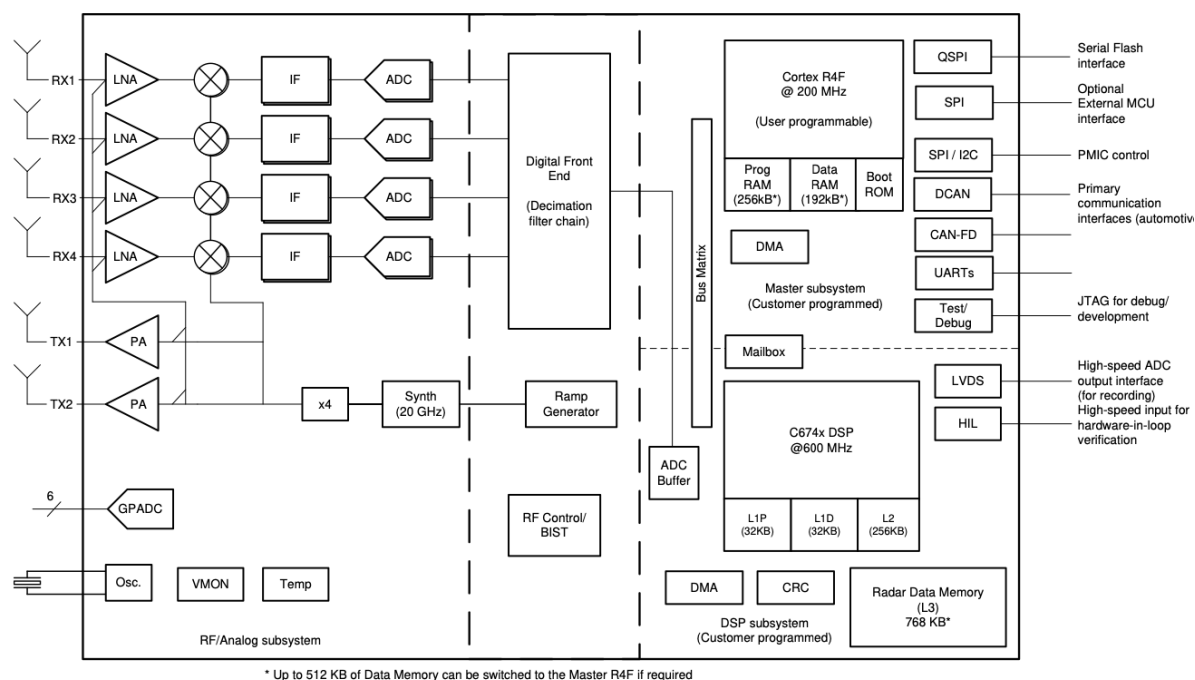


Figura 2.20. Diagrama de blocs funcional del dispositiu RADAR *mmWave* AWR1642.

En la Taula 2.5 s'hi pot veure un resum de les característiques del sensor FMCW que inclou el dispositiu RADAR *mmWave* AWR1642.

PLL integrat, transmissor, receptor, <i>Baseband</i> i A2D	
Bandes de freqüència disponibles	76 GHz a 81 GHz
Ample de banda	4 GHz
Quatre canals de recepció	RX1, RX2, RX3, RX4
Dos canals de transmissió	TX1, TX2
<i>Ultra-accurate chirp (timing) engine basat en fractional-N PLL.</i>	
Potència de transmissió Tx	12 dBm
Rx Noise Figure	14 dB (banda de 76 a 77 GHz).
	15 dB (banda de 77 a 81 GHz).
Phase Noise a 1 MHz	- 95 dBc/Hz (en la banda de 76 a 77 GHz).
	- 93 dBc/Hz (en la banda de 77 a 81 GHz).

Taula 2.5. Característiques del sensor FMCW que inclou el dispositiu RADAR *mmWave* AWR1642.

2.4.2. Comunicació i configuració dels paràmetres d'entrada.

El dispositiu RADAR utilitzat inclou una comunicació UART entre el PC i el sensor RADAR. En concret, són dos ports UART: un port per rebre la configuració des de l'ordinador (a través de l'anomenat *Fitxer de Configuració*), i un segon port per transmetre les dades processades. El primer és també a partir del qual es poden instal·lar els *drivers* necessaris per poder ser utilitzat des de l'ordinador, tal com es veurà més endavant.

El format d'entrada dels paràmetres a configurar del RADAR és a través de la línia de comandes per UART i es duu a terme mitjançant el Fitxer de Configuració. La GUI proporcionada en el disseny permet automatitzar l'entrada d'aquests paràmetres mitjançant els arxius .cfg, que recullen en un sol arxiu totes les comandes necessàries pel funcionament del RADAR. El format d'aquests arxius consisteix en una sèrie de comandes seguides dels valors que es volen configurar.

2.4.3. Format de les dades de sortida.

Les dades processades, tant a alt com baix nivell, són enviades a través de la UART en cada *frame* en ordre *little endian* (comença per l'enter amb el byte menys significatiu i acaba amb el més significatiu). El missatge està compost per una capçalera fixa més un número variable de segments en format *Tag Length Value* (TLV). Aquests, a la vegada, estan composts per una capçalera fixa amb informació sobre la longitud del segment i van seguits del cos del missatge (de mida variable segons el nombre de deteccions).



Figura 2.21. Format del missatge de sortida.

L'estructura del missatge és la següent:

- **Capçalera del missatge (*Frame Header*):** conté informació general com la versió i plataforma del hardware, paraula de sincronització, *timestamp* per mesurar el temps en el qual s'envien les dades, longitud del paquet a enviar, número de frame, temps de processat, etc.
- **TLV del núvol de punts:** conté informació sobre el núvol de punts detectat en format *range-azimuth-doppler*, així com la ràtio senyal-soroll.
- **TLV d'ID de *tracking*:** conté els valors de posició en m/s i acceleració en m/s^2 per cada detecció, al voltant dels eixos x i y, a més d'un número d'identificació per distingir entre diferents deteccions i una matriu de covariància, útil per l'algoritme d'EKF (necessari per calcular la trajectòria d'objectes i predir-ne la seva posició).
- **TLV de correlació:** associa els punts del núvol de punts amb cada *tracking*.

Cal fixar-se que s'enviarà un TLV d'ID de *tracking* per cada objecte monitoritzat i un TLV del núvol de punts per cada punt en el núvol de punts.

2.4.4. Distribució dels eixos en el RADAR.

El RADAR és capaç de capturar punts en 2-D en la superfície perpendicular a les seves antenes, tal com es pot veure en la Figura 2.22.

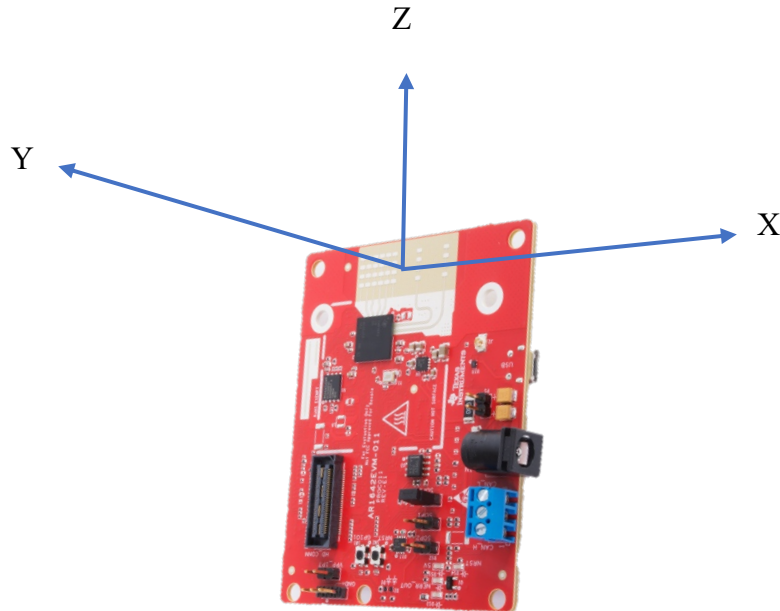


Figura 2.22. Situació dels eixos de coordenades respecte l'agrupació d'antenes virtual del RADAR.

Els punts de l'eix Z que retorna són sempre 0. És així ja que el RADAR només retorna punts en el pla horitzontal (en la direcció de l'agrupació d'antenes virtual).

2.4.5. mmWave Demo Visualizer.

En aquest punt es detallarà com utilitzar la GUI, la interfície gràfica accessible des d'un cercador online a través de la qual es poden realitzar captures amb el RADAR i des d'on també es pot configurar i descarregar el Fitxer de Configuració, necessari per poder enviar les comandes al RADAR i així poder ser utilitzat des d'altres plataformes (*Matlab, Python, etc.*).

En primer lloc, és necessari instal·lar-se el software adient per tal de poder utilitzar la GUI. Per fer-ho, cal anar a l'enllaç web [9].

Una vegada l'usuari es trobi a la pàgina web, s'ha de clicar el botó *TI Cloud Agent* per tal de poder-se instal·lar l'extensió que permetrà detectar i utilitzar el RADAR des de l'ordinador (Figura 2.23).

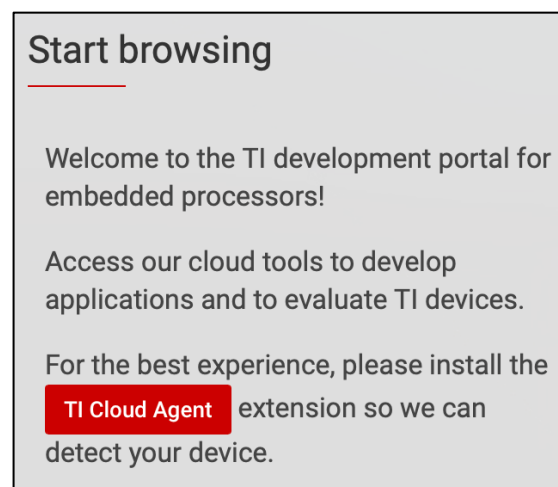


Figura 2.23. Quadre diàleg d'instal·lació del *TI Cloud Agent*.

Seguidament, apareixerà el següent quadre de la Figura 2.24, des d'on s'ha de prémer el botó 'Download' per descarregar la versió corresponent.

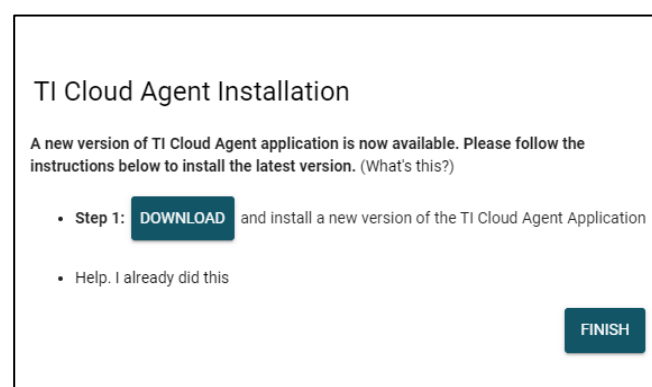


Figura 2.24. Quadre diàleg d'instal·lació del *TI Cloud Agent*.

Una vegada s'ha realitzat aquest pas, es procedirà a anar a l'enllaç web [10]. Com s'ha dit anteriorment, el RADAR disposa d'una interfície web gràfica des d'on es poden configurar diversos paràmetres i visualitzar les dades capturades pel RADAR (*Demo Visualizer*). La versió utilitzada és la 3.5.0.

La GUI es distribueix en diversos quadres de configuració [11]. La finestra principal és la *Configure* (Figura 2.25) que, a la vegada, es separa en diverses seccions. Aquestes seran descrites a continuació.

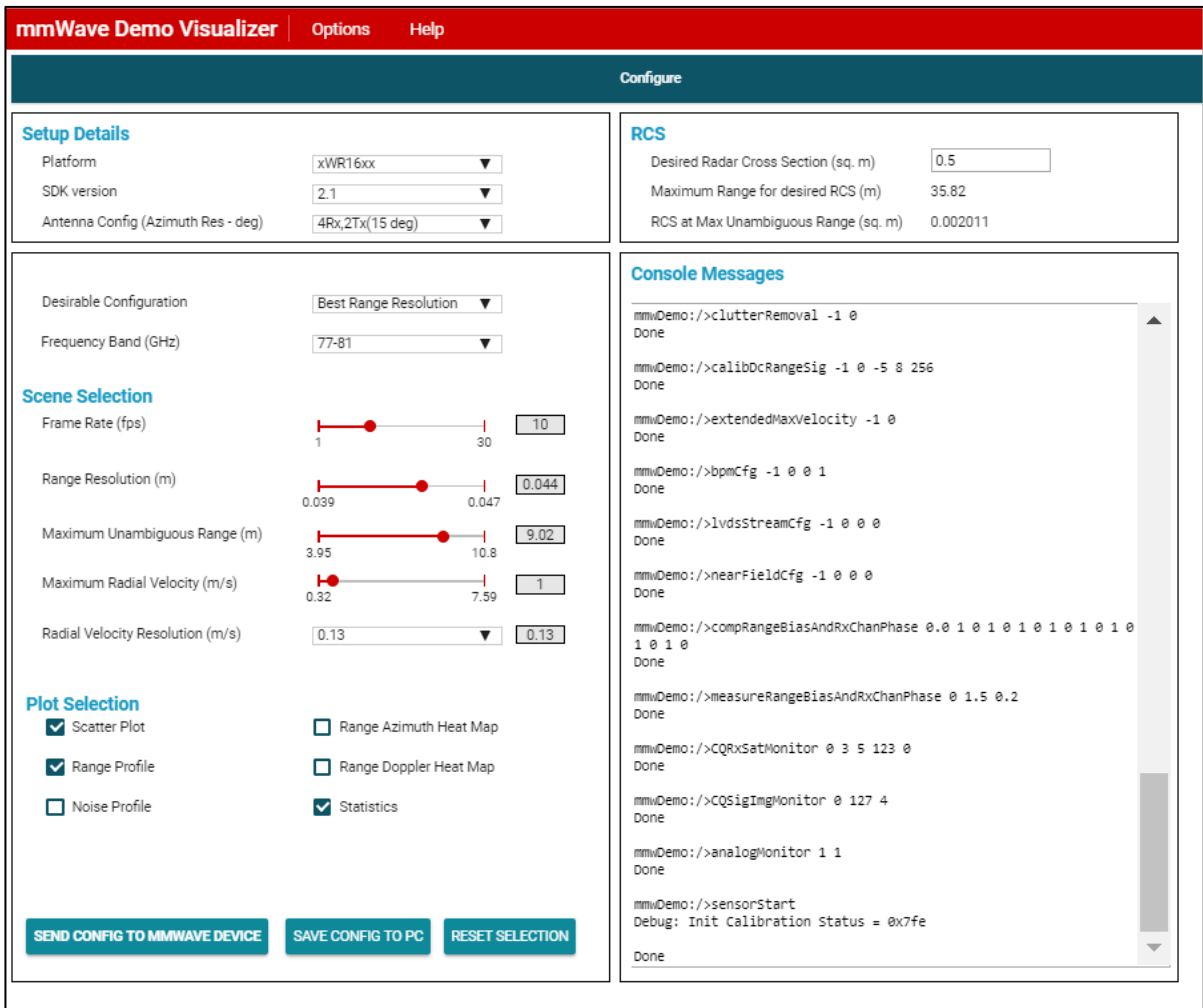


Figura 2.25. Aparença de la finestra 'Configure' de la GUI.

En primer lloc, s'hi troba la secció *Setup Details* (Figura 2.26), on s'hi seleccionen les característiques del RADAR. D'aquí només s'hi ha de seleccionar la *Platform xWR16xx* i la versió SKD que, com s'ha dit, la utilitzada és la 3.5.

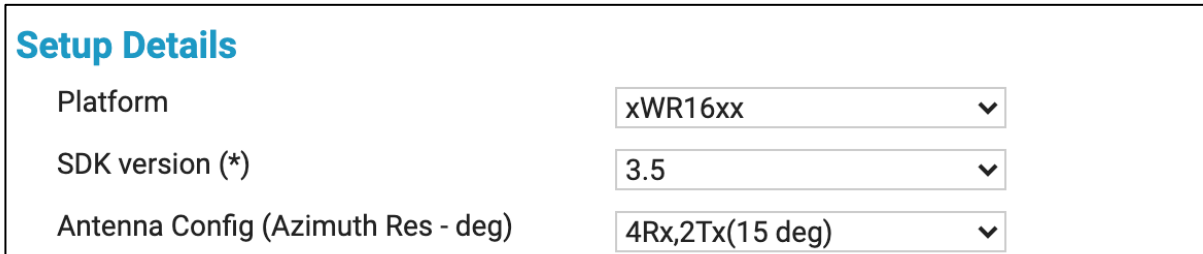


Figura 2.26. Aparença de la secció 'Setup Details'.

En segon lloc, hi ha una secció on es configuren els paràmetres in situ del RADAR (Figura 2.27). Entre aquests podem destacar la banda freqüencial, la *frame rate*, la *range resolution*, etc. Paràmetres que s'han descrit anteriorment.

The image shows a software interface for configuring radar parameters. It includes a 'Desirable Configuration' section with a dropdown menu set to 'Best Range Resolution'. Below this is the 'Frequency Band (GHz)' set to '77-81'. The 'Calibration Data Save/Restore' section has a dropdown set to 'None' and a text box containing '0x1F0000'. A blue header 'Scene Selection' is followed by five parameter controls:

- Frame Rate (fps):** A slider ranging from 1 to 30, with a red dot at 10 and a numeric input box showing 10.
- Range Resolution (m):** A slider ranging from 0.039 to 0.047, with a red dot at 0.044 and a numeric input box showing 0.044.
- Maximum Unambiguous Range (m):** A slider ranging from 3.95 to 10.8, with a red dot at 9.02 and a numeric input box showing 9.02.
- Maximum Radial Velocity (m/s):** A slider ranging from 0.47 to 7.47, with a red dot at 1 and a numeric input box showing 1.
- Radial Velocity Resolution (m/s):** A dropdown menu set to 0.13 and a numeric input box showing 0.13.

Figura 2.27. Aparència de la secció que configura els paràmetres in situ del RADAR.

En tercer lloc, s'hi situa la secció on seleccionar les gràfiques de dades a representar. S'hi pot seleccionar entre:

- *Scatter Plot.*
- *Range Profile.*
- *Noise Profile.*
- *Range Azimuth Heat Map.*
- *Range Doppler Heat Map.*
- *Statistics.*

En el següent punt es mostrarà l'aparència de cadascuna d'aquestes gràfiques.

En quart lloc, s'hi troba la secció *RCS (RADAR Cross Section)*. Aquí s'hi pot seleccionar l'àrea equivalent vista pel RADAR, és a dir, quina serà l'àrea dels objectes detectats per aquest. Per la detecció de persones, la secció recomanada és de 0.5 m (Figura 2.28).

RCS	
Desired Radar Cross Section (sq. m)	0.5
Maximum Range for desired RCS (m)	35.82
RCS at Max Unambiguous Range (sq. m)	0.002011

Figura 2.28. Aparença de la secció 'RCS'.

Un altre aspecte a configurar són els ports i les seves *baud rates*. S'han d'establir dos ports: l'*Application Port* (que serveix per enviar les dades de configuració) i el *Data Port* (que serveix per enviar les dades capturades).

Si el dispositiu és *Windows*, la configuració és la de la Figura 2.29.

Serial Port Configuration

mmWave:User/Application Port (CLI CFG_port)

mmWave:Auxillary Data port (Demo output DATA_port)

Ports:

Baud Rates:

REFRESH
OK
CANCEL

Figura 2.29. *Serial Port Configuration* per un dispositiu *Windows*.

Si el dispositiu és *MacBook*, la configuració és la de la Figura 2.30.

Serial Port Configuration

mmWave:User/Application Port (CLI CFG_port)

mmWave:Auxillary Data port (Demo output DATA_port)

Ports:

Baud Rates:

REFRESH
OK
CANCEL

Figura 2.30. *Serial Port Configuration* per un dispositiu *MacBook*.

Si es prem el botó *REFRESH*, els ports es seleccionen automàticament. Si els ports estan correctament connectats, es mostra la configuració de la Figura 2.31 a la part inferior de la pantalla mentre que, si estan desconnectats, la de la Figura 2.32.

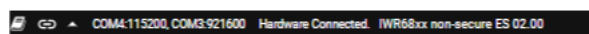


Figura 2.31. Aparença de la configuració quan els ports estan correctament connectats.



Figura 2.32. Aparença de la configuració quan els ports no estan correctament connectats.

En resum, els ports que s'han utilitzat per als ordinadors emprats en aquest projecte són els de la Taula 2.6.

	Application Port	Data Port
Windows	COM12	COM13
MacBook	/dev/tty.usbmodemR00410271	/dev/tty.usbmodemR00410274

Taula 2.6. Ports utilitzats segons el dispositiu utilitzat en aquest projecte.

Finalment, hi ha una última secció anomenada *Console Messages* on, si es prem el botó *SEND CONFIG TO MMWAVE DEVICE*, hi apareixen les comandes que s'envien al RADAR. Aquestes comandes es poden descarregar en un fitxer anomenat *Fitxer de Configuració*, que és el que s'ha utilitzat en el programa *Matlab* per tal de poder adquirir les dades del RADAR, tot prement el botó *SAVE CONFIG TO PC*. Els paràmetres es poden configurar a través de la GUI.

2.4.5.1. Paràmetres del Fitxer de Configuració.

A continuació, a la Taula 2.7 es detallaran aquells paràmetres del fitxer de configuració (*CLI command*) i a partir de quin *tab* dels vistos anteriorment es poden configurar [11].

Tab	Comanda fitxer	Descripció
Antenna Config (Azimuth Resolution - Degrees)	<i>channelCfg</i>	Defineix la màscara d'antena per RX i TX.
	<i>chirpCfg</i>	Crea els <i>chirpCfgs</i> .
Frequency Band (GHz)	<i>profileCfg</i>	Defineix la freqüència d' <i>start</i> en GHz.
Frame Rate (fps)	<i>frameCfg</i>	Defineix la duració del <i>frame</i> (periodicitat de <i>frame</i> en ms).
Range Resolution (m)	<i>profileCfg</i>	Defineix la <i>frequency slope constant</i> i <i>Ramp end time</i> en μ s.
Maximum Unambiguous Range (m)	<i>profileCfg</i>	Defineix la <i>frequency slope constant</i> .
Range Resolution (m)	<i>profileCfg</i>	Defineix el nombre de mostres ADC, <i>Ramp</i> i <i>end time</i> en μ s, i la freqüència de mostreig de l'ADC en kbps.
Maximum Radial Velocity (m/s)	<i>profileCfg</i>	Defineix l' <i>idle time</i> (temps d'inactivitat) en μ s.
Scatter Plot	<i>guiMonitor</i>	Defineix el paràmetre <i>detected-objects</i> .
Range Profile	<i>guiMonitor</i>	Defineix el paràmetre <i>log-magnitude range</i> .
Noise Profile	<i>guiMonitor</i>	Defineix el paràmetre <i>noise profile</i> .
Range Doppler Heat Map	<i>guiMonitor</i>	Defineix el paràmetre <i>range-doppler heat map</i> .
Statistics	<i>guiMonitor</i>	Defineix el paràmetre <i>statistics</i> .

Taula 2.7. Paràmetres configurables del Fitxer de Configuració.

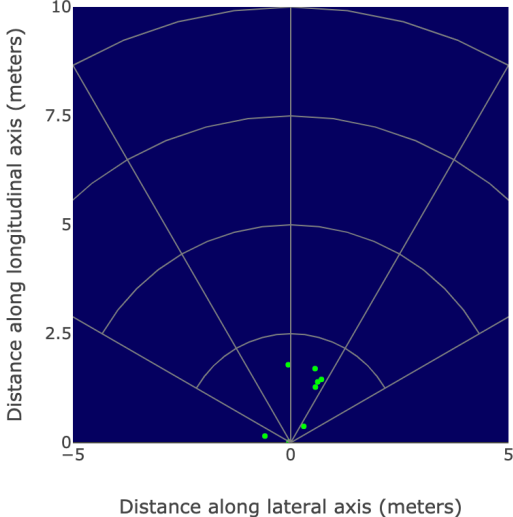
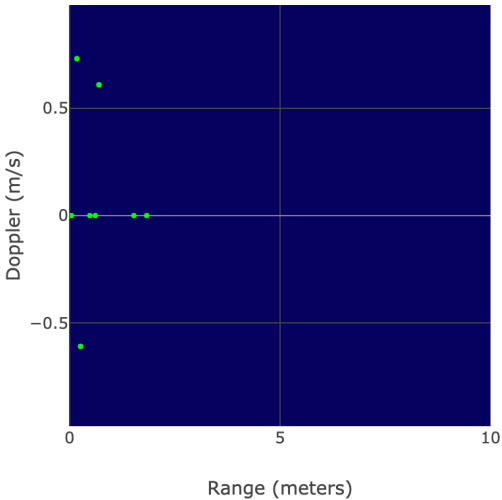
Aquests paràmetres s'han configurat directament des de la interfície gràfica detallada en el punt anterior. A banda, s'ha modificat el paràmetre del Fitxer de configuració que es pot trobar a la Taula 2.8. Serveix per capturar només aquells punts en moviment i no tots els objectes estàtics que es troben al voltant, de manera que, així, pot assegurar-se amb major fiabilitat que el que s'estigui detectant siguin persones o gestos i no objectes estàtics.

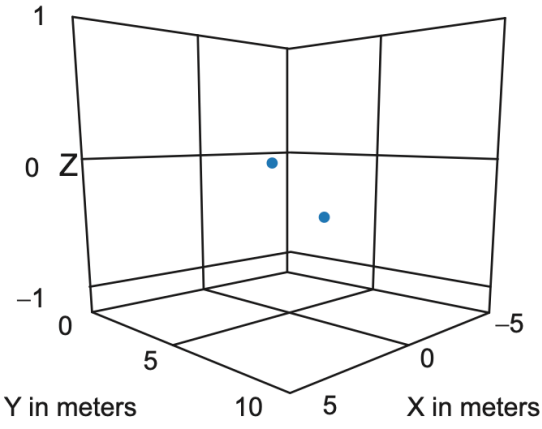
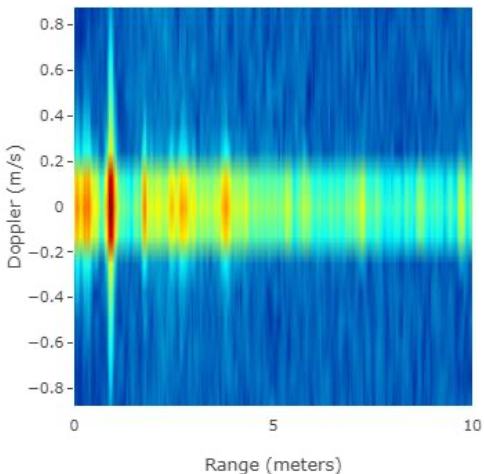
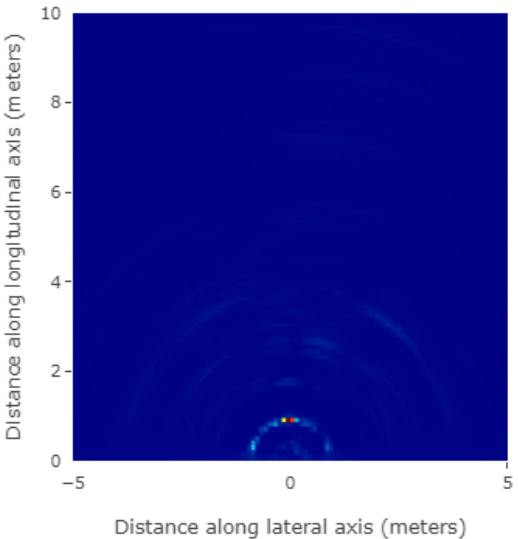
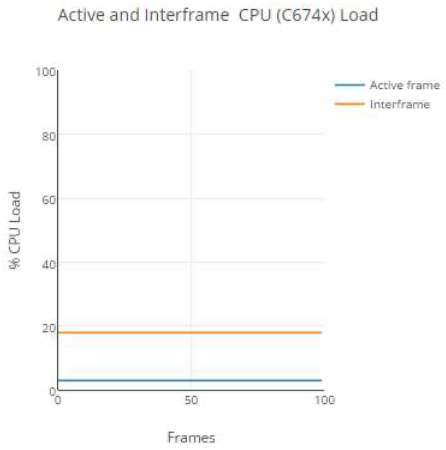
Paràmetre	Configuració	Descripció
<i>clutterRemoval</i>	-1 1	Aquest paràmetre serveix per tal que el RADAR detecti només objectes en moviment (1) o qualsevol tipus d'objecte (0). En la nostra aplicació hem optat per la primera opció (1).

Taula 2.8. Configuració del paràmetre *clutterRemoval*.

2.4.5.2. Plots mmWave Demo Visualizer.

A continuació, a la Taula 2.9 es mostraran els *plots* que es poden representar amb la GUI [11].

X-Y Scatter Plot	Doppler-Range Plot
 Figura 2.33. X-Y Scatter Plot. Aquest plot mostra els objectes detectats en el domini X-Y (m).	 Figura 2.34. Doppler-Range Plot. Aquest plot mostra els objectes detectats utilitzant les coordenades de <i>Range</i> (m) i <i>Doppler</i> (m/s).

3-D Scatter Plot	Doppler-Range Heatmap
 Figura 2.35. 3-D Scatter Plot. Aquesta gràfica només està disponible quan es selecciona el paràmetre <i>Elevation</i> a la secció de configuració <i>Scene Selection</i>. Aquest plot mostra els objectes detectats en l'espai 3-D.	 Figura 2.36. Doppler-Range Heatmap. Aquest plot mostra la matriu completa del cub del RADAR en coordenades de Rang i Doppler utilitzant un <i>heatmap</i> (mapa de calor) plot.
Range-Azimuth Heatmap	CPU Load
 Figura 2.37. Range-Azimuth Heatmap. Aquest plot mostra la matriu completa del cub del pel només 0 <i>Doppler</i>.	 Figura 2.38. CPU Load. Aquest plot mostra la CPU <i>load</i> de la cadena de processament de dades durant la secció activa o de <i>chirping</i> d'un <i>frame/subframe</i> donat i durant la secció d'un <i>frame/subframe</i> no actiu o entre <i>frames/subframes</i>.

Range Profile for zero Doppler

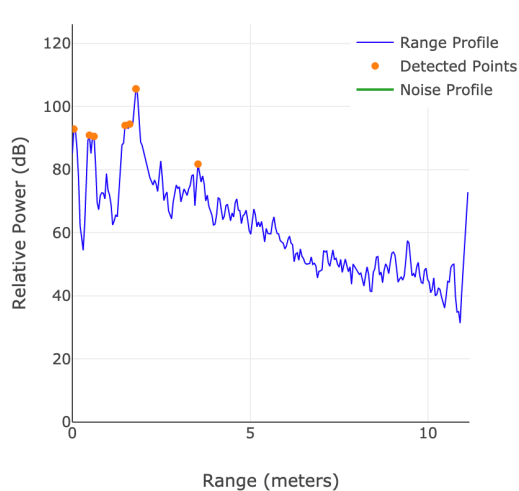


Figura 2.39. *Range Profile for zero Doppler* (escala logarítmica)

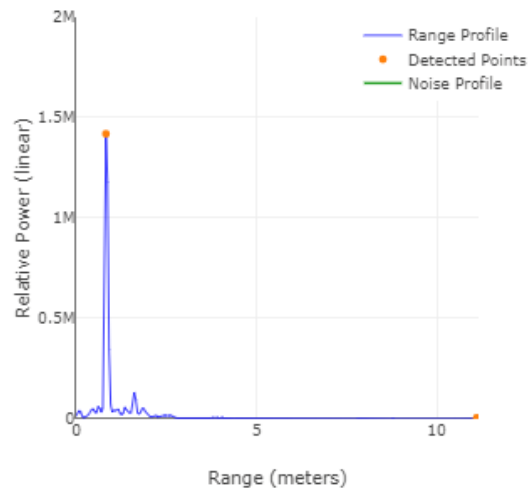


Figura 2.40. *Range Profile for zero Doppler* (escala lineal)

Aquest plot mostra el *range profile* al 0 Doppler (és a dir, objectes estàtics) mitjançant una línia blava i el *noise profile* (si està activat) mitjançant una línia verda. Per defecte, mostra els valors en logarítmic (es pot seleccionar escala lineal). Els objectes detectats es mostren com a punts de color taronja sobre la gràfica blava de *Range Profile*.

Taula 2.9. Representació dels plots de la GUI.

La GUI disposa d'un quadre de configuració que permet configurar els eixos X i Y de totes les gràfiques, els límits i la linealitat d'algunes. Cal prémer els botons *SENSOR STOP* i *RECORD START*, realitzant els canvis entremig, perquè es duguin a terme amb èxit (Figura 2.41).

Range Depth	10	Range Width	5	File Size Max (MB)	1
Range Profile Y max	2e6	☐ Range Profile Log Scale		Record time max (s)	10
SENSOR STOP				RECORD START	

Figura 2.41. Quadre de configuració de les gràfiques.

Per acabar, al realitzar els plots també mostra els anomenats *Display Parameters*. En són tres: Profiling, Chirp/Frame i Scene. Queden detallats a la Taula 2.10, Taula 2.11 i Taula 2.12.

Profiling parameters	
ProfilingChirp/FrameScene Platform0xa6843 SDK Version3.4.0.2 Number of Detected Objects5 Frame stats ...InterChirpProcessingMargin (usec)0 ...InterFrameProcessingMargin (usec)76033 ...InterFrameProcessingTime (usec)3948 ...TransmitOutputTime (usec)8707 ...Active/Interframe CPU Load (%)0/4 Detailed Temperature Report@time=0 days 00:25:37.575 ...@Rx channels (Deg C)43C, 43C, 43C, 44C ...@Tx channels (Deg C)44C, 45C, 46C ...@Pm, @Dig0, @Dig1 (Deg C)45C, 48C, 48C	
Figura 2.42. Aparença del quadre 'Profiling'.	
Platform	La cadena hexadecimal que correspon a l'output del RADAR.
SDK version	La versió del software de l'SDK que està utilitzant el RADAR.
Number of Detected Objects	Un camp que es va actualitzant contínuament (nova mesura en cada frame) que mostra el nombre d'objectes detectats per un frame donat.
Frame Stats o Stats for Subframe	En cas de la configuració avançada del frame, aquest paràmetre llista els <i>subframes</i> (separats per <i>command</i>) on cadascun dels següents estats són mostrats. Tots els camp es van actualitzant contínuament (nova mesura en cada frame). <i>InterChirpProcessingMargin</i>: mostra el temps (µs) emprat en cada duració d'<i>chirp</i> i <i>interchirp</i> després del chirp processing. <i>InterFrameProcessingMargin</i>: mostra el temps (µs) emprat en un processament de <i>frame/subframe</i> després que el processament de tots els <i>frames/subframes</i> sigui completat. <i>InterFrameProcessingTime</i>: mostra el temps (µs) que ha estat consumit pel processament de <i>frame/subframe</i> després que els <i>chirps</i> actius siguin completats. <i>TransmitOutputTime</i>: mostra el temps (µs) perquè la sortida sol·licitada (controlada mitjançant les seleccions en la secció <i>Plot Selection</i> de la finestra <i>Configure</i>) sigui enviada a través de la UART des del dispositiu mmWave al PC.

	• <i>Active/Interframe CPU Load (%)</i>: mostra el percentatge <i>CPU load</i> durant el temps de <i>chirping</i> actiu dins el <i>frame/subframe</i> i el temps d'<i>interframe/intersubframe</i>. Aquests valors també es reflecteixen en la gràfica de <i>CPU load</i> pel <i>frame</i>, o el primer <i>subframe</i> seleccionat al <i>guiMonitor command</i>. • <i>Temperature Report</i>: mostra l'informe de la temperatura del dispositiu des de varis sensors de temperatura. El temps mostrat és el nombre de dies/hores/minuts/segons/milisegons des que el dispositiu es va encendre.
--	--

Taula 2.10. Configuració dels 'Profiling Parameters'.

Chirp/Frame Parameters																				
Profiling	Chirp/Frame	Scene																		
<table><tr><td>Start Frequency (Ghz)</td><td>77</td></tr><tr><td>Slope (MHz/us)</td><td>70</td></tr><tr><td>Samples per chirp</td><td>256</td></tr><tr><td>Chirps per frame</td><td>32</td></tr><tr><td>Sampling rate (Msps)</td><td>5.2090</td></tr><tr><td>Sweep Bandwidth (GHz)</td><td>3.4402</td></tr><tr><td>Frame periodicity (msec)</td><td>100</td></tr><tr><td>Transmit Antennas</td><td>2</td></tr><tr><td>Receive Antennas</td><td>4</td></tr></table>			Start Frequency (Ghz)	77	Slope (MHz/us)	70	Samples per chirp	256	Chirps per frame	32	Sampling rate (Msps)	5.2090	Sweep Bandwidth (GHz)	3.4402	Frame periodicity (msec)	100	Transmit Antennas	2	Receive Antennas	4
Start Frequency (Ghz)	77																			
Slope (MHz/us)	70																			
Samples per chirp	256																			
Chirps per frame	32																			
Sampling rate (Msps)	5.2090																			
Sweep Bandwidth (GHz)	3.4402																			
Frame periodicity (msec)	100																			
Transmit Antennas	2																			
Receive Antennas	4																			

Figura 2.43. Aparència del quadre 'Chirp/Frame'.

Quan es selecciona aquest display *tab*, el paràmetres de *chirp/frame* prenen el valor de la configuració real que s'ha enviat al dispositiu i són mostrats en aquest quadre.

Taula 2.11. Configuració dels 'Chirp/Frame Parameters'.

Scene Parameters																			
<table><tr><th>Profiling</th><th>Chirp/Frame</th><th>Scene</th></tr><tr><td colspan="2">Range resolution (m)</td><td>0.0436</td></tr><tr><td colspan="2">Max Unambiguous Range (m)</td><td>8.9297</td></tr><tr><td colspan="2">Max Radial Velocity (m/s)</td><td>1.0018</td></tr><tr><td colspan="2">Radial Velocity Resolution (m/s)</td><td>0.1252</td></tr><tr><td colspan="2">Azimuth Resolution (Deg)</td><td>14.5</td></tr></table>		Profiling	Chirp/Frame	Scene	Range resolution (m)		0.0436	Max Unambiguous Range (m)		8.9297	Max Radial Velocity (m/s)		1.0018	Radial Velocity Resolution (m/s)		0.1252	Azimuth Resolution (Deg)		14.5
Profiling	Chirp/Frame	Scene																	
Range resolution (m)		0.0436																	
Max Unambiguous Range (m)		8.9297																	
Max Radial Velocity (m/s)		1.0018																	
Radial Velocity Resolution (m/s)		0.1252																	
Azimuth Resolution (Deg)		14.5																	

Figura 2.44. Aparència del quadre 'Scene'.

Quan es selecciona aquesta finestra de visualització, els *scene parameters* es calculen a partir de la configuració real que s'ha enviat al dispositiu. Aquests valors haurien de coincidir, aproximadament, amb els del *Configure tab* si el botó *SEND CONFIG TO MMWAVE DEVICE* s'ha utilitzat per configurar el dispositiu.

Taula 2.12. Configuració dels 'Scene Parameters'.

2.5. Adquisició i representació de dades amb Matlab.

A continuació, s'explicarà com, a partir de *Matlab*, s'obtenen les dades detectades amb el RADAR. S'han desenvolupat 3 programes auxiliars per ser utilitzats en els programes principals:

- *RADARSetup18XX.m*
- *readAndParseData18XX.m*
- *objectComponentsExists.m*

Els dos primers van ser extrets i modificats de treballs desenvolupats prèviament [12], mentre que el tercer va ser de creació pròpia.

Aquests programes serveixen per enviar un Fitxer de Configuració amb les comandes CLI al RADAR i, després, iniciar la captura i recollir les dades que va mesurant mitjançant la UART. L'execució que segueixen és la del diagrama de flux que es pot veure a la Figura 2.45.

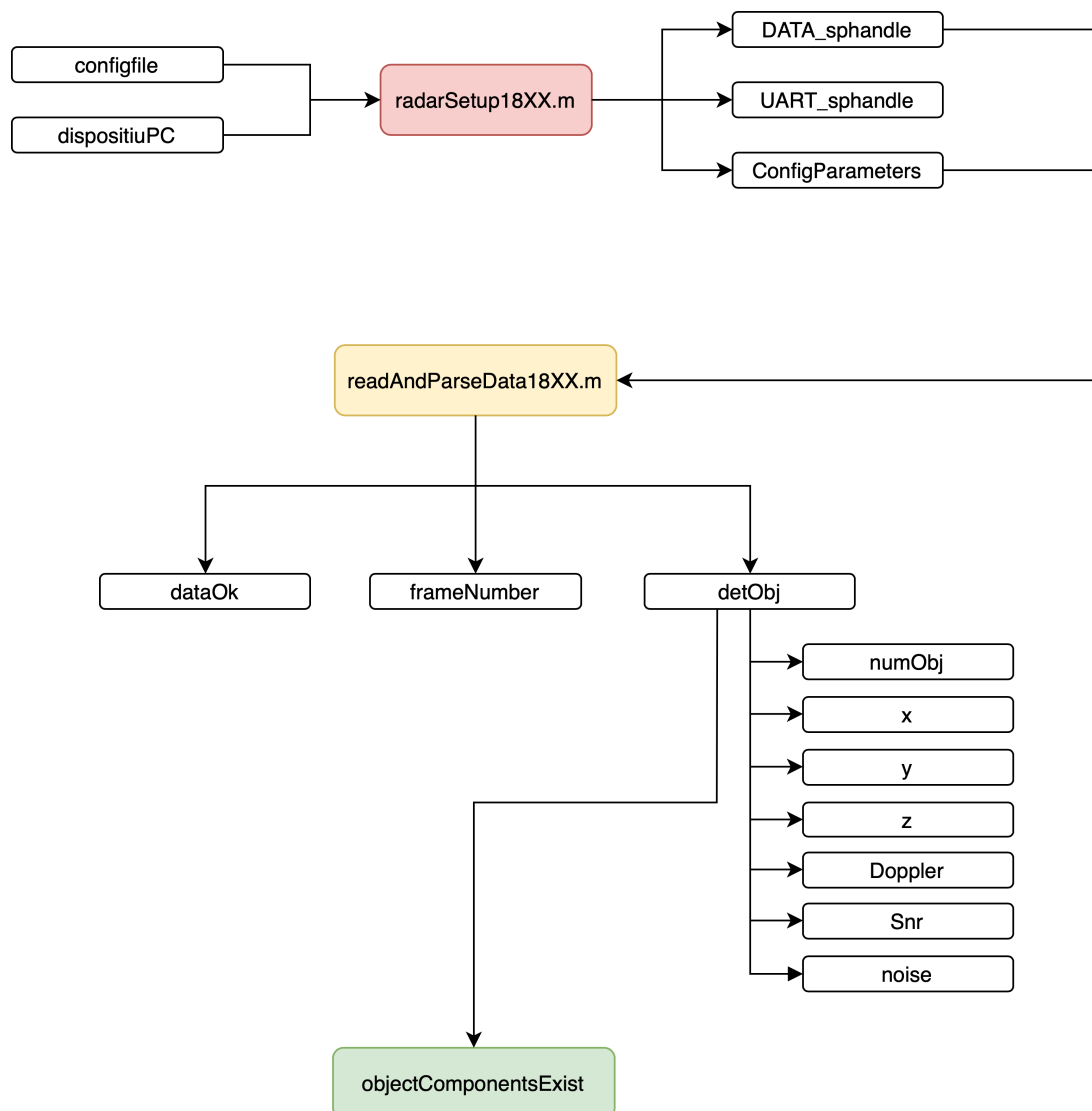


Figura 2.45. Diagrama de flux sobre la captura de dades amb el RADAR.

2.5.1. Programa *RADARSetup18XX.m*.

El primer de tots, *RADARSetup18XX.m*, serveix per seleccionar els ports de comunicació (UART i DATA), en funció del dispositiu que se li indiqui que s'està utilitzant (*MacBook* o *Windows*), la *BaudRate* de cadascun i enviar les comandes CLI al RADAR a partir de la lectura del Fitxer de Configuració. Conté la següent funció del Codi 2.1.

```
[DATA_sphandle, UART_sphandle, ConfigParameters] =  
RADARSetup18XX(configfile, dispositiuPC)
```

Codi 2.1. Funció *RADARSetup18XX*.

A aquesta funció se li passa per paràmetre el *Path* del fitxer de configuració (*configfile*) i el dispositiu PC (*dispositiuPC*) que s'està utilitzant. La funció ja selecciona el port correcte en funció del dispositiu PC que s'utilitza ja que, com s'ha vist anteriorment, és diferent. Els paràmetres que retorna són:

- *DATA_sphandle*: *serial port object* amb el Port que correspon a DATA i la seva velocitat de transmissió de dades.
- *UART_sphandle*: *serial port object* amb el Port que correspon a la UART i la seva velocitat de transmissió de dades. S'utilitza en aquest mateix fitxer per enviar al RADAR les dades del fitxer de configuració.
- *ConfigParameters*: objecte amb les dades llegides del fitxer de configuració.

2.5.2. Programa *readAndParseData18XX.m*.

El segon fitxer, *readAndParseData18XX.m*, conté la funció que permet obtenir els punts capturats mitjançant el RADAR. Recull les dades que va mesurant mitjançant la UART. És la funció del Codi 2.2:

```
[data0k, frameNumber, detObj] = readAndParseData18XX(DATA_sphandle,  
ConfigParameters)
```

Codi 2.2. Funció *readAndParseData18XX*.

Com es pot observar, se li passen per paràmetre dos dels objectes obtinguts en la funció de l'anterior programa: *DATA_sphandle*, que servirà per la comunicació de dades entre el RADAR i el programa Matlab, i *ConfigParameters*.

Els paràmetres que retorna són els mostrats en la Taula 2.13.

<i>dataOk</i>	Si les dades s'han llegit i retornat correctament, retorna un '1'. En cas contrari, un '0'.
<i>frameNumber</i>	Correspon al número de frame. No s'utilitza al programa principal.
<i>detObj</i>	Objecte que conté els paràmetres retornats llegits pel RADAR. Aquests són: • <i>numObj</i>: nombre d'objectes detectats. • <i>x</i>: coordenada x de l'objecte detectat (és un vector de punts que conté totes les coordenades x dels objectes detectats, en cas de ser més d'un). • <i>y</i>: coordenada y de l'objecte detectat (és un vector de punts que conté totes les coordenades y dels objectes detectats, en cas de ser més d'un). • <i>z</i>: coordenada z de l'objecte detectat (és un vector de punts que conté totes les coordenades z dels objectes detectats, en cas de ser més d'un). Aquest paràmetre és sempre 0 ja que el RADAR només retorna punts en el pla horitzontal (en la direcció de l'agrupació d'antenes virtual). • <i>Doppler</i>: velocitat de l'objecte detectat (és un vector de punts que conté tots els doppler dels objectes detectats, en cas de ser més d'un). Pot ser negativa o positiva. • <i>Snr</i>: <i>signal to noise ratio</i> de l'objecte detectat (és un vector de punts que conté tots els snr dels objectes detectats, en cas de ser més d'un). Es retorna multiplicat per un factor 10 i, per tant, per tenir l'SNR correcte cal dividir-lo entre 10. • <i>noise</i>: soroll de l'objecte detectat (és un vector de punts que conté tots els <i>noise</i> dels objectes detectats, en cas de ser més d'un).

Taula 2.13. Paràmetres retornats per la funció readAndParseData18XX.m.

El paràmetre de més interès és *detObj*. Serà aquell a partir del qual s'obtindran les dades dels punts detectats al programa principal. Com que és una variable-objecte, conté la informació de tots els paràmetres descrits anteriorment. Així doncs, per exemple, per obtenir la coordenada 'x' es pot fer de la següent manera: *detObj.x*.

2.5.3. Programa *objectComponentsExists.m*.

El tercer fitxer, *objectComponentsExists.m*, conté una funció (Codi 2.3) per tal de comprovar si existeixen tots els components dels *detObj*.

```
dataOk = objectComponentsExist(detObj)
```

Codi 2.3. Funció *objectComponentsExist*.

Retorna un altre *dataOk*. D'aquesta manera, si existeixen tots els *detObj* d'interès (*x*, *y*, *doppler*, *snr*) *dataOk* pren valor d'1 i, en cas contrari, pren valor de '0'. És útil perquè el programa principal no doni un error de no existència d'un element.

2.5.4. Programa principal.

A continuació es detallarà com s'utilitzen les funcions per obtenir els punts en un programa principal i poder-los processar posteriorment.

En primer lloc, s'utilitza la funció *RADARSetup18XX* per obtenir els paràmetres *DATA_sphandle*, *UART_sphandle* i *ConfigParameters*. A continuació, es fa servir la funció *readAndParseData18XX*. D'aquí, s'obté *dataOk1* i els punts a retornar. El paràmetre *dataOk1* serà útil per determinar si s'han detectat punts o no. En cas afirmatiu, s'utilitzarà la funció *objectComponentsExist* per comprovar si totes les components s'han capturat correctament. Si és així, s'escriuran els punts en una matriu.

En algun cas, enlloc d'una matriu de punts, les dades s'han escrit en un fitxer de text per després tractar-les en altres programes. Les components que s'utilitzen són la *x*, *y*, *z*, *doppler* i *snr*. Per escriure-les en un fitxer de text, es pot fer de la manera que detalla el Codi 2.4.

```
fprintf(fileID, '%d %d %d %d %d %d %d %d\n', -detObj.x(i), detObj.y(i),  
detObj.z(i), detObj.doppler(i), detObj.snr(i),  
range(i), myInd, etime(clock, t));
```

Codi 2.4. Codi per escriure en un fitxer de text.

El programa principal que s'ha utilitzat per prendre mesures amb el RADAR s'anomena *prendreMesuresCasos.m*. Escriu les dades en un fitxer de text. Es fa servir per prendre mesures per l'aplicació d'ocupació a l'interior d'un vehicle, tal com es veurà en el següent capítol. És necessari definir un nom pel fitxer de text per poder-hi accedir. Per aquest motiu, el criteri que s'ha seguit és que el fitxer es digui 'dades' seguit d'una notació binària (1-ocupat/0-desocupat) de la forma: XXXX (pel cas de 4 seients) o XXXXX (pel cas de 5 seients), on X és un '1' si hi ha una persona o '0' si no hi ha ningú.

Finalment, per tal d'acabar l'execució del RADAR, s'utilitzen les instruccions del Codi 2.5.

```
fprintf(UART_sphandle, 'sensorStop');  
delete instrfind
```

Codi 2.5. Codi per acabar l'execució del RADAR.

2.5.5. Representació de les dades.

En primer lloc, es poden representar les dades a mesura que es van capturant a temps real, tal com es pot veure en la Figura 2.46.

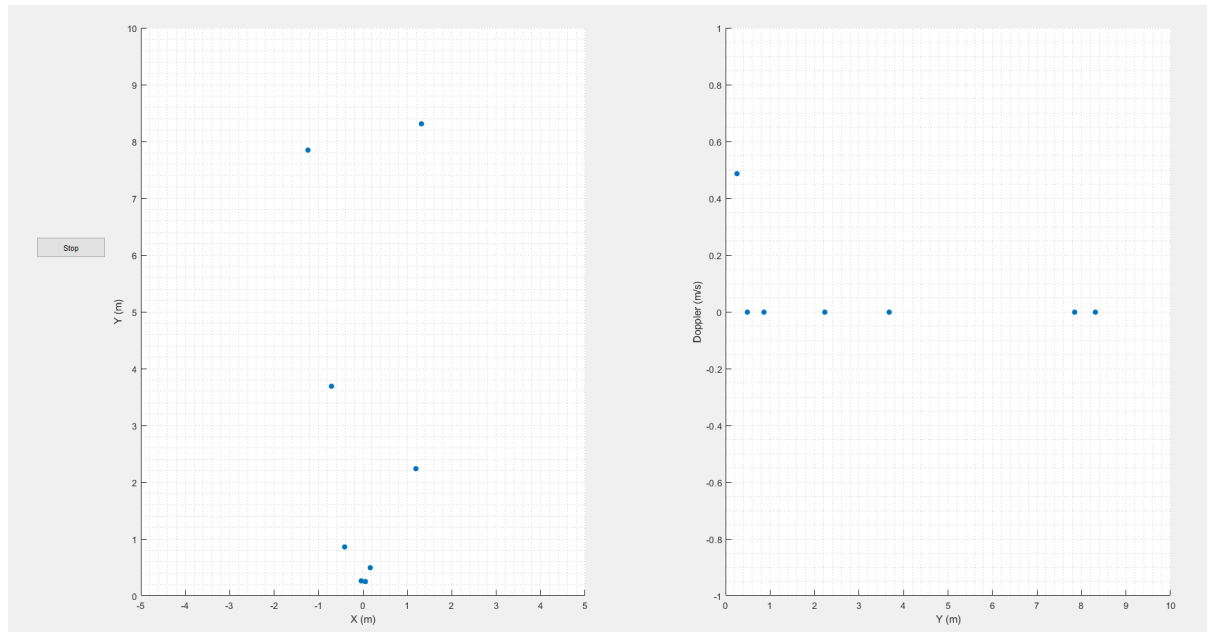


Figura 2.46. Representació de les dades capturades amb el RADAR a temps real (gràfica X-Y a la dreta, *Range-Doppler* a l'esquerra).

Quan es vol aturar l'execució del programa, es pot fer prement el botó 'Stop'.

Una vegada es tenen els punts capturats, és important poder visualitzar les dades tot representant-les gràficament. Per fer-ho, es té el programa *processData.m*, encarregat de llegir els punts en un fitxer de text i representar-les.

Per fer-ho, utilitza la funció *carregarDades*. Aquesta funció llegeix el fitxer de text i retorna els vectors de les dades corresponents. Seguidament, en fa dues representacions (*subplots*): la primera amb les coordenades X-Y i, la segona, amb el *Doppler-Range*. Aquest programa està dissenyat per l'aplicació d'ocupació en un vehicle, per la qual cosa les representacions correspondran a les captures en l'interior d'un vehicle (Figura 2.47, 2.48, 2.49, 2.50).

A més, s'ha fet el codi de manera que per la segona representació només es representen aquells punts amb un SNR superior al marcat pel programa.

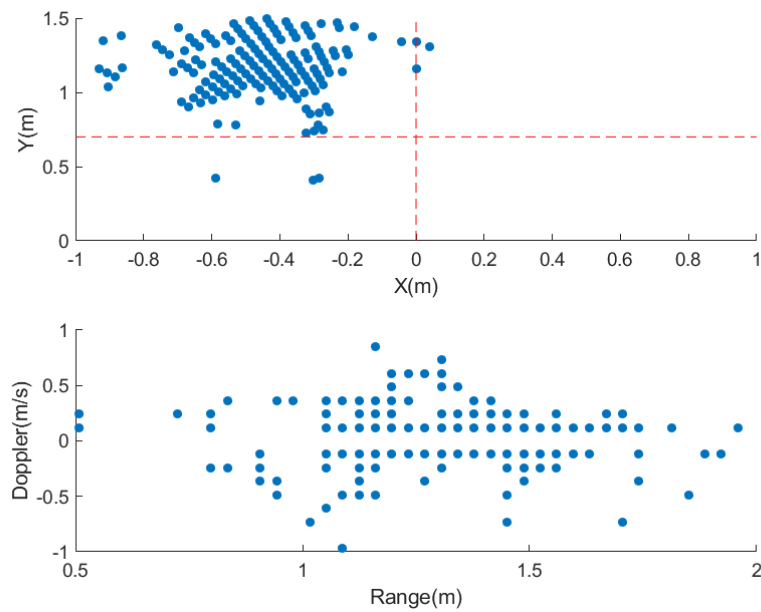


Figura 2.47. Com es pot observar en el primer *subplot*, la majoria de punts recauen sobre la zona de darrere a l'esquerra. En el segon *subplot*, es representa el doppler (m/s) en funció del temps.

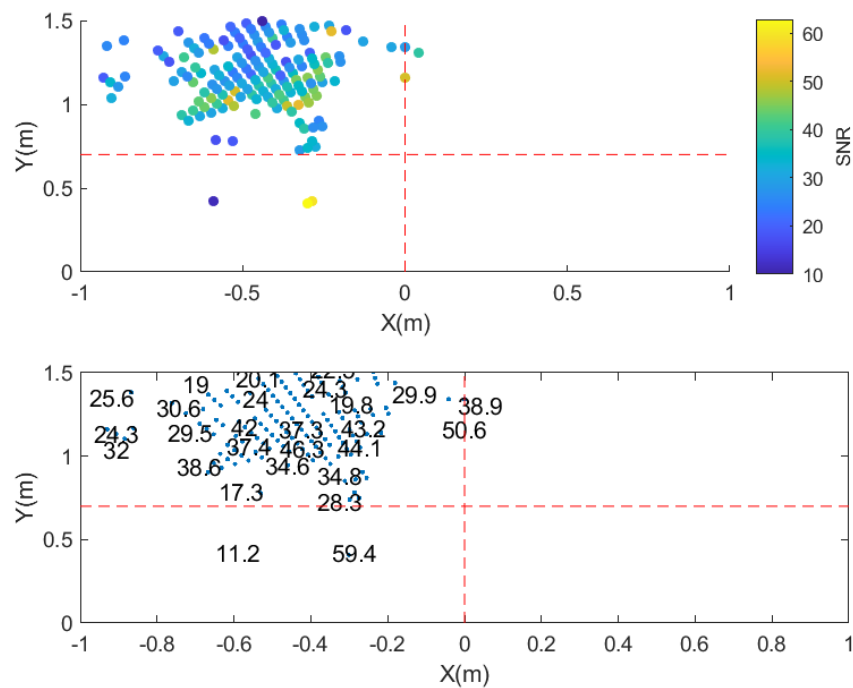


Figura 2.48. En aquesta segona representació, en el primer *subplot* es representen els punts amb el color del seu SNR. En el segon *subplot*, es posa el valor d'aquest SNR.

La funció *configureAxis* és l'encarregada d'ajustar els eixos i dibuixar les línies que separen les zones. En una segona versió de la funció, enlloc de dibuixar línies, es va optar per dibuixar l'àrea definida per cada zona i el seu centre, tal com pot veure's a continuació.

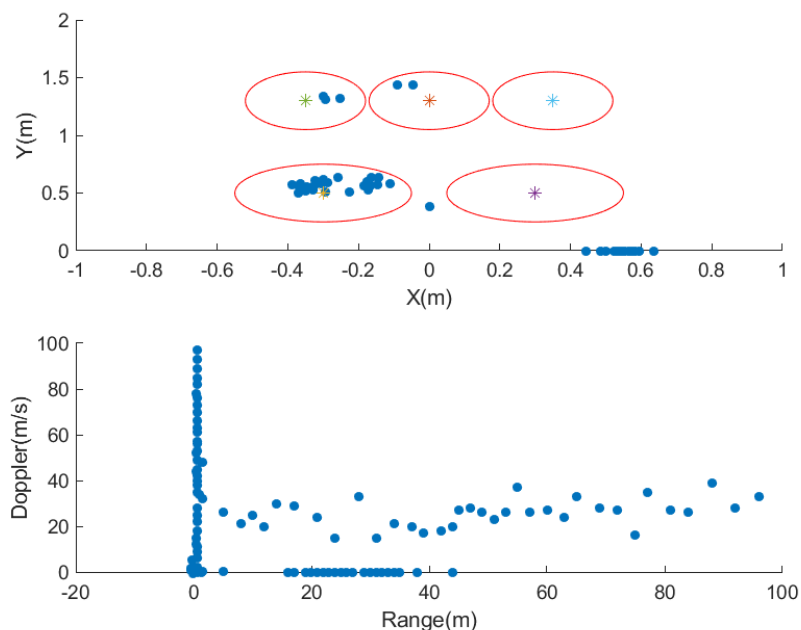


Figura 2.49. Com es pot observar en el primer *subplot*, la majoria de punts recauen la zona de davant a l'esquerra i alguns recauen darrere, tant al mig com a l'esquerra.

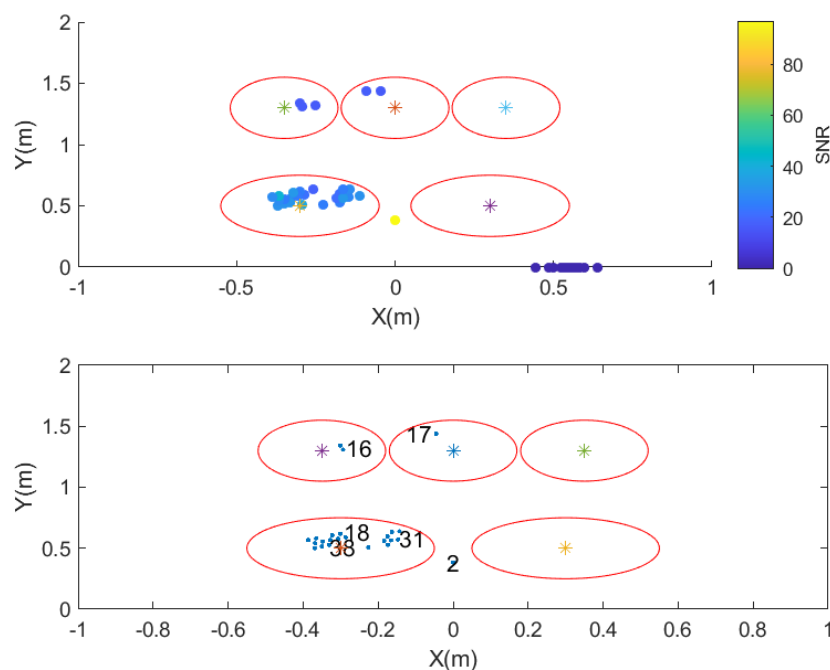


Figura 2.50. En aquesta segona representació, en el primer *subplot* es representen els punts amb el color del seu SNR. En el segon *subplot*, es posa el valor d'aquest SNR.

2.6. Referències.

- [1] CHEN, Victor C. The micro-Doppler effect in RADAR. Artech House, 2019.
- [2] Carlos del Amo Velasco. Trabajo de Fin de Grado. Análisis e implementación de un sistema de seguimiento y detección de elementos móviles en escenarios críticos de ámbito ferroviario basado en tecnología radar (2019). Universidad Politécnica de Madrid (UPM). Enllaç web: http://oa.upm.es/54113/1/TFG_CARLOS_DEL_AMO_VELASCO.pdf (consulta: 30 de març de 2021).
- [3] ZIEBINSKI, Adam, et al. Review of advanced driver assistance systems (ADAS). En AIP Conference Proceedings. AIP Publishing LLC, 2017. p. 120002.
- [4] ZHONG, Ziguo, et al. Camera RADAR fusion for increased reliability in adas applications. Electronic Imaging, 2018, vol. 2018, no 17, p. 258-1-258-4.
- [5] GADRINGER, Michael Ernst, et al. RADAR target stimulation for automotive applications. IET RADAR, Sonar & Navigation, 2018, vol. 12, no 10, p. 1096-1103.
- [6] MOLCHANOV, Pavlo, et al. Multi-sensor system for driver's hand-gesture recognition. En 2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG). IEEE, 2015. p. 1-8.
- [7] Eloi Guerrero - Menéndez. Frequency-Modulated Continuous-Wave Radar in Automotive Applications (2018). Universitat Autònoma de Barcelona (UAB). Enllaç web: https://ddd.uab.cat/pub/tfg/2018/195050/TFG_eloiguerreromenendez.pdf (consulta: 30 de març de 2021).
- [8] Texas Instruments. AWR1642 Single-Chip 77- and 79- GHz FMCW Radar sensor. Ref: SWRS203B. Data del *Datasheet*: maig de 2017 (revisat l'abril de 2020). Enllaç web: <https://ja.cat/AWR1642> (consulta: 15 de març de 2021).
- [9] TI DevTools. Enllaç web: <https://dev.ti.com/> (consulta: 15 de març de 2021).
- [10] mmWave Demo Visualizer. Enllaç web: https://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/3.5.0/ (consulta: 15 de març de 2021).
- [11] Texas Instruments. mmWave Demo Visualizer. User's Guide. Ref: SWRU529C. Data de l'*User's Guide*: maig de 2017 (revisat l'abril de 2020). Enllaç web: <https://www.ti.com/lit/ug/swru529c/swru529c.pdf?ts=1625810534719> (consulta: 15 de març de 2021).
- [12] A Matlab reading program for TI AWR1642. Github. Enllaç web: https://github.com/lynnlilu/mmWave_reading (consulta: 1 d'abril de 2021).

CAPÍTOL III

DETECCIÓ DE L'OCUPACIÓ EN VEHICLES

En aquest capítol s'explicarà, en primer lloc, l'estat de l'art de l'aplicació sobre detecció de l'ocupació en vehicles (tecnologies, mètodes i mercat actual). Seguidament, es detallarà l'aplicació realitzada amb *Matlab*, tot explicant els passos seguits des de la captura de mostres en un vehicle fins el seu processament per determinar-ne l'ocupació. També es presentarà la interfície gràfica realitzada per seleccionar els paràmetres bàsics d'una captura (característiques del vehicle i dispositius utilitzats) i l'aplicació realitzada amb *Raspberry Pi* i *Python*. Finalment, es presentaran els resultats obtinguts mitjançant diverses matrius de confusió i es mostrarà la precisió de l'algorisme realitzat.

3.1. Estat de l'art.

S'han realitzat diversos estudis sobre aquest tòpic. Un detector d'ocupació de seients és una part indispensable del sistema de seguretat d'airbag, que protegeix els usuaris de danys greus. En la seva forma més simple, informa sobre l'estat (ocupat/vacant) d'un seient a la unitat de control de l'airbag. Per tant, és important tenir un control precís sobre l'estat d'ocupació de tots els seients d'un vehicle; un sistema que pugui presentar errors (informar sobre un seient ocupat quan realment no ho està, o a l'inrevés), pot portar a errors mecànics i accidentals pels usuaris de dins el vehicle.

Els sistemes més habituals fins el moment són els basats en mètodes capacitius. Són eficients, però requereixen d'elèctrodes que han d'estar situats a la capa superficial de les zones del seient i del respall del seient. El sensor presentat en la referència [1], un sensor inductiu-capacitiu combinat, utilitza una estructura d'elèctrode simple i es situa sota l'escuma del seient. Aquestes característiques permeten un sensor més econòmic ja que pot ser fàcilment fabricat i instal·lat. Combina principis sensorials de proximitat capacitius i inductius. El sensor detecta la presència d'un ocupant aprofitant l'efecte d'apantallament del camp elèctric, mentre que la seva funció de proximitat inductiva detecta la presència d'objectes conductors (per exemple, un ordinador portàtil) que poden estar col·locats en el seient i ajuda a aconseguir una detecció fiable de l'ocupació. El sistema de mesura utilitza una unitat de condicionament de la senyal basada en el principi de la freqüència portadora.

Un altre mètode, que és l'utilitzat en aquest Treball de Fi de Grau, és mitjançant la disposició d'un RADAR a l'interior del vehicle. Els mètodes que es descriuen a continuació basen el seu principi en la captura de dades mitjançant un RADAR.

En la referència [2] s'utilitza una tècnica de localització de persones a l'interior del vehicle utilitzant un model de xarxa neuronal profunda (DNN) que s'entrena amb dades experimentals. En primer lloc, s'instal·la un RADAR de ràdio d'impulsos de banda ultra-ampla (IR-UWB) a l'interior del vehicle i s'adquireixen els senyals rebuts canviant la disposició de les persones assegudes. A continuació, sobre les dades adquirides, s'aplica la DNN per entrenar un classificador, que pot predir si una persona està asseguda o no en cada seient. Per dissenyar una xarxa adequada, s'avalua el rendiment canviant el tipus de funció d'activació, el nombre de capes i el nombre de nodes en cada capa oculta de la DNN. L'avantatge d'aquest mètode és que no és necessari extreure característiques d'una senyal de RADAR determinada.

En la referència [3] es presenta un sistema capaç de detectar respiracions en individus (taxa respiració) i objectes d'alta precisió. També s'utilitza un RADAR de ràdio d'impuls de banda

ultra-ampla (IR-UWB). Aquests tipus de RADAR es caracteritzen per enviar polsos de duració molt curta per tenir molta resolució temporal i en distància i, per tant, requereixen d'un gran ample de banda (entre 3.1-10 GHz). El sistema hardware consisteix en un circuit integrat (IC) correlador generador d'impulsos en el transmissor i un IC correlador amb un circuit de banda base integrador com a receptor de la correlació. El pols radiat s'aproxima a un cinquè impuls gaussià derivat amb duració típica $\sigma=51$ ps, utilitzant eficaçment la màscara interior de la FCC (entre 3.1 i 10.6 GHz), que correspon a la banda freqüencial UWB. Pel seguiment, el transmissor irradia un tren d'impulsos i el receptor en recull les reflexions dels objectes que es troben dins del sensor. Amb els senyals reflectits, es calcula una correlació de hardware contínua mitjançant una correlació d'impulsos d'escombrat. La correlació s'aplica per evitar el mostreig de l'impuls d'RF amb una precisió de piconsegons.

En la referència [3], la precisió del sensor es demostra mitjançant una mesura amb una placa metàl·lica d'1 mm de desviació sinusoidal. Unes altres mesures verifiquen l'ús del principi demostrat com sensor de respiració. S'han detectat les senyals de respiració d'homes i d'un nadó de set setmanes, cosa que permet qualificar el principi del RADAR IR-UWB com una eina útil per la determinació de la freqüència respiratòria.

No obstant, els RADAR UWB es caracteritzen normalment per un curt abast en comparació amb els RADAR FMCW (com l'utilitzat en aquest projecte). La tecnologia UWB ha despertat un gran interès en aplicacions de localització (especialment en interiors i esports) donada la gran resolució en distància i baix consum tot utilitzant tècniques de localització amb diferències en el temps d'arribada.

En la referència [4] s'hi tracta un sistema RADAR de quatre canals a la banda de 24 GHz SRD (ISM) amb un ample de banda de 250 MHz. S'utilitza per la detecció de persones assegudes als seients de darrere d'un vehicle. S'aplica un *array* d'antenes amb un patró de radiació dedicada. Es basa en l'aproximació a la modulació FMCW i per això és possible mesurar el rang i el senyal Doppler degut al moviment del passatger al darrere del cotxe.

Utilitza tècniques de *beamforming* (DBF) amb els quatre canals receptors per tal de determinar l'angle d'arribada. Per això, és possible determinar la localització del seient del passatger. L'adquisició de dades es duu a terme amb un ADC de *National Instruments* i, el processat i avaluació dels resultats, amb *Matlab*. Aquest RADAR és similar a l'utilitzat en el projecte però aquest últim té major resolució degut a què utilitza un major ample de banda (bandes freqüencials 76-22 GHz i 77-81 GHz).

Nota: la resolució en distància (m) es calcula de la forma $2c/B$, essent c la velocitat de la llum en el buit i B l'ample de banda. A major ample de banda B , augmenta la resolució. Per tant, la resolució del RADAR utilitzat en aquest projecte ($B = 4$ GHz) és major que la resolució del RADAR desenvolupat en [4] ($B = 250$ MHz).

3.2. Processat del senyal.

L'objectiu que es vol aconseguir és la detecció del nombre d'ocupants en un vehicle de quatre i cinc places i quina és aquesta posició aproximada. En aquest punt es detallaran els càlculs teòrics emprats per realitzar l'algorisme d'ocupació.

En primer lloc, es parteix de les coordenades (x,y) on estan situats els centres dels seients del vehicle.

En segon lloc, partint dels punts que ha capturat el RADAR, s'agafen només aquells compresos en un radi definit més propers a cada seient o zona. Aquests punts entren dins la zona de cada seient en un vector de punts.

Seguidament, es duu a terme el càlcul de la desviació típica. Pels punts de cada zona, es realitza:

$$\sigma = \sqrt{\sum_{i=1}^{N_{punts}} (X_i - E(X))^2 + (Y_i - E(Y))^2} \quad (3.1)$$

On:

- N_{punts} són el total de punts que hi ha dins el radi de cada zona.
- X_i i Y_i correspon al punt retornat pel RADAR i del vector de punts de la coordenada X i Y, respectivament, de cada zona.
- $E(X)$ i $E(Y)$ és el *mean value*, el valor mig de cada vector de punts.

A continuació, es normalitza la desviació típica, tot obtenint la desviació típica ponderada, multiplicant el valor obtingut anteriorment pel nombre total de punts de la zona (tant el vector X com Y tenen el mateix nombre de punts):

$$\sigma_{ponderada} = \sigma \cdot N_{punts_{zona}} \quad (3.2)$$

Finalment, es divideix la desviació típica ponderada pel nombre total de punts de totes les zones, obtenint així la desviació típica final:

$$\sigma_{final} = \frac{\sigma_{ponderada}}{N_{punts_{total}}} \quad (3.3)$$

Partint d'aquest resultat final, en el següent punt s'explicaran els criteris de decisió a l'hora de triar si hi ha un ocupant o no en un determinat seient.

Cal notar que la divisió $N_{punts,zona}/N_{punts,total}$, presents a les equacions (3.2) i (3.3), respectivament, representa la fracció de punts en cada zona i, per tant, la densitat de punts en cada zona. En cas que el seient estigui buit en aquesta zona, hi ha molts pocs punts i, sovint, aquests estan molt dispersos, de manera que la desviació ponderada pren un valor baix. En canvi, si el seient està ocupat es rep un núvol de punts bastant concentrat i la desviació dóna una idea de l'extensió de l'objectiu en qüestió; quan es correspon a una persona pren un valor elevat.

Tal com s'explica a continuació, aquestes paràmetres permetran definir unes regles de decisió per determinar si un seient està buit o ocupat.

3.3. Anàlisi dels resultats obtinguts.

Per tal de realitzar els càlculs explicats en el punt anterior mitjançant *Matlab*, s'han creat tres funcions:

- *training.m*: es duen a terme la definició de les zones (centre, radi i àrea). Es criden les dues funcions que es detallen seguidament (*zonepoints.m*, *calculq.m*).
- *zonepoints.m*: obtenció dels vectors amb els punts capturats de cada zona. Retorna tres objectes, un per la coordenada X, un altre per la Y i un últim per l'SNR, que contenen els vectors de punts de cada zona (Xz, Yz i Sz, contenint 4 o 5 vectors de punts en funció si el vehicle s'ha indicat que és de 4 o 5 places). Per exemple, l'objecte Xz pel vehicle de 4 places seria com el de la Taula 3.1.

Xz			
vector punts eix X - Zona 1	vector punts eix X - Zona 2	vector punts eix X - Zona 3	vector punts eix X - Zona 4

Taula 3.1. Objecte Xz amb els vectors de punts de l'eix X pel vehicle de 4 places.

- *calculq.m*: càlcul de les desviacions explicades en el punt anterior. Retorna la desviació típica final i el nombre total de punts per zona.

Mitjançant aquestes funcions s'han construït els programes principals que es descriuran en els següents punts. S'ha considerat no detallar el codi d'aquestes funcions ja que no presenten cap peculiaritat a destacar en aquesta memòria i l'explicació donada és suficient.

3.3.1. Definició de les zones del vehicle.

Per la definició de zones, s'han tingut en compte les distàncies X i Y tant als seients de darrere i davant, i dos radis, un corresponent a la zona de davant i un altre de darrere, tal com es detalla en la Taula 3.2.

Distància X seient davant (m)	Correspon a la distància que hi ha del centre del RADAR al seient dret o esquerra davanters en l'eix X.
Distància Y seient davant (m)	Correspon a la distància que hi ha del centre del RADAR al seient dret o esquerra davanters en l'eix Y.
Distància X seient darrere (m)	Correspon a la distància que hi ha del centre del RADAR al seient dret o esquerra posteriors en l'eix X.
Distància Y seient darrere (m)	Correspon a la distància que hi ha del centre del RADAR al seient dret o esquerra posteriors en l'eix Y.
Radi Davant (m)	Correspon al radi de la zona de davant, amb alguns matisos que es detallaran a continuació.
Radi Darrere (m)	Correspon al radi de la zona de darrere, amb alguns matisos que es detallaran a continuació.

Taula 3.2. Definició dels paràmetres que indiquen les dimensions del vehicle.

Per tal que les mesures s'ajustessin el millor possible al cotxe, les següents dues figures (Figura 3.1, 3.2) il·lustren com s'han definit les zones corresponents als seients tant per un cotxe de quatre places com de cinc.

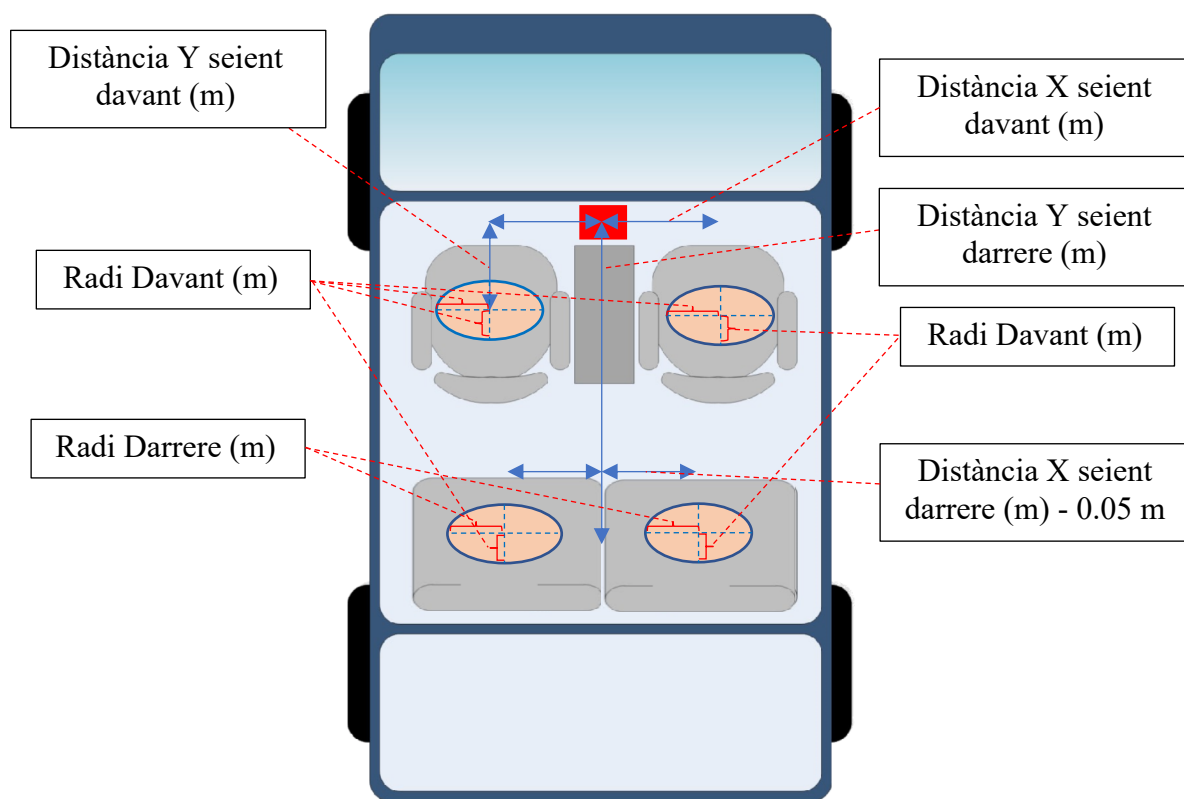


Figura 3.1. Il·lustració de les zones i paràmetres que defineixen les dimensions del vehicle de 4 places.

Cal notar que es va utilitzar com a radi de l'eix Y el de davant per les quatre zones, mentre que es va utilitzar el corresponent a la zona pel de l'eix X. Es va fer de manera que l'àrea s'ajustés a la mida d'una persona. A més, es van ajustar les zones dels seients de darrere de manera que tinguessin més visibilitat pel RADAR, tot essent desplaçades més cap al centre en comparació a les del davant.

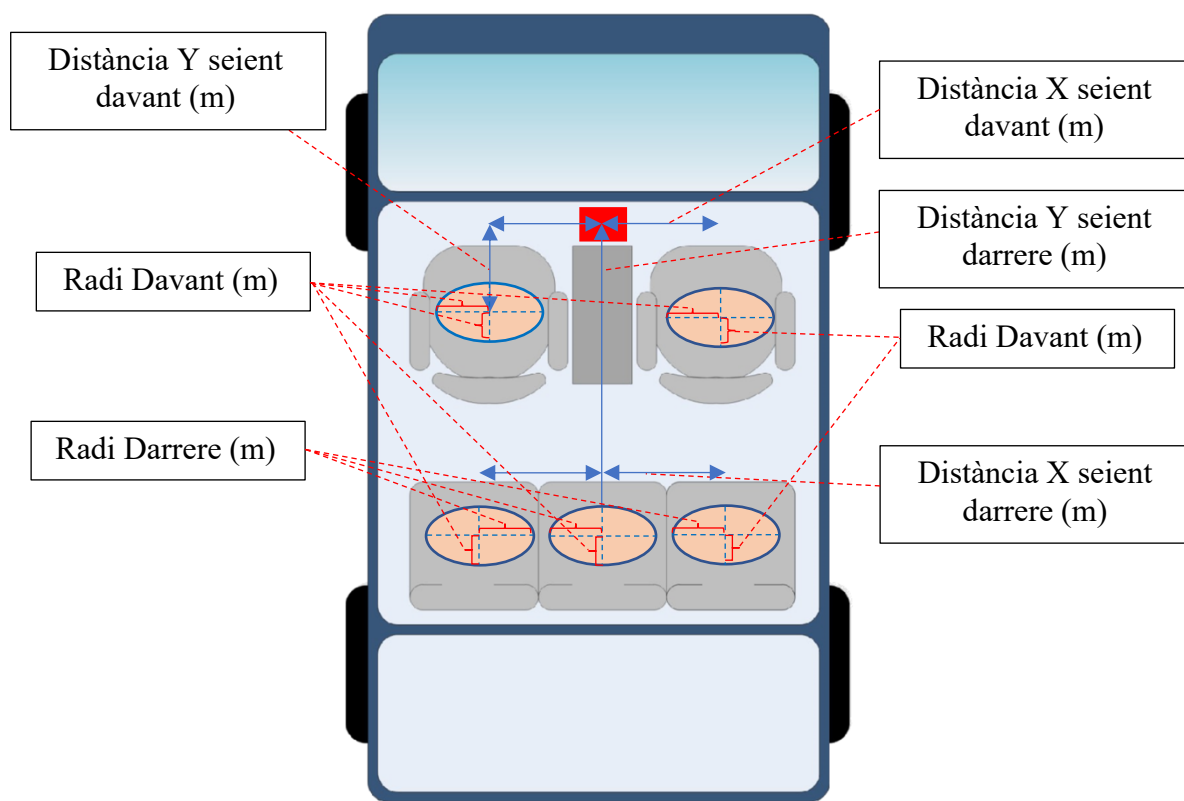


Figura 3.2. Il·lustració de les zones i paràmetres que defineixen les dimensions del vehicle de 5 places.

En aquest cas, es va també es va optar per agafar com a radi de l'eix Y el de davant per les cinc zones, mentre que es va utilitzar el corresponent a la zona pel de l'eix X.

Per tal de poder configurar ràpidament les especificacions del cotxe, es va generar un fitxer de text anomenat *carSpecs.txt*. Aquest pot ser llegit des d'un programa principal, de manera que es poden carregar les dades a les variables del programa. Les especificacions del fitxer que han estat utilitzades per aquest projecte són les de la Taula 3.3.

Vehicle de quatre places	
<i>NumPlaces</i> Nombre de places.	4
<i>DistX_RADAR_centre_seient_davant(m)</i> Distància en l'eix X del RADAR al centre d'un dels dos seients de davant.	0.3
<i>DistX_RADAR_centre_seient_darrere(m)</i> Distància en l'eix X del RADAR al centre d'un dels dos seients de darrere.	0.3
<i>DistY_RADAR_centre_seient_Dav(m)</i> Distància en l'eix Y del RADAR al centre d'un dels dos seients de davant.	0.6
<i>DistY_RADAR_centre_seient_Darr(m)</i> Distància en l'eix Y del RADAR al centre d'un dels dos seients de darrere.	1.4
<i>Radi_Davant(m)</i> Extensió del radi en cadascun dels seients de davant.	0.2
<i>Radi_Darrere(m)</i> Extensió del radi en cadascun dels seients de darrere.	0.2
Vehicle de cinc places	
<i>NumPlaces</i> Nombre de places.	5
<i>DistX_RADAR_centre_seient_davant(m)</i> Distància en l'eix X del RADAR al centre d'un dels dos seients de davant.	0.3
<i>DistX_RADAR_centre_seient_darrere(m)</i> Distància en l'eix X del RADAR al centre d'un dels dos seients de darrere.	0.4
<i>DistY_RADAR_centre_seient_Dav(m)</i> Distància en l'eix Y del RADAR al centre d'un dels dos seients de davant.	0.5
<i>DistY_RADAR_centre_seient_Darr(m)</i> Distància en l'eix Y del RADAR al centre d'un dels dos seients de darrere.	1.3
<i>Radi_Davant(m)</i> Extensió del radi en cadascun dels seients de davant.	0.25
<i>Radi_Darrere(m)</i> Extensió del radi en cadascun dels seients de darrere.	0.17

Taula 3.3. Dimensions utilitzades en els paràmetres pel vehicle de 4 i 5 places.

Al fitxer de text queden escrits tal com es pot veure en la Figura 3.3. És necessari que hi hagi un espai en cada banda del símbol '=' de manera que, a partir de la funció *carregarCarSpecs.m*, es puguin llegir i carregar correctament al programa principal. Aquesta funció defineix aquests paràmetres com a variables globals.

```
NumPlaces = 5
DistX_Radar_centre_seient_davant(m) = 0.3
DistX_Radar_centre_seient_darrere(m) = 0.4
DistY_Radar_centre_seient_Dav(m) = 0.5
DistY_Radar_centre_seient_Darr(m) = 1.3
Radi_Davant(m) = 0.25
Radi_Darrere(m) = 0.17
```

Figura 3.3. Aspecte visual del fitxer de text amb els paràmetres del vehicle de 5 places.

3.3.2. Càlcul de la resposta amb *Matlab*.

La funció utilitzada per realitzar els càlculs finals ha estat la funció *calculPosicions.m*. A partir d'aquesta, s'ha obtingut el resultat final tot indicant en un vector lògic quines posicions estan ocupades ('1') i quines no ho estan ('0').

En primer lloc, es multiplica la desviació final obtinguda (*Out1*) pel nombre de punts de cada zona (*Out2*), obtenint així un vector de tantes posicions com seients se li hagi indicat. Aquestes dues dades provenen de la funció *calculq.m* i *training.m*, explicades anteriorment. El vector resultant s'anomena *OutFINAL* (Codi 3.1).

```
OutFINAL=(Out1.*Out2);
```

Codi 3.1. Obtenció del vector *OutFINAL*.

Seguidament, es divideix aquest vector entre la suma del contingut de totes les posicions d'aquest mateix vector, tal com es pot veure a continuació. D'aquesta manera, es normalitzen els valors i sempre es té el mateix marge de variació.

```
suma = sum(OutFINAL);
OutFINAL=OutFINAL/suma;
```

Codi 3.2. Normalització del vector *OutFINAL*.

Finalment, s'apliquen una sèrie de criteris o regles de decisió diferents en funció de la posició que s'està tractant.

- Pels seients de davant, que corresponen a les posicions 1 i 2 del vector, es considera que, si el valor és major a 1/200, el seient està ocupat ('1'). En cas contrari, estarà desocupat ('0')

```
OutFINAL(1)=OutFINAL(1)>=(1/200);
OutFINAL(2)=OutFINAL(2)>=(1/200);
```

Codi 3.3. Aplicació dels criteris als seients de davant.

- Pels seients de darrere, primer de tot s'arrodoneix a les mil·lèsimes el resultat i, posteriorment, es considera que si el valor és major a 1/500, el seient està ocupat ('1'). En cas contrari, estarà desocupat ('0').

```
OutFINAL(3)=round(OutFINAL(3),3)>=(1/500);
OutFINAL(4)=round(OutFINAL(4),3)>=(1/500);
```

Codi 3.4. Aplicació dels criteris als seients de darrere.

- Finalment, quan el vehicle té 5 places, es compara el valor 1/500 amb la plaça del seient del mig de darrere.

```
OutFINAL(5)=OutFINAL(5)>=(1/500);
```

Codi 3.5. Aplicació del criteri al seient del mig d'un vehicle de cinc places.

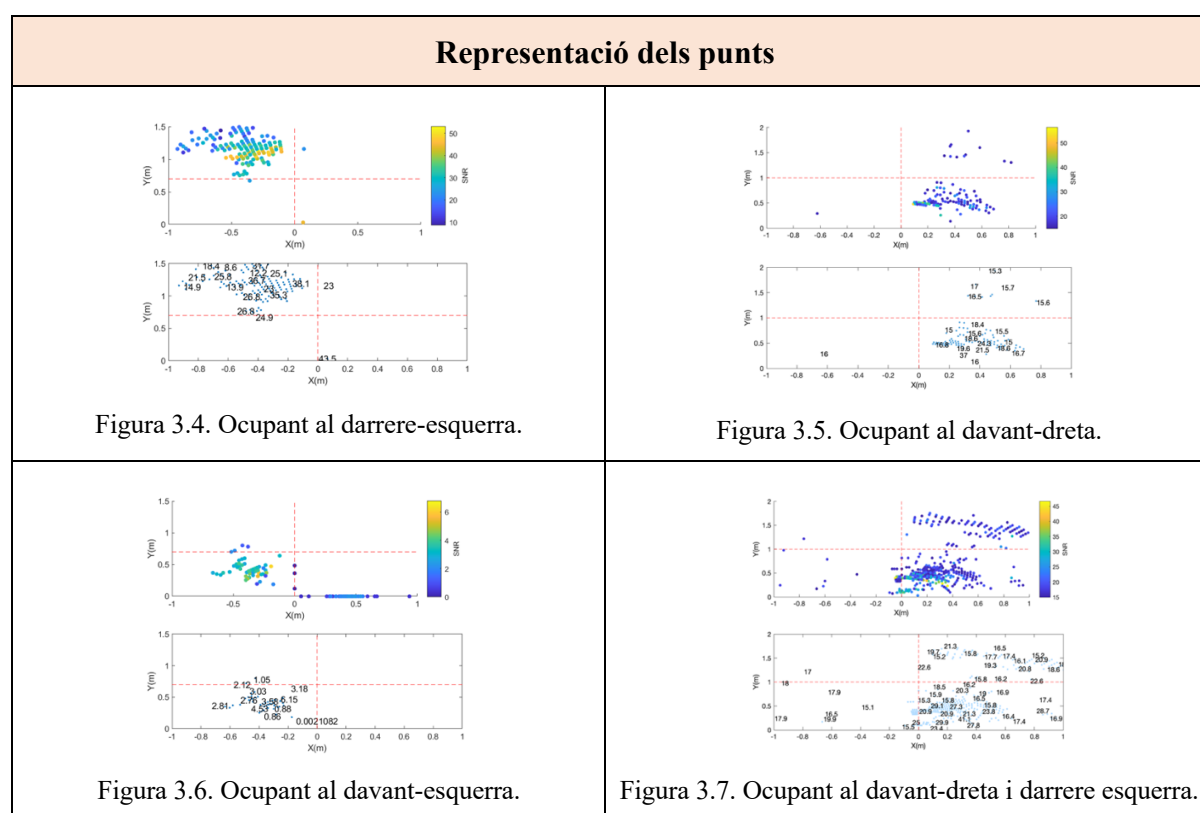
Aquests criteris s'han establert d'aquesta manera ja que la potència rebuda als seients de davant és major que al darrere i, a més, no hi ha cap obstacle entre mig que pugui dificultar l'arribada dels punts com en són els seients a les zones del darrere.

3.3.3. Presentació i anàlisi dels resultats obtinguts.

A continuació es presentaran els resultats obtinguts.

En primer lloc, es va realitzar un programa per representar els punts capturats, tal com es pot veure a la Taula 3.4. S'hi representen alguns dels casos capturats. El programa utilitzat s'anomena *processData.m*, tal com s'ha explicat en el Capítol II.

Les proves van realitzar-se en quatre cadires disposades com un cotxe de 4 places per poder veure el funcionament de l'aplicació en un entorn fora de rebots i poder fer les modificacions pertinents abans de portar-ho a l'entorn real. Es van delimitar les 4 zones amb una línia discontinua de color vermell.



Taula 3.4. Plots X-Y per diversos casos (amb l'SNR en format de color i numèric).

Cal notar que, per exemple, en la Figura 3.5, alguns punts han sortit al darrere-dreta, sense haver-hi cap ocupant en aquella posició. Són possibles reflexions amb l'entorn.

Per elaborar l'aplicació de detecció i ocupació d'usuaris a l'interior d'un vehicle, s'han creat diversos programes:

- *recognition.m*.
- *dadesFuncioTemps.m* i *dadesFuncioTempsReal.m*.
- *createConfusionMatrix.m*.

A continuació, es detallarà el funcionament de cadascun d'ells.

3.3.3.1. Programa *recognition.m*.

En primer lloc, hi ha el programa anomenat *recognition.m*. Aquest programa llegeix les dades capturades des d'un fitxer de text i representa la resposta cada cert nombre de mostres en una matriu de colors: en vermell, un seient ocupat ('1') i, en verd, un seient lliure ('0'). Està fet de manera que s'ha de prémer el botó 'Intro' del teclat per calcular la següent resposta dins el següent bloc de mostres preses.

En els exemples que s'exposen en la Taula, les captures són de 15 segons i el càlcul de les solucions és cada 100 mostres. S'hi mostren, per una banda, una figura amb dos *subplots*: ambdós mostren les dades X-Y; el primer, marcant amb una escala de colors el valor de l'SNR de cada punt i, el segon, amb format numèric. Per altra banda, també es mostra l'ocupació en una imatge tal com s'explicarà a continuació. S'han realitzat diverses captures per alguns casos per vehicles de quatre i cinc places.

La resposta queda representada de la següent manera per quatre ocupants (Figura 3.8):

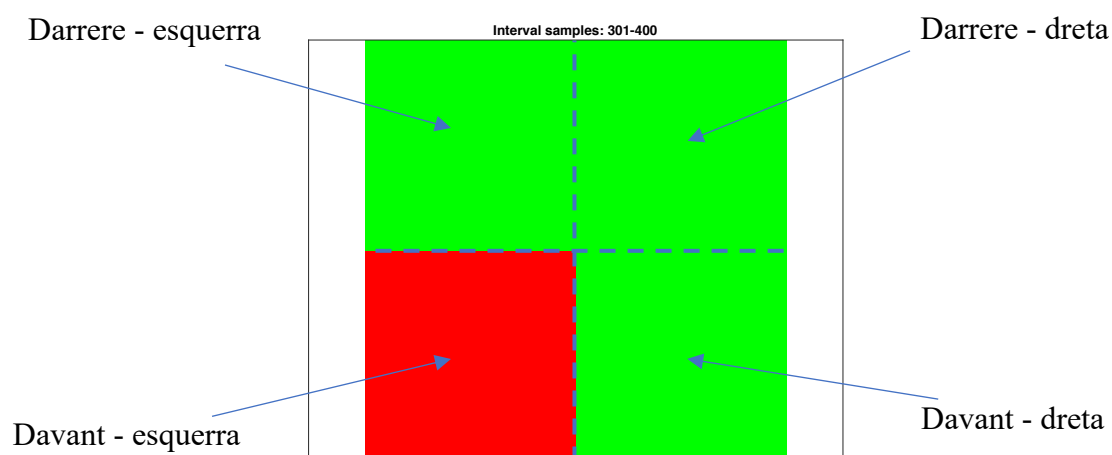


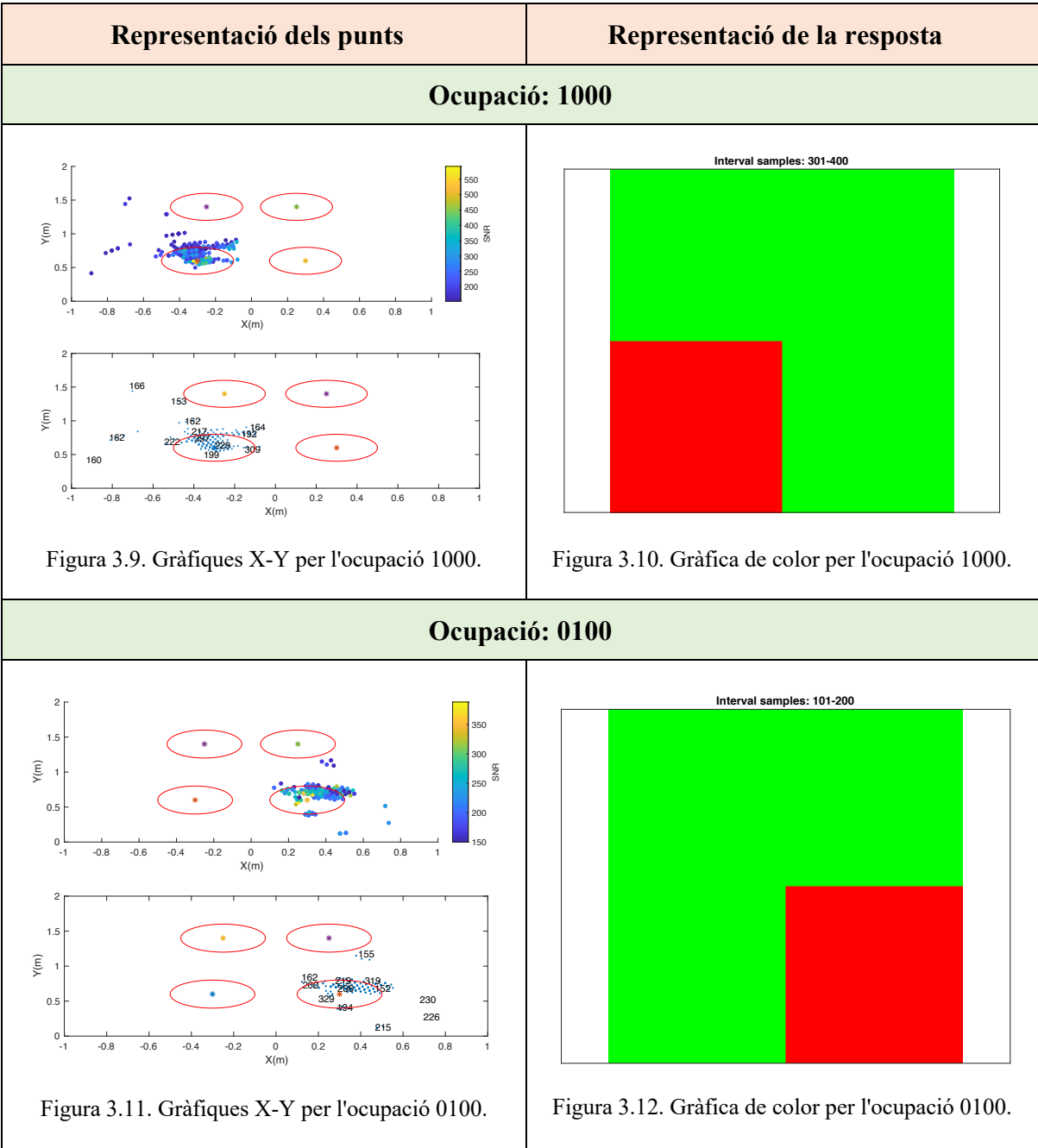
Figura 3.8. Resposta pel cas d'un vehicle de quatre ocupants (verd - lliure, vermell - ocupat).

Per tal de desar les dades en fitxer de text i representar-les posteriorment, s'ha guardat amb el nom de 'dades' seguit d'un nombre binari de 4 dígit, seguint el criteri de la Taula 3.5.

X	X	X	X
Davant Esquerra	Davant Dreta	Darrere Esquerra	Darrere Dreta

Taula 3.5. Estructura binària sobre la representació de l'ocupació per un vehicle de 4 places.

El format de la matriu segueix l'ordre binari, és a dir, va del '0000' al '1111' (del '0' al '15'), de tots els seients buits a tots els seients ocupats. En la Taula 3.6 s'hi representen algunes de les respostes (punts i solució) per a un, dos, tres i quatre ocupants pel cas d'un vehicle de quatre places.



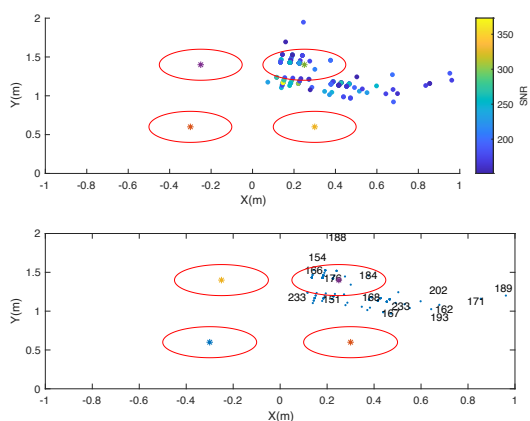
Ocupació: 0001

Figura 3.13. Gràfiques X-Y per l'ocupació 0001.

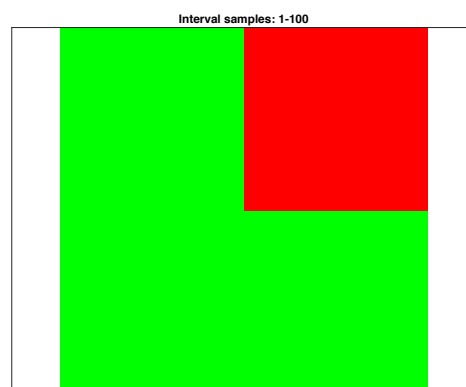


Figura 3.14. Gràfica de color per l'ocupació 0001.

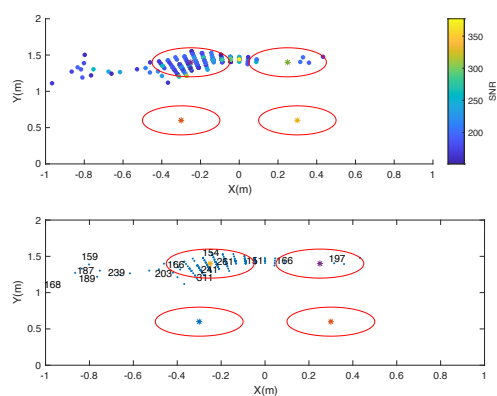
Ocupació: 0010

Figura 3.15. Gràfiques X-Y per l'ocupació 0010.

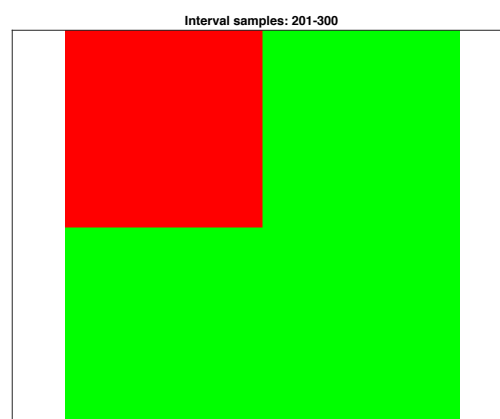


Figura 3.16. Gràfica de color per l'ocupació 0010.

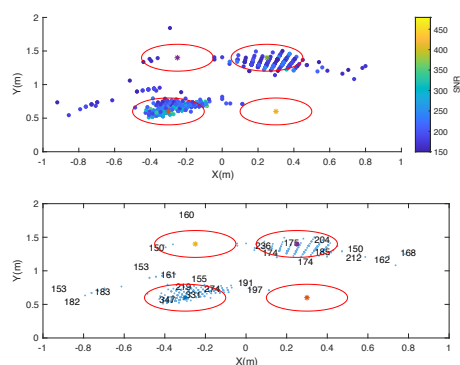
Ocupació: 1001

Figura 3.17. Gràfiques X-Y per l'ocupació 1001.

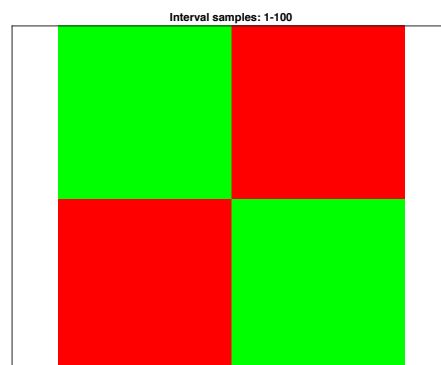


Figura 3.18. Gràfica de color per l'ocupació 1001.

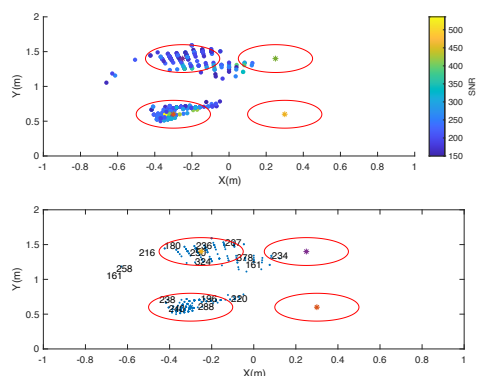
Ocupació: 1010

Figura 3.19. Gràfiques X-Y per l'ocupació 1010.

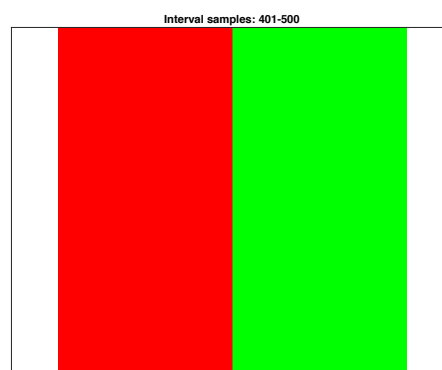


Figura 3.20. Gràfica de color per l'ocupació 1010.

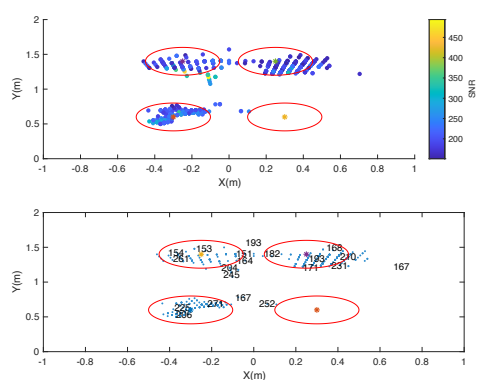
Ocupació: 1011

Figura 3.21. Gràfiques X-Y per l'ocupació 1011.

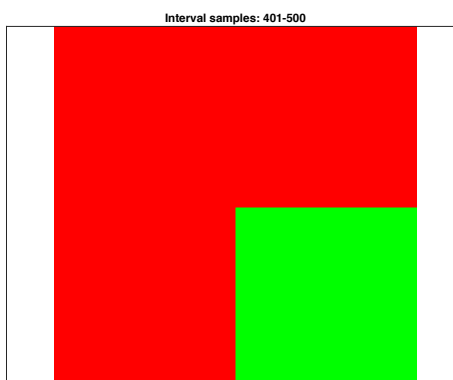


Figura 3.22. Gràfica de color per l'ocupació 1011.

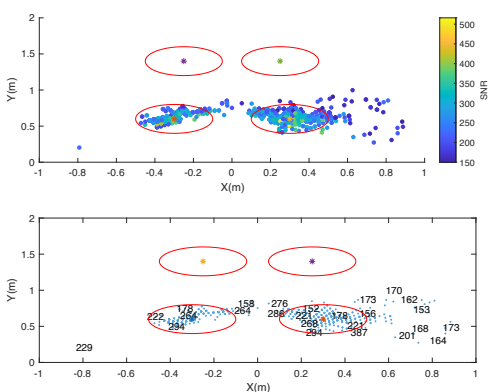
Ocupació: 1100

Figura 3.23. Gràfiques X-Y per l'ocupació 1100.

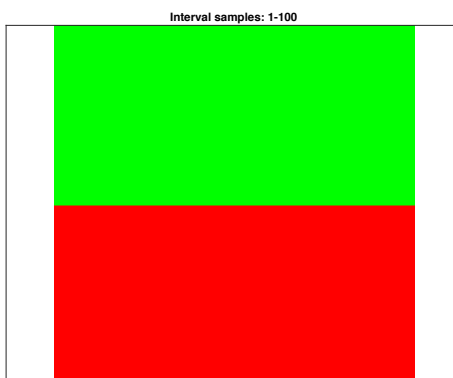
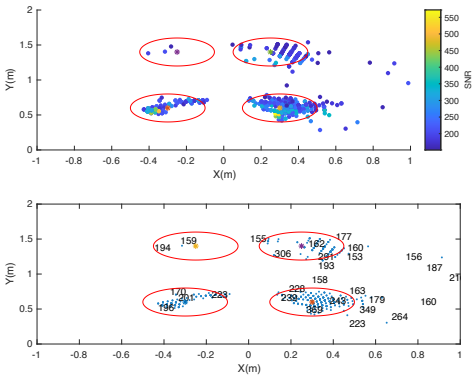
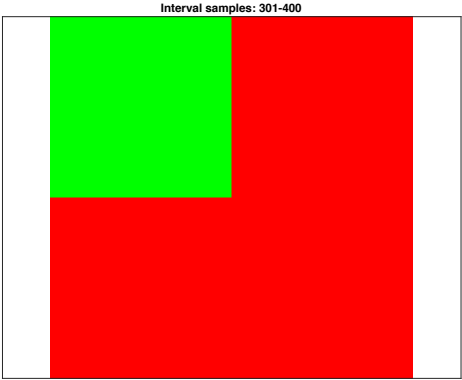
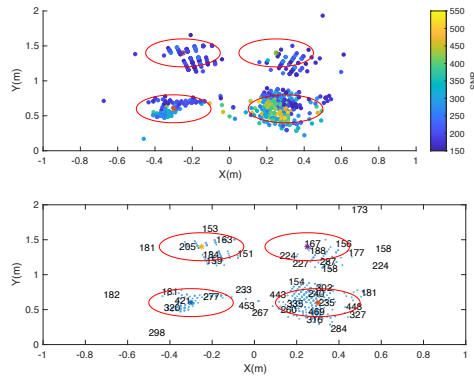
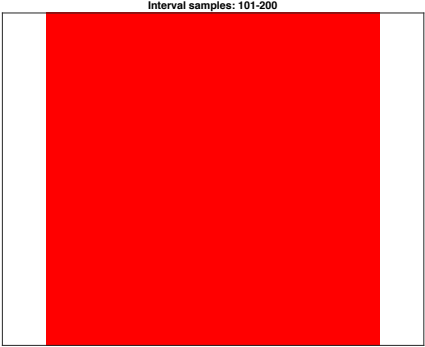


Figura 3.24. Gràfica de color per l'ocupació 1100.

Ocupació: 1101	
	
Figura 3.25. Gràfiques X-Y per l'ocupació 1101.	Figura 3.26. Gràfica de color per l'ocupació 1101.
Ocupació: 1111	
	
Figura 3.27. Gràfiques X-Y per l'ocupació 1111.	Figura 3.27. Gràfica de color per l'ocupació 1111.

Taula 3.6. Respostes obtingudes en un vehicle de quatre places.

La resposta queda representada de la següent manera per cinc ocupants (Figura 3.28):

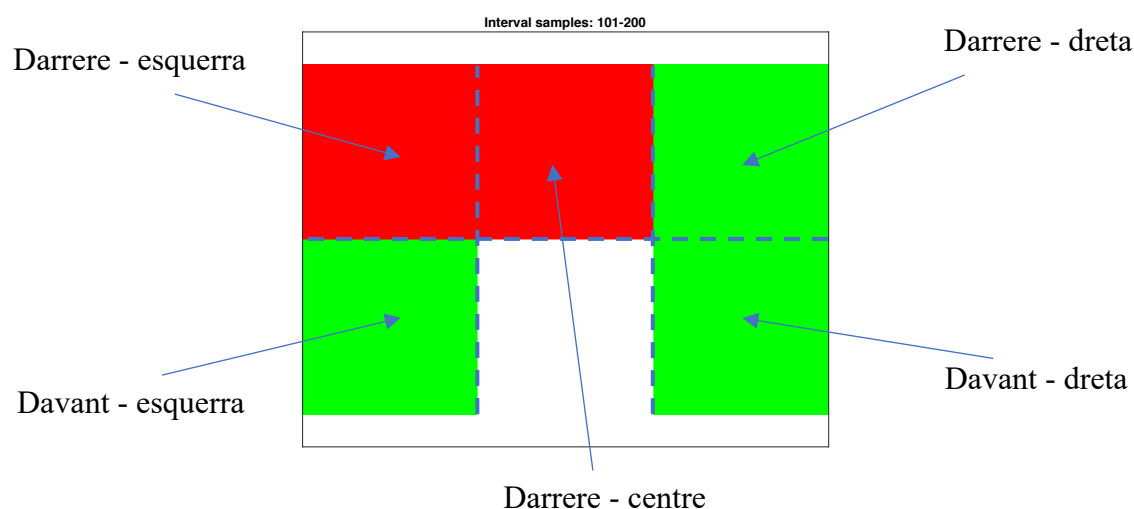


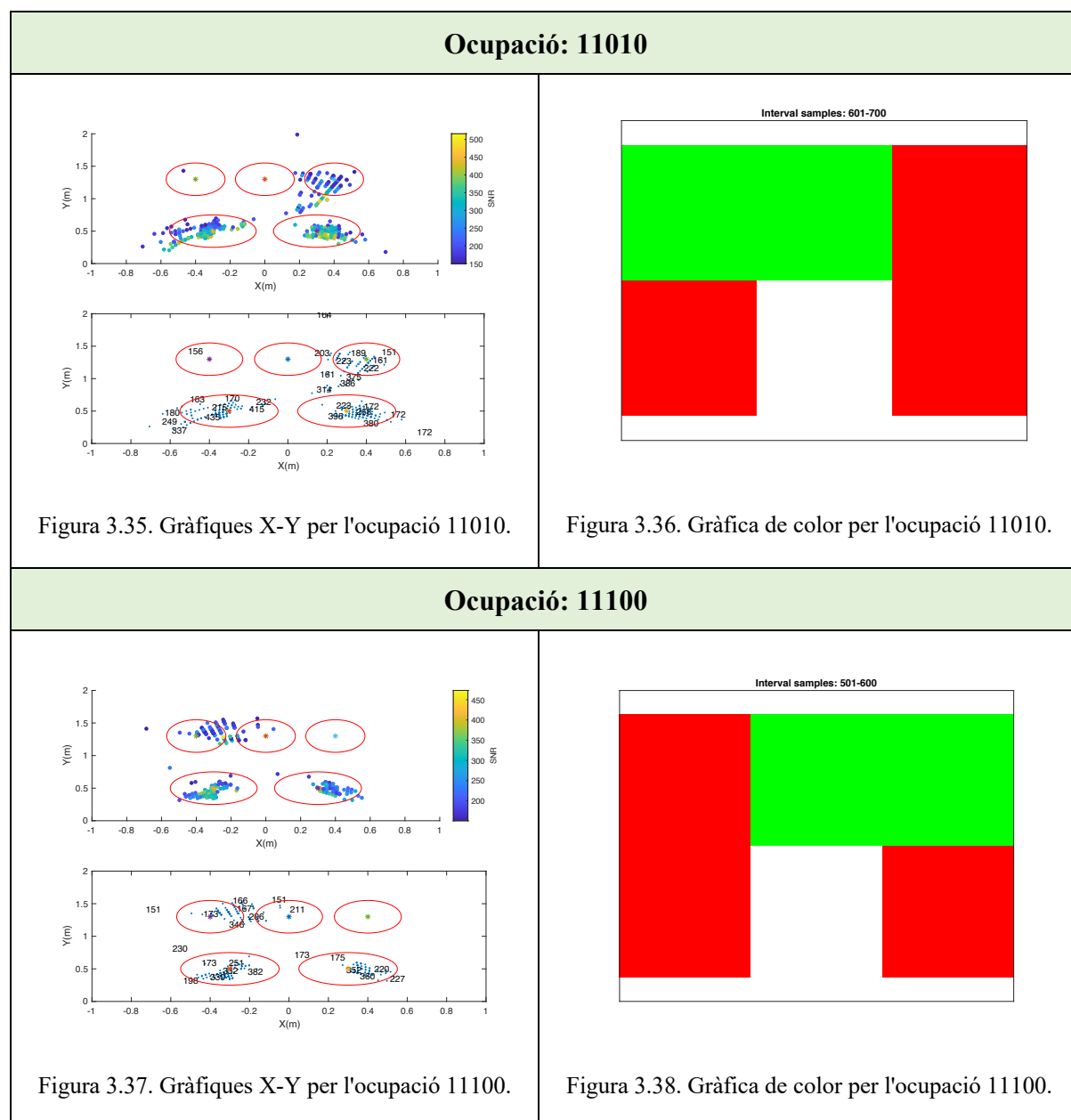
Figura 3.28. Resposta pel cas d'un vehicle de cinc ocupants (verd - lliure, vermell - ocupat).

Per tal de desar les dades en fitxer de text i representar-les posteriorment, s'ha guardat amb el nom de 'dades' seguit d'un nombre binari de 5 dígits, seguint el criteri de la Taula 3.7 pel vehicle de 5 places.

X	X	X	X	X
Davant Esquerra	Davant Dreta	Darrere Esquerra	Darrere Dreta	Darrere Centre

Taula 3.7. Estructura binària sobre la representació de l'ocupació per un vehicle de 5 places.

El format de la matriu segueix l'ordre binari, és a dir, que va del '00000' al '11111' (del '0' al '31'), de tots els seients buits a tots els seients ocupats. En la Taula 3.8 s'hi representen algunes de les respostes (punts i solució) per a un, dos i tres ocupants pel cas d'un vehicle de cinc places.



Taula 3.8. Respostes obtingudes en un vehicle de cinc places.

Per tal de representar els punts, s'ha utilitzat el programa *processData.m*. S'ha modificat per tal que, enlloc de representar les zones mitjançant línies discontinües, representi les àrees de detecció de cada ocupant del vehicle en la zona corresponent.

Com a observació, s'ha de deixar passar un cert nombre de mostres per tal que la resposta sigui correcta. Aquest nombre de mostres és el temps en el què l'usuari entra al cotxe i es situa al seient indicat ja que, mentrestant, genera moviment en altres posicions. Per aquest motiu, en la captura del reconeixement de la taula anterior sempre s'hi representa a partir de la mostra 500-600.

Com apunt final, el valor de l'SNR ha de ser dividit per 10 ja que en la representació realitzada en les figures anteriors no es va fer.

Els dos programes següents són per representar, per cada seient, l'ocupació en funció del temps.

3.3.3.2. Programes *dadesFuncioTemps.m* i *dadesFuncioTempsReal.m*.

El programa *dadesFuncioTemps.m* parteix d'un fitxer de text des d'on en llegeix les dades i ho representa, mentre que el programa *dadesFuncioTempsReal.m* va realitzant captures i representant-ne les respostes cada cert nombre de mostres, acumulant les anteriors, a temps real. A més, també representa la matriu de colors d'ocupació com en el programa *recognition.m*, tot i que en aquest apartat no ho tornarem a mostrar.

En la Figura 3.39 se'n pot veure un exemple pel cas d'un vehicle de 4 places.

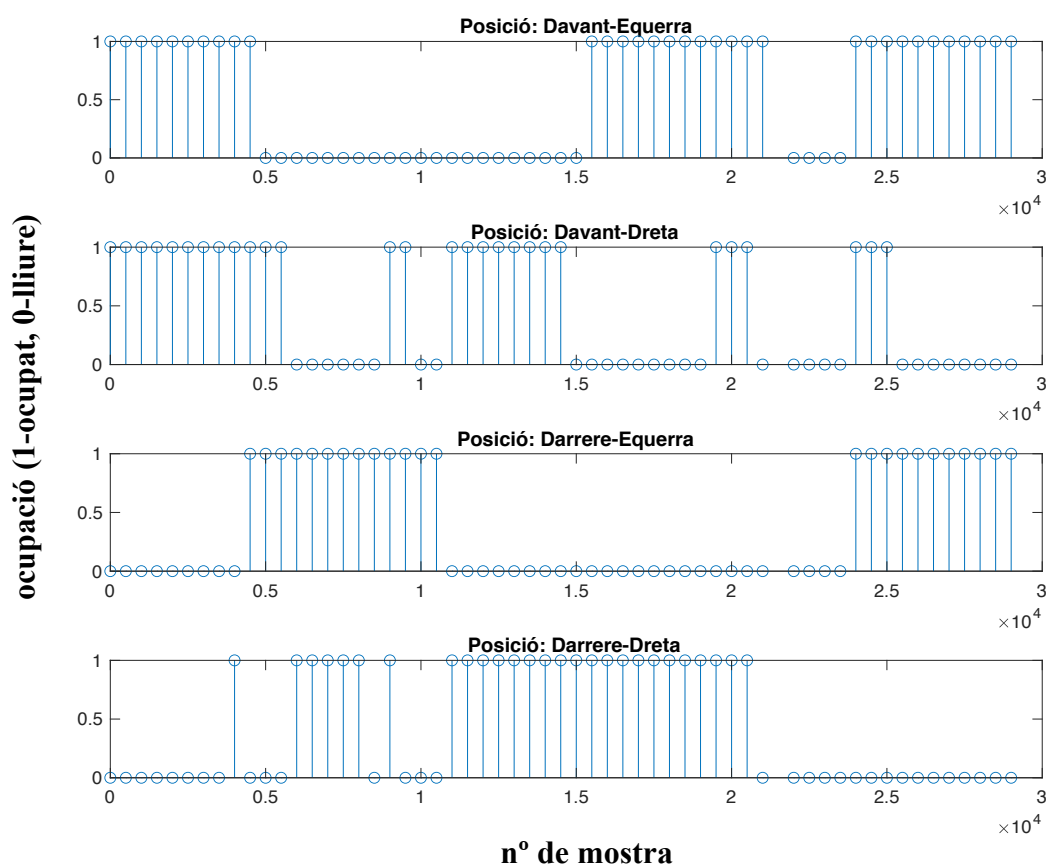


Figura 3.39. Representació de l'ocupació en funció del número de mostra.

En aquest exemple es pot anar veient la seqüència de l'ocupació d'un vehicle amb 29.000 mostres capturades. El processat es duu a terme amb dues i tres persones, anant alternant i, cada 1000 mostres. Per tant, hi haurà 29 solucions representades.

En el diagrama de blocs de la Figura 3.40 es pot veure la seqüència seguida. Aquest esquema es va elaborar a partir dels punts mostrats en l'eix X i Y i es va optar per realitzar-se cada 500 mostres, a diferència del resultat anterior. Realitzant proves, es va observar que era més exacte fer el processat cada 500 que en un altre nombre de mostres.

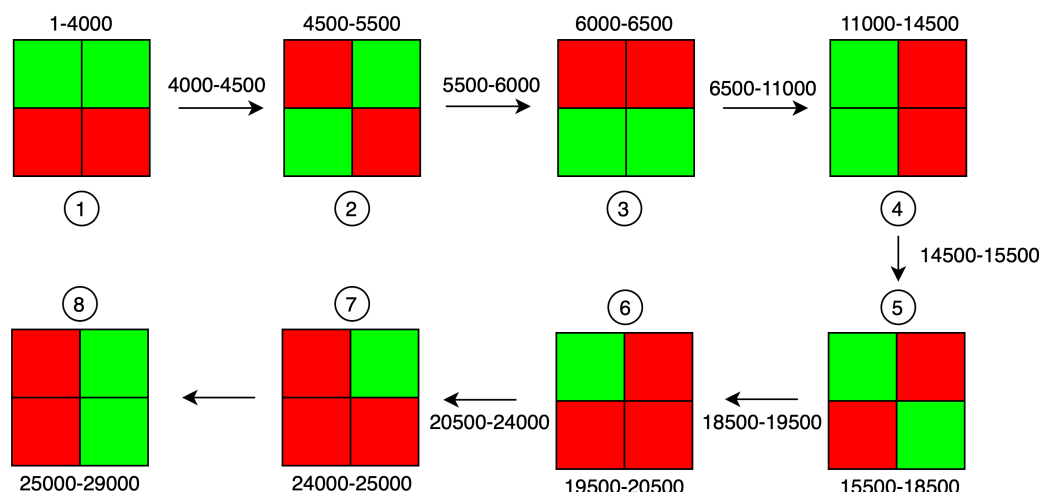


Figura 3.40. Esquema sobre l'evolució de l'ocupació en 29000 mostres i un pas de 500.

En vermell, es representen els seients ocupats, que corresponen als '1' de la Figura 3.39. En verd, es representen els seients lliures ('0'). Les transicions són representades en fletxes i també s'hi indica el nombre de mostres que ocupen, així com a l'ocupació de les 8 combinacions que s'han realitzat.

A la Figura 3.41 s'hi pot veure més detalladament la relació entre la Figura 3.39 i la Figura 3.40.

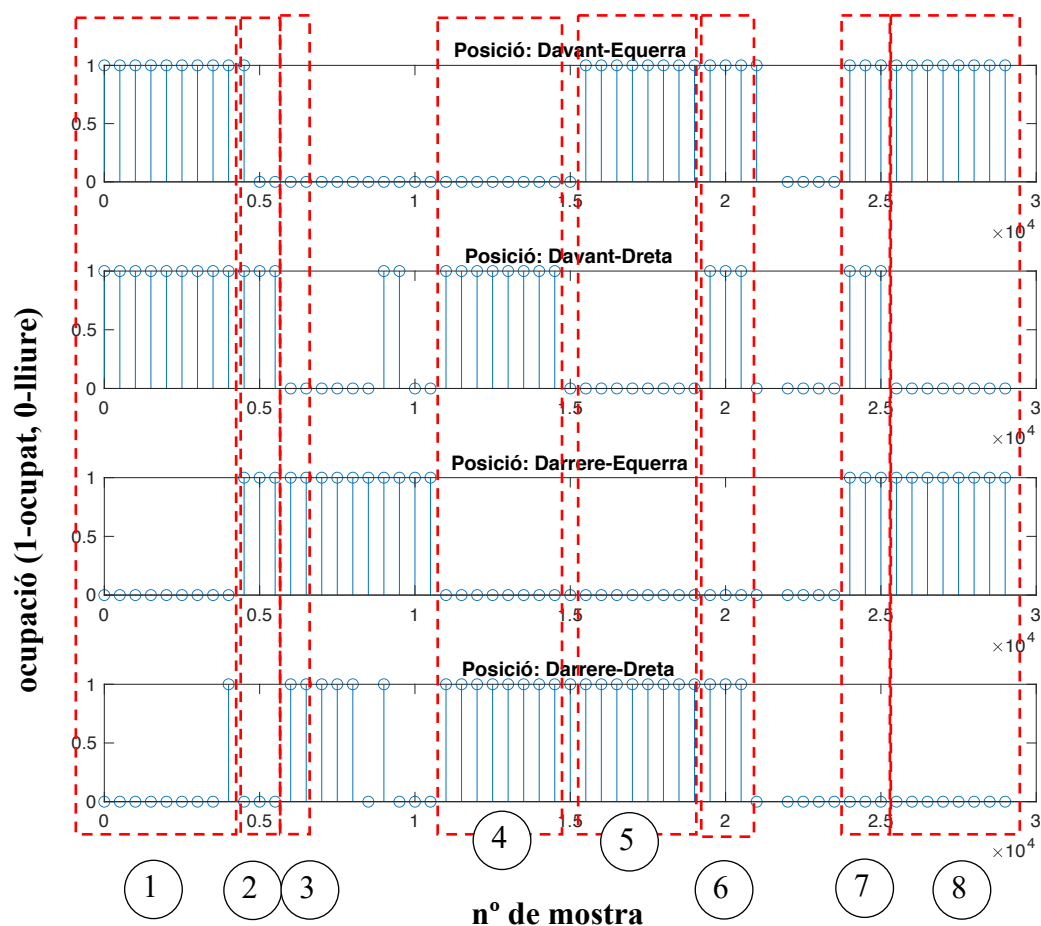


Figura 3.41. Representació de l'ocupació en funció del número de mostra, seguint la seqüència de la Figura 3.40.

Com es pot observar, hi tenim:

- *Seqüència 1*: Davant-Esquerra i Davant-Dreta.
- *Seqüència 2*: Davant-Dreta i Darrere-Dreta.
- *Seqüència 3*: Darrere-Esquerra i Darrere-Dreta.
- *Seqüència 4*: Davant-Dreta i Darrere-Dreta.
- *Seqüència 5*: Davant-Esquerra i Darrere Dreta.
- *Seqüència 6*: Davant-Esquerra, Davant-Dreta i Darrere-Dreta.
- *Seqüència 7*: Davant-Esquerra, Davant-Dreta i Darrere-Esquerra.
- *Seqüència 8*: Davant-Esquerra i Darrere-Esquerra.

Es correspon amb la Figura 3.40. Les transicions queden representades fora dels requadres vermells. Com es pot observar, hi ha alguns pics que donen error i s'obté una solució que no es correspon amb la realitat. Un exemple en són aquells que donen '1' i estan envoltats de '0'. Com és lògic, si hi ha un ocupant, donarà un '1' en un seguit de mostres consecutives i no només en una, per la qual cosa s'ha de trobar una manera de descartar-la.

Una possible solució és aplicar un *filtre alpha* (Fórmula 3.4). Consisteix en, mitjançant la solució actual, l'anterior i un coeficient *alpha* entre 0 i 1, filtrar la resposta i, per tant, obtenir una resposta amb més sentit.

$$sortida_{filtre} = \alpha \cdot sortida_{actual} + (1 - \alpha) \cdot sortida_{anterior} \quad (3.4)$$

Com més petit i proper a 0 sigui *alpha*, més filtratge hi haurà i, per tant, més restrictiu serà el filtre. En la Taula 3.9 es poden observar diversos exemples de resposta per diversos valors d'*alpha*. En cada gràfica hi apareix el valor de la sortida del filtre i, a més, s'aproxima cada valor a '0' o '1' segons si és més gran o més petit que 0.5, per tenir només sortides binàries.

Aplicació del filtre α a la resposta

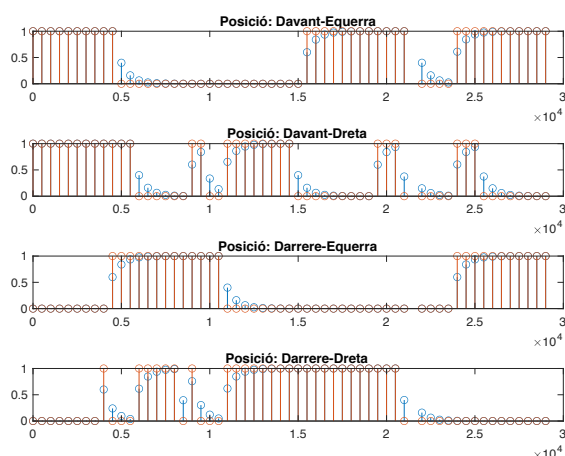


Figura 3.42. Aplicació del filtre amb $\alpha = 0.8$.

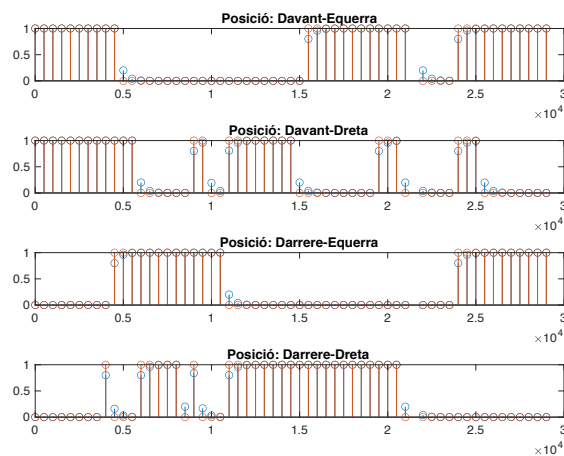


Figura 3.43. Aplicació del filtre amb $\alpha = 0.6$.

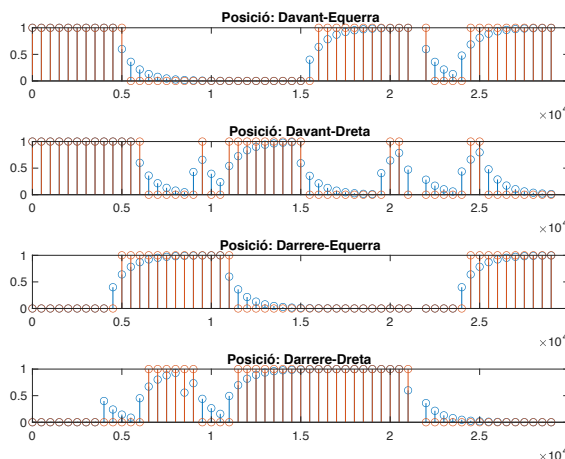


Figura 3.44. Aplicació del filtre amb $\alpha = 0.4$.

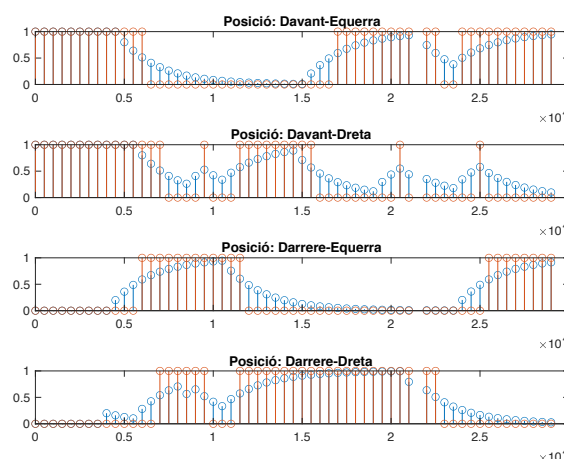


Figura 3.45. Aplicació del filtre amb $\alpha = 0.2$.

Taula 3.9. Resultats de l'aplicació del filtre α a la resposta.

Com es pot observar, el que s'ha fet és aplicar per quatre valors diferents el filtre α i, seguidament, comparar el valor amb 0.5. Si el valor és menor, pren valor de '0' mentre que, si és major, prendrà valor d'1'. Com més petit sigui el valor d' α , més arrodonida queda la resposta; cal fixar-se que amb els valors baixos d' α s'han eliminat alguns pics que no tenien cap significat.

Tot i així, mitjançant l'execució de diferents casos, s'ha optat per no aplicar el filtre α . Per cada aplicació, per tal que concordés el millor possible, s'havia de seleccionar un valor diferent i realitzar una aproximació diferent. S'ha considerat que amb la resposta prèvia a l'aplicació del filtre el resultat és força bo, tal com es veurà a continuació, i suficient.

3.3.3.3. Programa *createConfusionMatrix.m*.

Per acabar, amb l'últim programa (*createConfusionMatrix.m*), es pot comprovar la precisió de l'algorisme d'ocupació dissenyat de dues maneres, tot creant una matriu de confusió:

- En primer lloc, el nombre de persones que hi ha al cotxe. Pel cas del vehicle de 4 places, serà una matriu 4×4 mentre que pel cas de 5 places serà una matriu 5×5.
- En segon lloc, la detecció de l'ocupació en un seient en concret. Pel cas del vehicle de 4 places, com que hi ha 16 casos possibles, serà una matriu 16×16, mentre que pel cas del vehicle de 5 places serà una matriu 32×32 ja que hi ha 32 casos possibles.

En ambdós casos la matriu representa el resultat esperat enfront l'obtingut.

Les següents dues matrius (Figura 3.46, 3.47) representen la detecció del nombre d'ocupants per a vehicles de 4 i 5 persones, respectivament. Com es pot veure, la precisió ('*accuracy*', és a dir, el percentatge d'encert de l'algorisme calculat com a *Casos Encertats/Casos Totals*) és del 95% i 88%, respectivament, amb el que es pot concloure que l'algorisme és molt bo a l'hora de detectar el nombre d'ocupants en un vehicle.

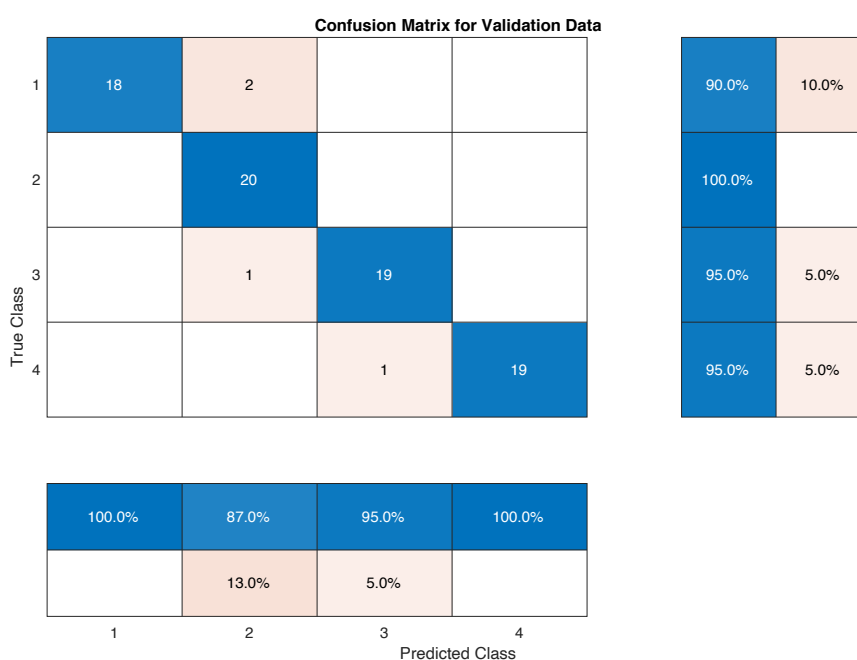


Figura 3.46. Matriu del nombre d'ocupació per vehicles de 4 places.

La precisió (*accuracy*) és del 95 %. En aquest cas, l'algorisme ha confós dues vegades la detecció d'una persona (ho ha classificat com a 2 ocupants quan realment n'era 1 a la posició [1,2]). També ha classificat 2 ocupants enlloc dels 3 que eren [3,2], i 3 ocupants enlloc dels 4 [4,3].

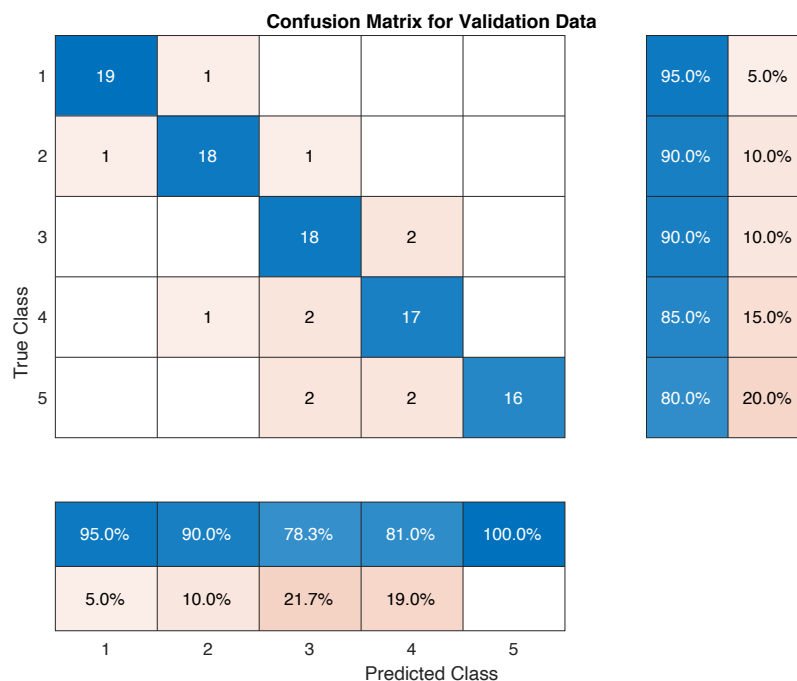


Figura 3.47. Matriu del nombre d'ocupació per vehicles de 5 places.

La precisió (*accuracy*) és del 88%. Pel cas del vehicle de 5 places ja hi ha més discussió. Sense entrar en detall, es pot observar que el fet d'haver-hi un seient el mig pot crear confusió en certes ocasions i fer que es detectin més ocupants dels que realment hi ha. Això és degut a la proximitat entre les tres zones del darrere. Tot i així, com ja s'ha dit, la precisió que ha donat l'algorisme és molt correcte.

Pel cas de la detecció de l'ocupació per un vehicle de quatre places, és a dir, de la posició dels ocupants en un vehicle, tenim 16 casos possibles (2 posicions, ocupat i no ocupat, i 5 seients, creant 2^4 combinacions possibles). Per aquest motiu, s'ha optat per fer 10 captures per cada cas, ja que amb aquest nombre se'n pot obtenir una aproximació de la precisió de l'algoritme força correcta. Es poden veure els resultats a la Figura 3.48.

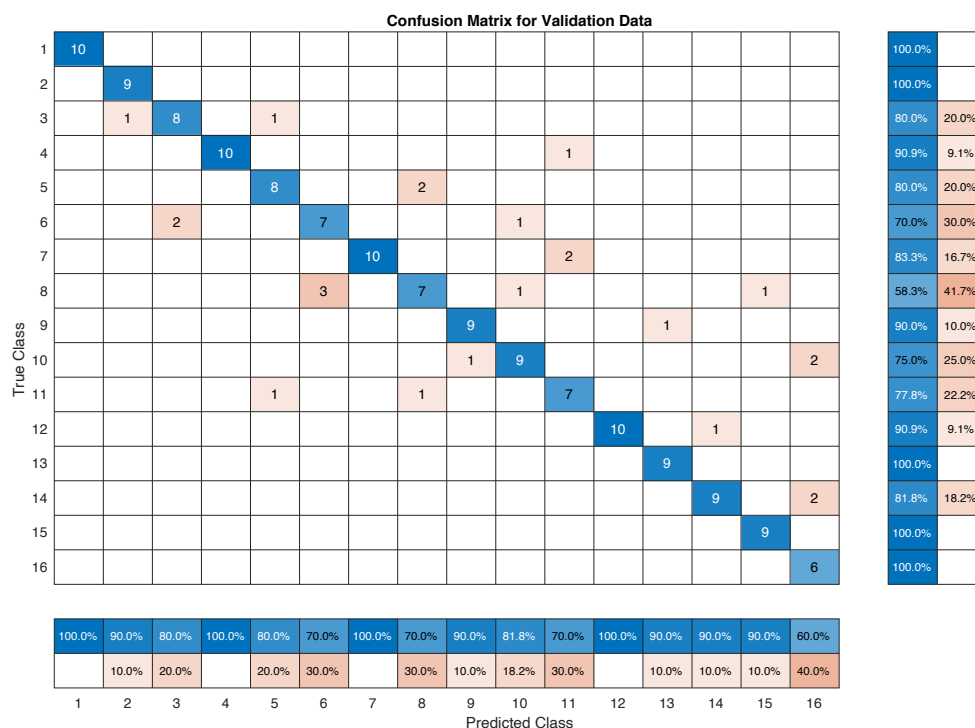


Figura 3.48. Matriu d'ocupació en cada seient per vehicles de 4 places.

La precisió és del 85.6 %. Com pot observar-se, l'algorisme té una precisió força correcta. S'ha optat presentar els resultats només pel vehicle de 4 places ja que, pel cas del vehicle de 5 places, el nombre de casos pujaria a 32 i la precisió seria semblant. Per tant, el resultat donat per la matriu 16×16 és el resultat final d'aquest punt.

Els resultats d'aquest punt s'han presentat només mostrant la matriu de confusió. Tot i així, el programa *createConfusionMatrix.m* mostra d'altres gràfiques (Figura 3.49): X-Y dels punts que es van capturant, *Range-Doppler*, la resposta representada en la matriu verd-vermell, i la disposició dels punts en cada detecció. També s'hi pot veure el número de detecció que s'està produint, el cas que s'està analitzant i el cas que s'està detectant, a partir dels quals es construirà la matriu de confusió.

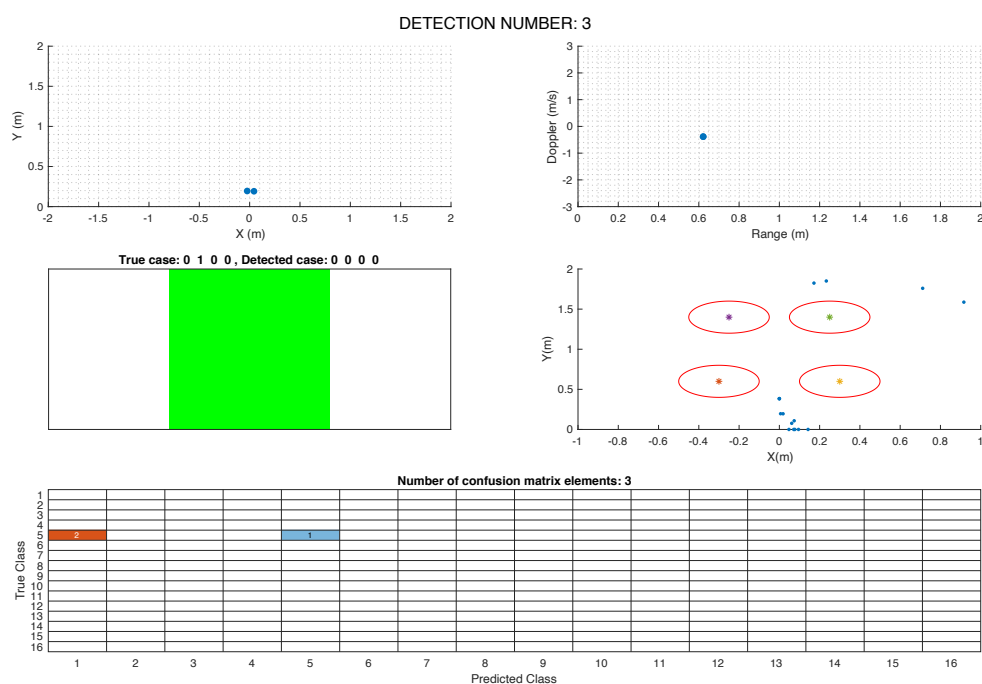


Figura 3.49. Gràfiques i matriu mostrades pel programa *createConfusionMatrix.m*.

En el diagrama de flux de la Figura 3.50 es pot observar el seguiment de l'execució completa dels programes que s'han comentat en aquest punt. Mostra detalladament els programes i funcions utilitzats per l'aplicació de detecció i ocupació de persones en un vehicle.

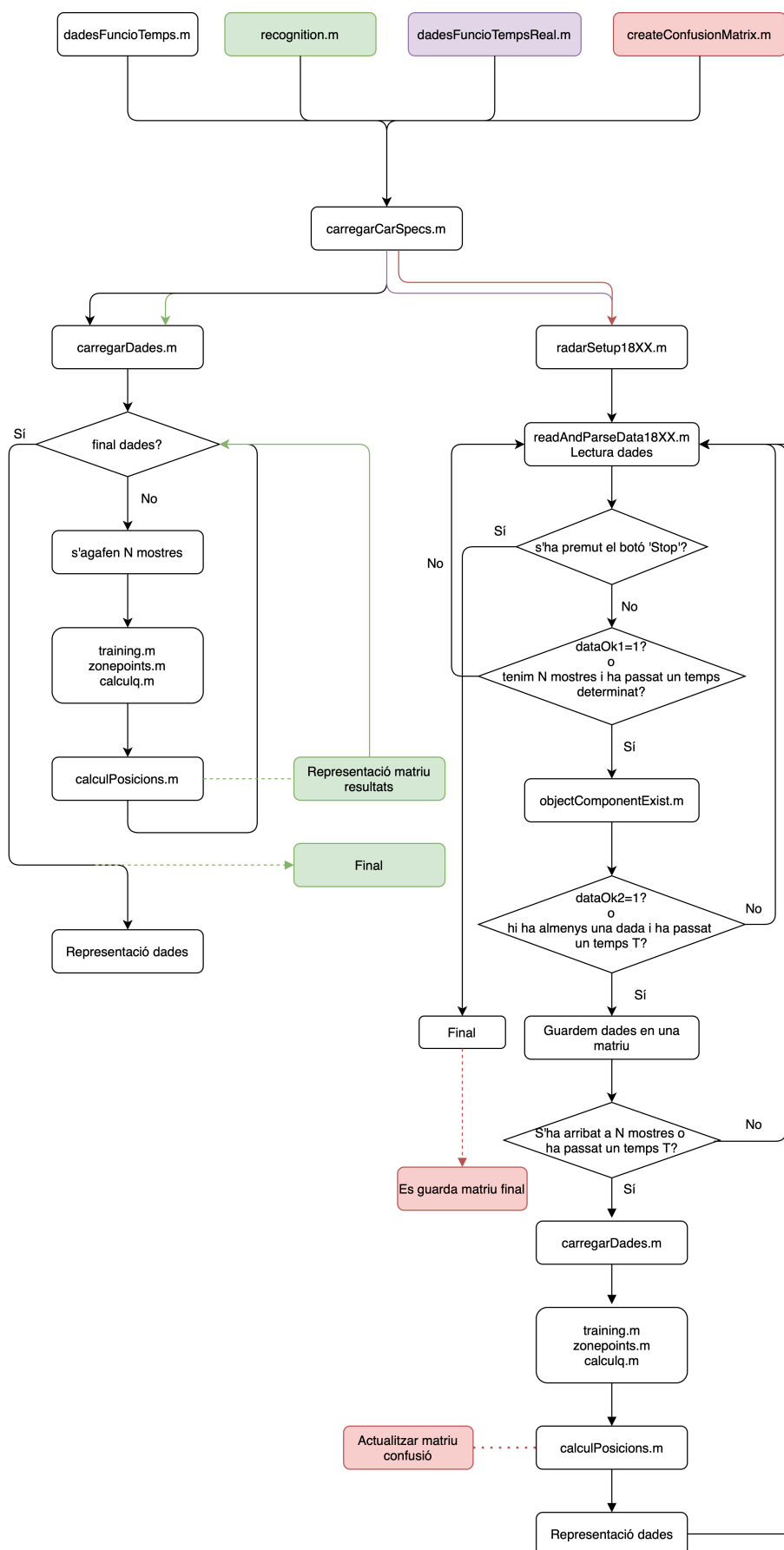


Figura 3.50. Diagrama de flux dels programes d'ocupació en vehicles detallats en el Capítol III.

3.4. App interfície gràfica.

Per tal que l'usuari pugui configurar alguns paràmetres bàsics sense haver de modificar el codi manualment, s'ha creat una APP de *Matlab* amb la funció que el procés sigui més automàtic i senzill. Des d'aquesta podem configurar els paràmetres de la Taula 3.10.

Dispositiu PC	El dispositiu que s'utilitza, per tal de configurar els ports en funció d'aquest.
Programa	El programa a executar. Es pot seleccionar entre <i>dadesFuncioTempsReal.m</i> i <i>createConfusionMatrix.m</i> .
Temps entre mesures (s)	El temps en el qual es van prenent mostres per posteriorment processar-les i donar la resposta. Quan acaba aquest temps, es reinicien els vectors de dades.
Nombre de places	Selecció si és un vehicle de 4 o 5 places.
Distància X seient davant (m)	Correspon a la distància que hi ha del centre del RADAR al seient dret o esquerra davanters en l'eix X.
Distància Y seient davant (m)	Correspon a la distància que hi ha del centre del RADAR al seient dret o esquerra davanters en l'eix Y.
Distància X seient darrere (m)	Correspon a la distància que hi ha del centre del RADAR al seient dret o esquerra posteriors en l'eix X.
Distància Y seient darrere (m)	Correspon a la distància que hi ha del centre del RADAR al seient dret o esquerra posteriors en l'eix Y.
Radi Davant (m)	Correspon al radi de la zona de davant.
Radi Darrere (m)	Correspon al radi de la zona de darrere.

Taula 3.10. Paràmetres del vehicle a modificar amb l'APP creada mitjançant *Matlab*.

La gràfica que mostra l'aplicació està feta de tal manera que, si es modifiquen les dades, es va actualitzant automàticament i al moment, de manera que l'usuari pot veure a temps real la distribució de les zones. A més, mostra els paràmetres seleccionats per defecte tant pel vehicle de 4 places com pel de 5 utilitzats en el projecte (els detallats en la Taula 3.3). La Figura 3.51 mostra el cas d'un vehicle de 4 places i, la Figura 3.52, la d'un de 5 places.

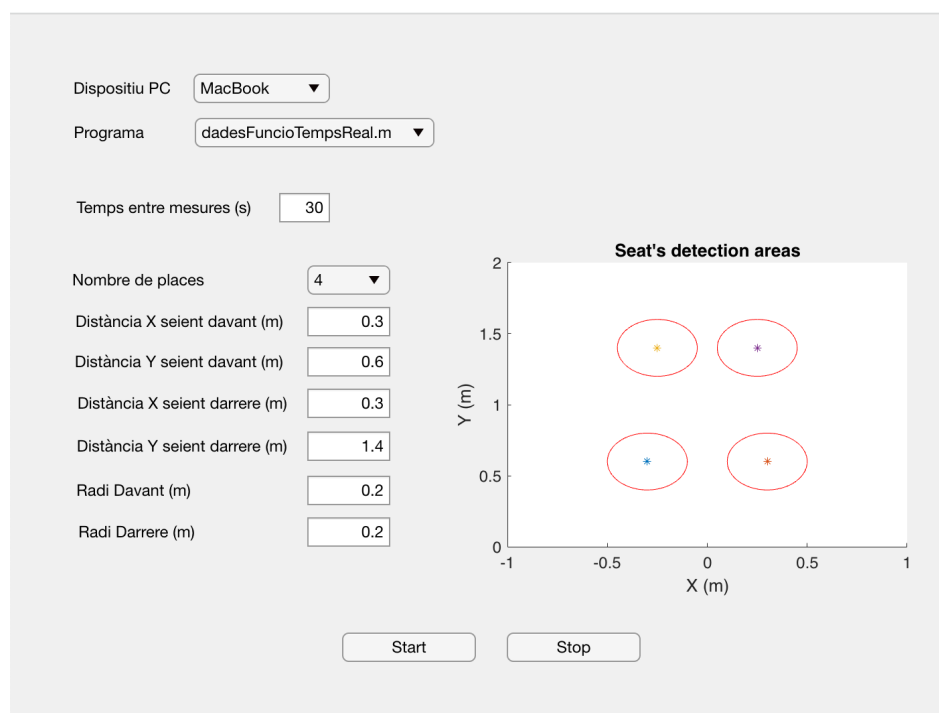


Figura 3.51. Aparença visual de l'APP si es configura per un vehicle de 4 places.



Figura 3.52. Aparença visual de l'APP si es configura per un vehicle de 5 places.

3.5. Processat amb *RaspberryPi* i *Python*.

L'objectiu d'aquesta part del treball és millorar la portabilitat de l'aplicació i poder-la adaptar a les condicions d'espai d'un vehicle. Fins el moment, s'ha treballat amb *Matlab* per tal de processar les dades i extreure'n una resposta. Aquesta aplicació requereix d'un ordinador portàtil per tal de funcionar. En condicions real, és inviable utilitzar-ne un. Per tant, és necessari buscar un ordinador capaç de processar les dades i que ocupi poc espai.

La solució a aquest problema és la *Raspberry Pi*, un ordinador de placa única i de dimensions força reduïdes. Aquest ordinador no pot treballar amb *Matlab*, però sí amb *Python*, un llenguatge força senzill d'utilitzar i semblant a *Matlab*.

La traducció de codi de *Matlab* a *Python* és força senzilla ja que el codi és força semblant. És necessari instal·lar l'eina *pip* i *Python*, si la *Raspberry* no ho incorpora, i les següents llibreries:

- *serial*
- *time*
- *math*
- *numpy*
- *pyqtgraph*, en cas que es vulguin representar les dades. En el cas d'aquesta aplicació, tot i així, no és d'interès ja que només es vol tenir la resposta d'ocupació.

La lectura de les dades del RADAR amb *Python* ja estava creada per un altre usuari [6]. Per tant, en aquest projecte es van afegir les següents funcions, idèntiques a les de *Matlab* però amb codi *Python*, per tal de crear l'aplicació de detecció de l'ocupació en un vehicle de quatre i cinc places:

- *defZones()*. Definició de les zones, com en la funció *training.m*.
- *zonePoints()*. Es compten i s'agrupen els punts per zones.
- *calculQ()*. Càlcul de la solució a partir dels diversos criteris esmentats en punts anteriors.

Tal com es pot veure a la Figura 3.53, la solució es mostra pel terminal en '0' (no ocupació) i '1' (ocupació).

```

1 import serial
2
3 import time
4
5 import math as m
6
7 import numpy as np
8
9
10 # Change the configuration file name
11
12 configFile = 'xw16xx_profile_2021_03_24T11_58_43_671.cfg'
13
14
15 CLIPort = {}
16
17
18 sensorStart
19 [0. 1. 0. 0.]
20 [0. 1. 0. 0.]
21 [0. 1. 0. 0.]
22 [1. 1. 0. 0.]
23 [1. 1. 0. 0.]
24
25 Python 3.7.3 (/usr/bin/python3)
26 >>>
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figura 3.53. Resposta mostrada pel terminal de la Raspberry Pi.

3.6. Referències.

- [1] B. George, H. Zangl, T. Bretterkieber and G. Brasseur, "A Combined Inductive–Capacitive Proximity," IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, vol. 59, no. 5, pp. 1463-1470, MAY 2010.
- [2] S. LIM, J. JUNG, S.-C. KIM and S. LEE, "Deep Neural Network-Based In-Vehicle People Localization Using Ultra-Wideband RADAR," IEEE Access, vol. 8, pp. 96606-96612, May 2020.
- [3] B. Schleicher, I. Nasr, A. Trasser and H. Schumacher, "IR-UWB RADAR Demonstrator for Ultra-Fine Movement Detection and Vital-Sign Monitoring," IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, vol. 61, no. 5, pp. 2076-2085, MAY 2013.
- [4] M. Hoffmann, D. Tatarinov, J. Landwehr and A. R. Diewald, "A four-channel RADAR System for Rear Seat Occupancy Detection in the 24 GHz ISM band," GeMiC, pp. 95-98, 12-14 March 2018.
- [5] Texas Instruments. Vehicle Occupant Detection Reference Design. Ref: TIDUE95A. Data de la Design Guide: abril de 2018 (revisat l'abril de 2020).
Enllaç web: <https://ja.cat/vehicleTI> (consulta: 13 de maig de 2021).
- [6] Python program to read and plot the data in real time from the AWR1642 and IWR1642 mmWave radar boards (Texas Instruments). Github.
Enllaç web: <https://github.com/ibaiGorordo/AWR1642-Read-Data-Python-MMWAVE-SDK-2> (consulta: 4 de juny de 2021).

CAPÍTOL IV

SISTEMA DE DETECCIÓ DE GESTOS AMB LA MÀ BASAT EN RADAR FMCW EN LA BANDA MIL·LIMÈTRICA

En aquest capítol s'explicarà, en primer lloc, l'estat de l'art de l'aplicació sobre el sistema de detecció de gestos amb la mà basat en RADAR FMCW en la banda mil·limètrica (tecnologies, mètodes i mercat actual). Seguidament, es detallarà l'aplicació realitzada amb *Matlab*, tot explicant els passos seguits des de la captura de mostres fins el seu processament i reconeixement per determinar-ne el gest (conversió a imatge, aplicació de la DNN). Finalment, es presentaran els resultats obtinguts i es mostrarà la precisió de l'algorisme realitzat.

4.1. Estat de l'art.

L'aplicació de detecció de gestos també és un altre camp que està creixent exponencialment avui en dia i que té múltiples aplicacions en diversos sectors. Un d'ells n'és a l'interior del vehicle. El fet de poder detectar diversos gestos que realitza el conductor o copilot permet realitzar múltiples accions diferents sobre el mateix vehicle: pujar i baixar finestres, obrir i tancar les llums, aire condicionat, etc. sense haver-hi cap tipus de distracció pel conductor.

Existeixen diversos projectes que ja han tractat aquest tòpic i desenvolupat la seva aplicació, tal com es detallarà a continuació.

En la referència [1] es detalla el desenvolupament d'una tècnica de reconeixement de gestos mitjançant un sensor RADAR d'ones mil·limètriques de 60 GHz pel control de l'infoentreteniment en l'automòbil. El reconeixement de gestos s'està convertint en la forma més prominent d'interacció entre el ser humà i l'ordinador i pot utilitzar-se en la indústria de l'automòbil per proporcionar una interfície de control segura i intuïtiva que limiti la distracció del conductor. S'utilitza el RADAR per detectar característiques de moviment fi. Les característiques específiques dels gestos s'extreuen i s'utilitzen per construir un motor d'aprenentatge automàtic que pot realitzar el reconeixement de gestos en temps real.

En la referència [2] es presenta un sistema basat en un RADAR d'ones mil·limètriques polsades (a 60 GHz) per la detecció i classificació de gestos de la mà. El RADAR funciona a 144 Hz i s'utilitza per registrar 2160 imatges (amplitud reflectida en funció del temps) de 12 gestos genèrics de la mà. El reconeixement dels gestos s'aconsegueix mitjançant l'aprenentatge automàtic, utilitzant l'aprenentatge de transferència d'una xarxa neuronal convolucional preentrenada. S'obtenen excel·lents resultats de classificació amb una precisió de validació del 99,5%, basada en una divisió del 60% d'entrenament enfront el 40% de validació. També es presenta la matriu de confusió corresponent, que mostra un alt nivell d'ortogonalitat de classificació entre els gestos provats. Les dades del RADAR s'utilitzen pel reconeixement de gestos mitjançant l'aprenentatge automàtic. Es demostra que la representació de l'envoltant rang-temps de les dades d'alta velocitat de fotogrames d'un RADAR polsat és adequada pel reconeixement dels gestos de la mà.

En la següent referència, [3], es proposa un nou mètode per la detecció i el reconeixement del gest de la mà en continu, basat en un RADAR d'ona contínua modulada en freqüència (FMCW). En primer lloc, s'agafa la 2D-FFT per estimar els paràmetres de rang i Doppler de les dades brutes del gest de la mà i es construeix el mapa Rang-Temps (RTM) i el mapa Doppler-Temps (DTM). Per una altra banda, apliquem l'algoritme de classificació de senyals múltiples (MUSIC) per calcular l'angle i construir el mapa angle-temps (ATM). En segon lloc,

es proposa un mètode de detecció de gestos de la mà per segmentar els gestos continus de la mà utilitzant un llindar de decisió. En tercer lloc, la trajectòria central de temps-freqüència de cada espectrograma dels gestos de la mà s'agrupa mitjançant l'algoritme *k-means* i, a continuació, es presenta l'algoritme FDTW per reconèixer els gestos de la mà. Finalment, els experiments mostren que la precisió del mètode de detecció de gestos de la mà proposat pot arribar al 96,17%. La precisió mitja de reconeixement dels gestos de la mà de l'algoritme FDTW proposat és del 95,83%, mentre que la seva complexitat temporal es redueix a més d'un 50%.

A la referència [4] tracta sobre el reconeixement de gestos de la mà mitjançant sensors de RADAR per la HCI. Aquestes interfícies s'ocupen de l'estudi de la interfície entre els éssers humans i ordinadors. L'ús de RADARS i altres sensors de radiofreqüència per desenvolupar HCI basades en el HGR ha guanyat una atenció creixent en l'última dècada (Figura 4.1). Avui en dia, els dispositius porten incorporats RADARS per reconèixer i categoritzar els moviments de la mà.

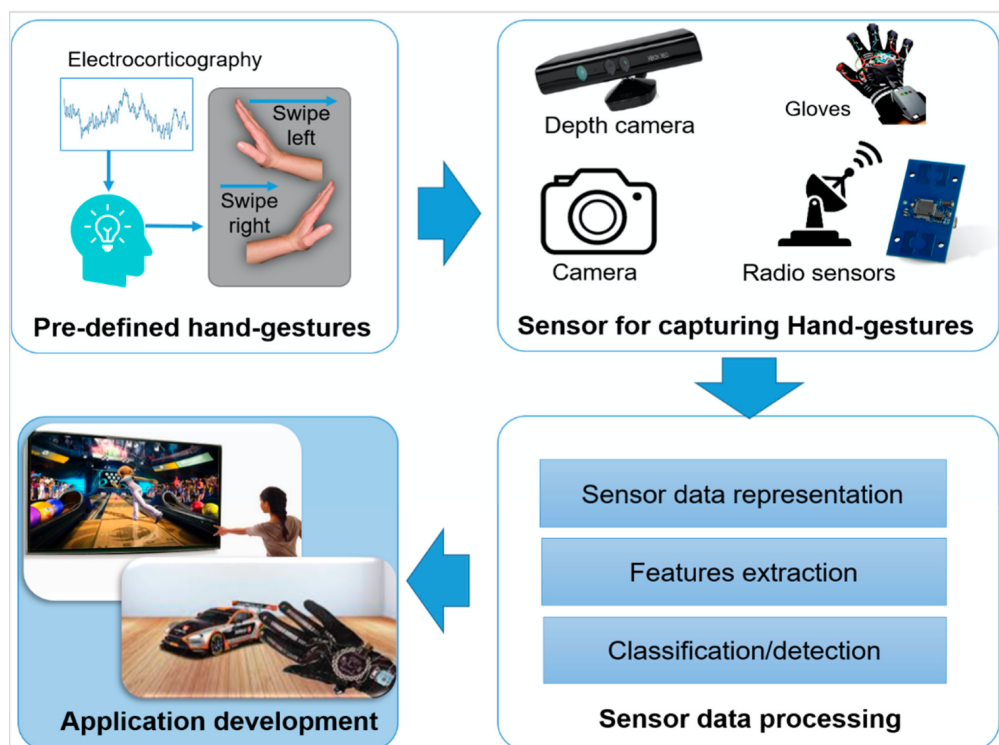


Figura 4.1. Esquema dels diferents sensors que poden ser usats en una aplicació de detecció de gestos.

A la Taula 4.1 es resumeix la comparació de sensors *wearable* i *non-wearable* (sense fils) utilitzats per reconèixer gestos de mans. Pot veure's com ambdós tipus de tecnologia tenen els seus avantatges i inconvenients i se'n pot seleccionar una o altra d'acord amb els requeriments de l'aplicació en consideració. Tot i així, dins els sensors *non-wearable*, els sensors RADAR tenen molts beneficis respecte els sistemes de reconeixement basats en càmera: el RADAR no és afectat per condicions de llum i no hi ha polítiques de privacitat relacionades. Els usuaris, normalment, no se senten còmodes sent observats constantment per una càmera.

Criteri de comparació	Sistemes <i>wearable</i>	Sistemes sense fils
Qüestions relacionades amb salut	Poden causar desconfort als usuaris, ja que sempre han de dur guants o altres sensors.	No causen cap problema o desconfort relacionat amb la pell.
Rang d'operació/sensorial	Normalment alt, si la transmissió sense fils és compatible.	Opera en un ambient contingut. Normalment es requereix la LoS entre la mà i els sensors.
Conveniència d'ús	Menys convenient (pel cas d'HCI): els usuaris sempre han de dur un sensor.	Els usuaris no han de dur cap sensor.
Simultaneïtat de reconeixement de múltiples usuaris/mans dins la mateixa àrea	Poden detectar-se gestos de diferents usuaris simultàniament en una única localització.	En una localització, la capacitat de reconeixement està normalment limitada a un nombre específic d'usuaris/mans.
Sensibilitat a les condicions de <i>background</i> (com ara soroll)	Normalment menys sensible a les condicions ambientals.	Més sensible que els sistemes <i>wearable</i> .
Qüestions de pèrdua del dispositiu	Pot ser perdut o oblidat en algun lloc.	No hi ha aquestes preocupacions ja que el sensor es fabrica normalment dins un dispositiu o és instal·lat en una certa localització.

Taula 4.1. Comparació dels sistemes *wearable* i sense fils per aplicacions de detecció de gestos amb les mans.

Els RADARs que estudien són els polsats (concretament, en la banda UWB) i de CW (SFCW i FMCW).

En aquest projecte, a més, es presenta la primera revisió relacionada amb l'HGR utilitzant sensors RADAR. Es revisen les tècniques disponibles per la representació de dades de gestos de la mà en múltiples dominis per diferents algorismes d'HGR basats en el processament de senyals i l'aprenentatge profund. Es classifiquen els RADARs utilitzats per HGR com RADARs d'ona contínua i polsada, i es presenten en detall tant el hardware com els detalls algorítmics de cada categoria.

4.2. Sistema de detecció de gestos utilitzant xarxes neuronals convolucionals.

L'objectiu d'aquest punt és realitzar un seguit de gestos amb la mà i que, mitjançant l'adquisició del moviment a través del RADAR i el seu processat corresponent amb *Matlab*, es puguin distingir entre ells.

Es pretén identificar un conjunt de gestos. En aquest projecte s'han realitzat 7 gestos diferents, tot i que es podria ampliar. La característica que permetrà la identificació consistirà en enregistrar la distància o *range* en funció del temps retornat pel RADAR corresponent al moviment de la mà davant el RADAR. Per aquest motiu, i com que el RADAR és més sensible al moviment perpendicular a aquest que en paral·lel, la majoria de gestos són combinacions de moviments perpendiculars al RADAR.

Els gestos que s'han dut a terme són els següents:

- De davant cap enrere (gest *DavDarr*) (Figura 4.2).
- De darrere cap endavant (gest *DarrDav*) (Figura 4.3).
- De davant cap enrere i tornant endavant (gest *DavDarrDav*) (Figura 4.4).
- De darrere cap endavant i tornant endarrere (gest *DarrDavDarr*) (Figura 4.5).
- De darrere cap endavant dues vegades i tornant enrere (gest *DarrDavDarrDavDarr*) (Figura 4.6).
- Obrir, tancar i tornar a obrir la mà (gest *ObrirTancarObrirMa*) (Figura 4.7).
- Fer una pinça amb la mà, obrint i tancant (gest *PinçaDits*) (Figura 4.8).

A continuació, en les següents figures es pot observar la representació de cadascun d'ells:

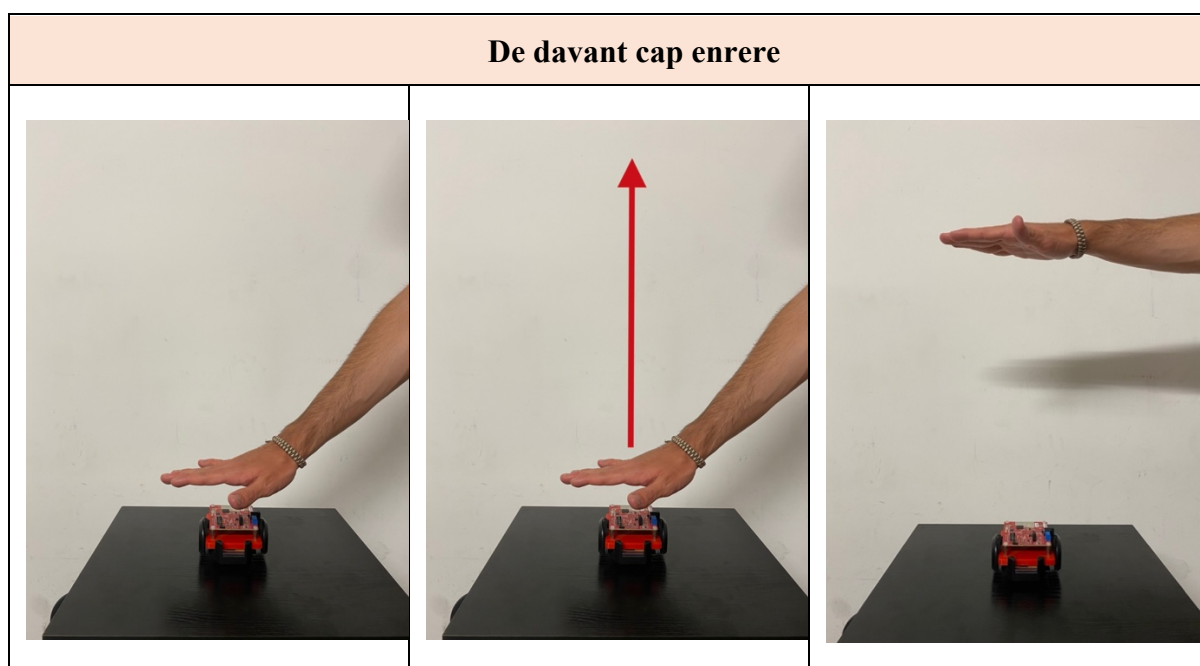


Figura 4.2. Successió de moviments pel gest *DavDarr*.

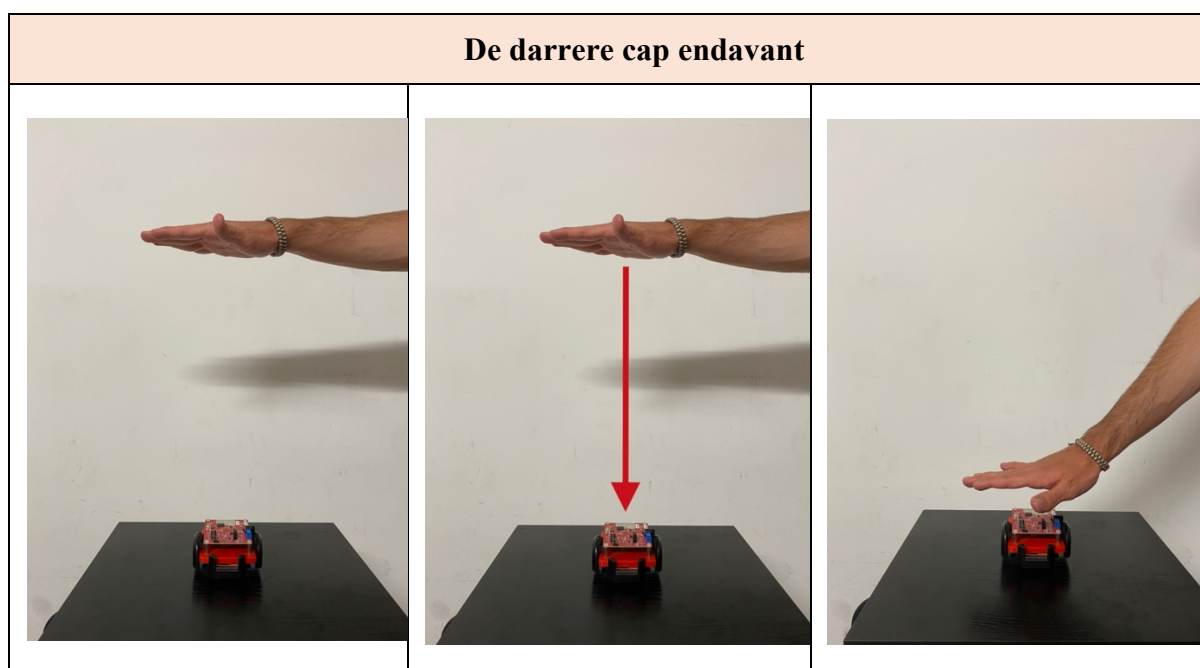


Figura 4.3. Successió de moviments pel gest *DarrDav*.

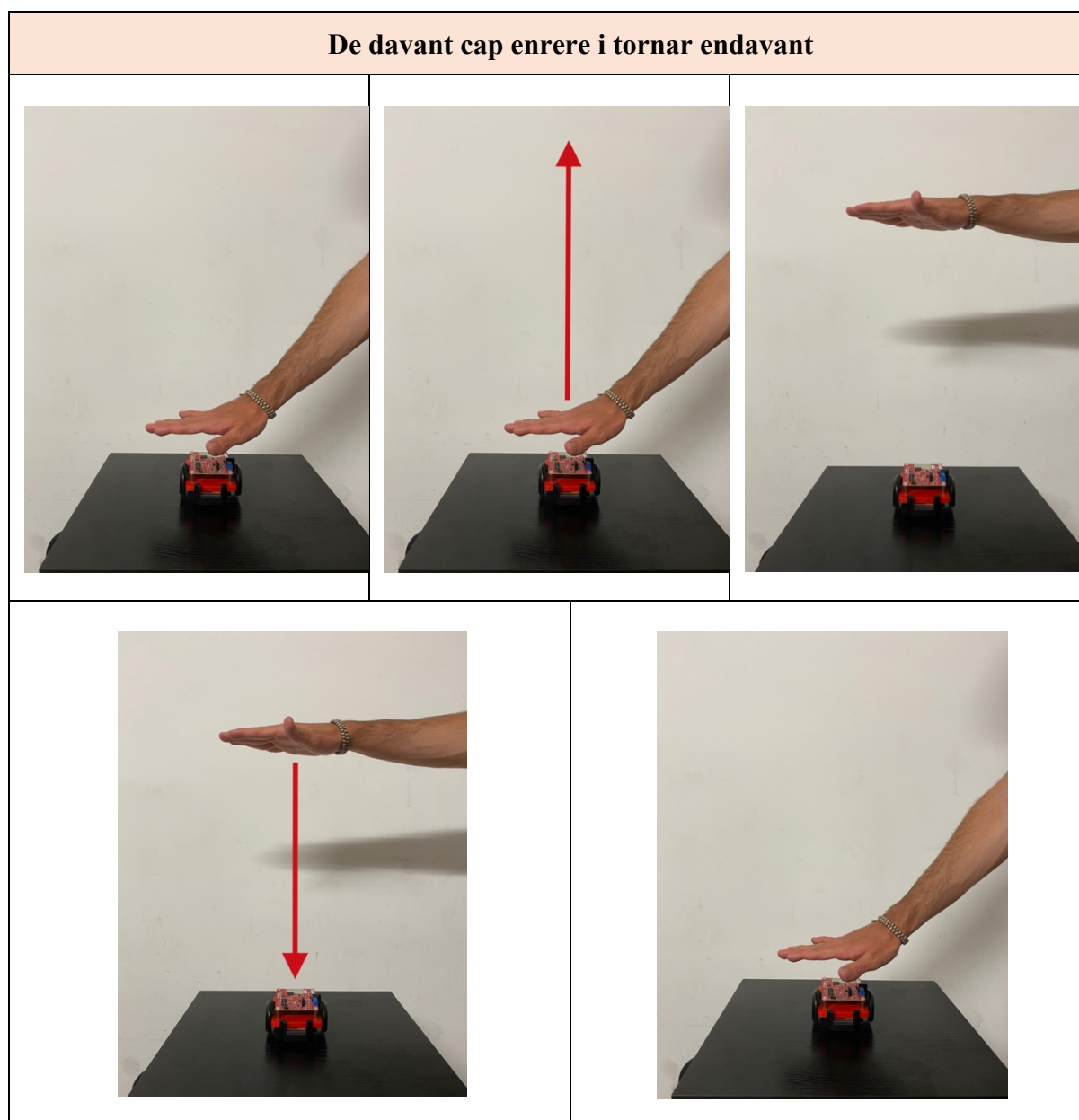


Figura 4.4. Successió de moviments pel gest *DavDarrDav*.

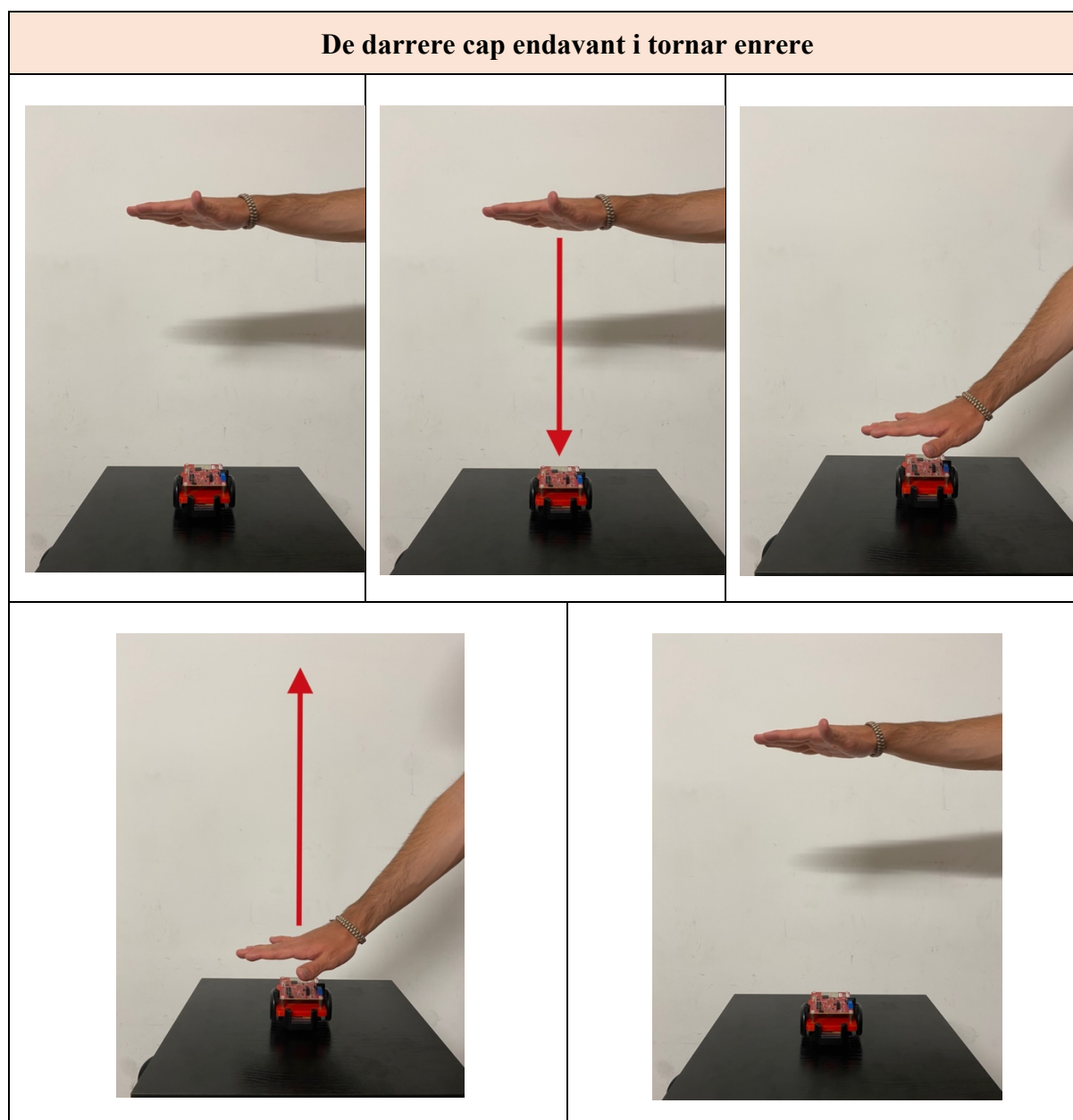


Figura 4.5. Successió de moviments pel gest *DarrDavDarr*.

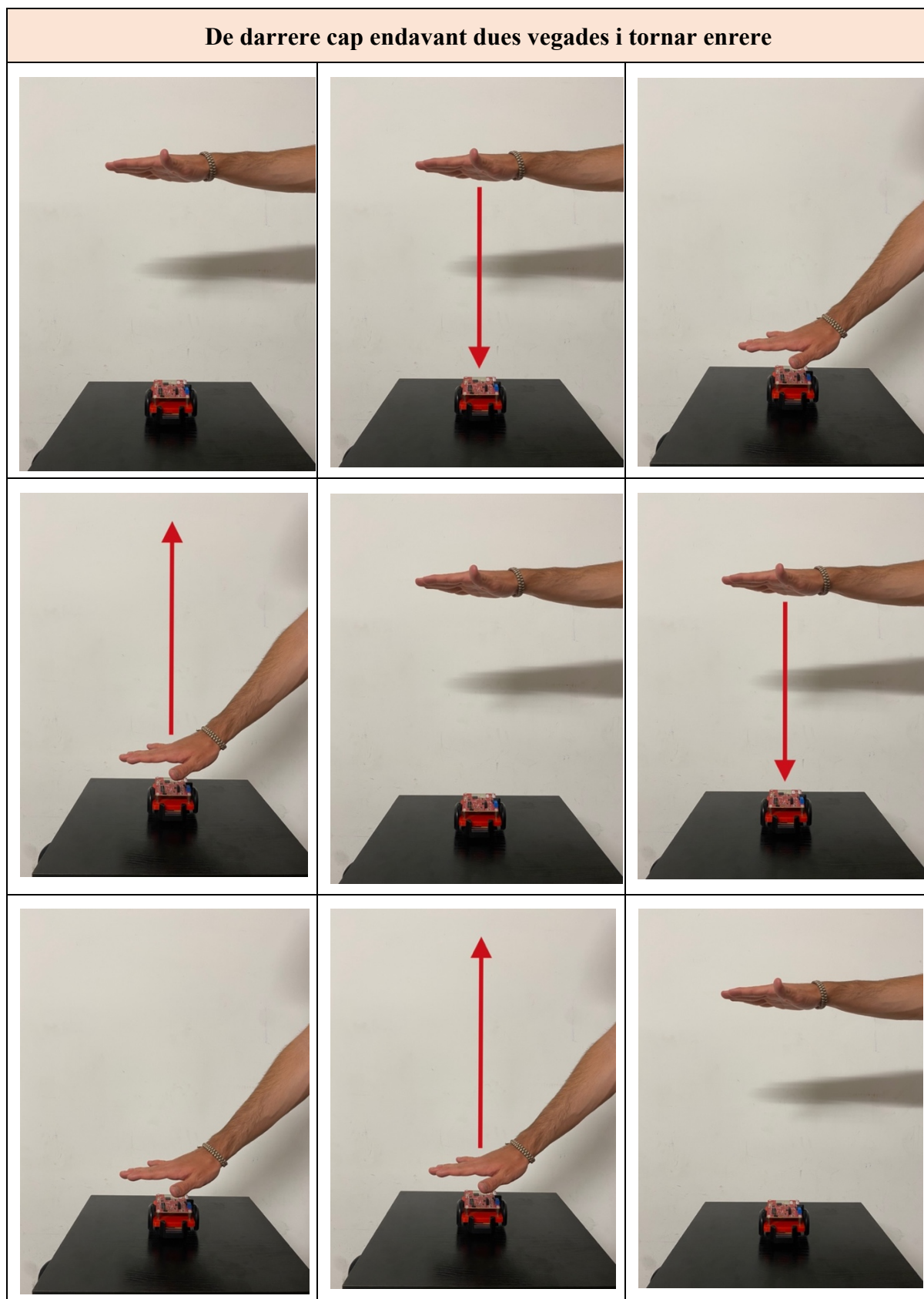
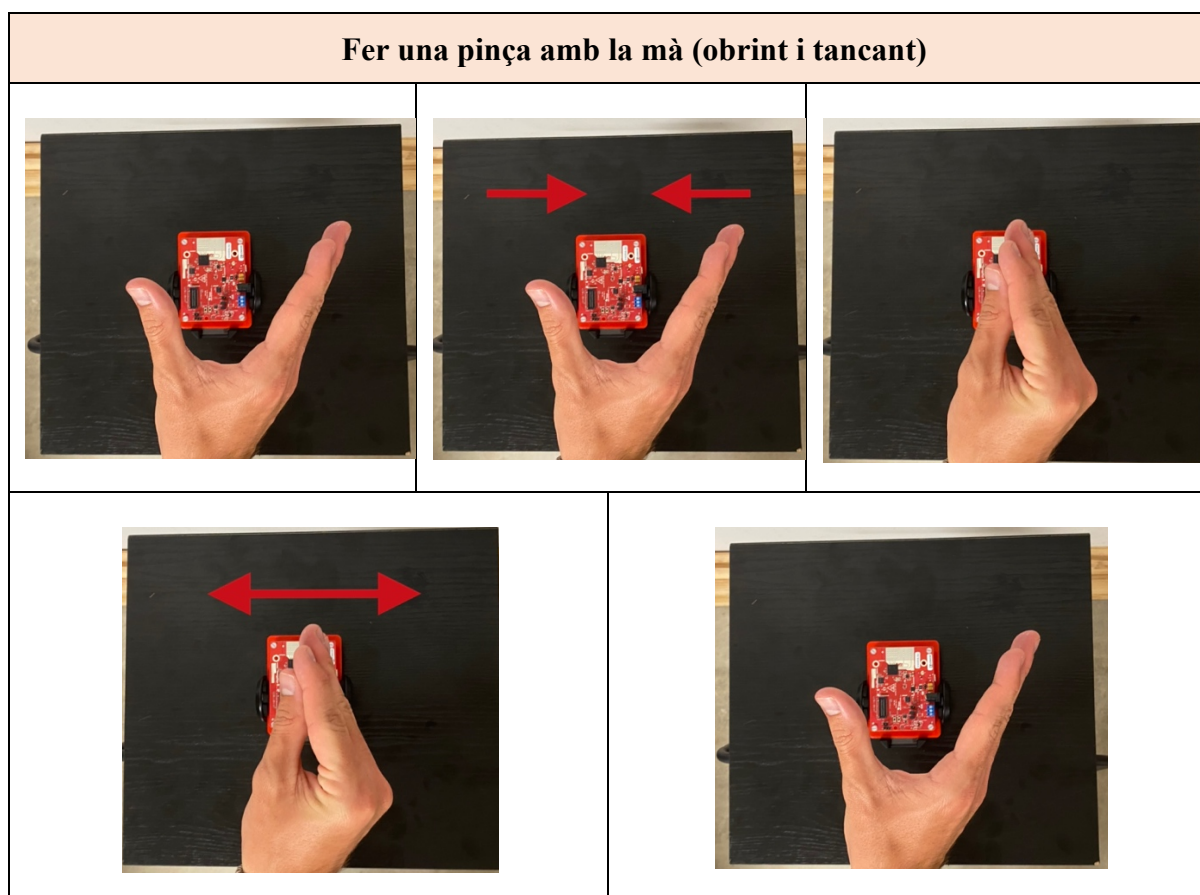


Figura 4.6. Successió de moviments pel gest *DarrDavDarrDavDarr*.

Figura 4.7. Successió de moviments pel gest *ObrirTancarObrirMa*.Figura 4.8. Successió de moviments pel gest *PinçaDits*.

La detecció de gestos consta de dues fases: la fase d'adquisició i la fase d'entrenament i prova.

4.2.1. Fase d'adquisició.

La primera consisteix en reproduir cadascun dels gestos un nombre considerat de vegades per tal de poder entrenar i comprovar la xarxa neuronal. S'ha optat perquè aquest nombre sigui 50, un nombre elevat per tal de poder entrenar i verificar el funcionament de la xarxa neuronal.

El programa utilitzat per prendre les mostres dels gestos s'anomena *prendreMesuresCasos.m*, semblant a l'utilitzat amb el vehicle però canviant el nom del fitxer segons el gest pertinent. Realitza la captura del moviment (a continuació es detallarà com es representa), converteix la figura en una imatge en blanc i negre i, posteriorment, la guarda a la carpeta que se li indica.

Cada captura del gest té una durada de 6 segons. D'aquesta manera, s'assegura que tots els moviments es duguin a terme en aquest període de temps.

Per crear la imatge es parteix d'una representació del rang en funció del temps. L'eix X, del temps, utilitza les dades del temps de les mostres preses en aquest interval de 6 segons. L'eix Y, el rang, es calcula de la forma:

$$rang = \sqrt{x^2 + y^2 + z^2} \quad (4.1)$$

La z sempre és 0, tal com s'ha explicat en el Capítol II i, per tant, acaba quedant que:

$$rang = \sqrt{x^2 + y^2} \quad (4.2)$$

Les captures s'han fet de manera que només s'agafin els punts on tant el valor absolut d'x com d'y sigui menor a 0.3 metres per tal de limitar la interferència de moviments de la mà i altres objectes fora de la regió d'identificació de gestos.

Per tal de fer el tractament posterior de les imatges, cal que siguin totes de la mateixa mida. S'ha optat perquè siguin 100×100. Per tant, cal que les dades anteriors s'adaptin a un vector de 100 posicions. Per fer-ho, s'han creat dos vectors auxiliars: un de temps, de 100 mostres i que va de 0 a 6, i un de rang, de 100 mostres i que va de 0 a 0.25 (s'ha optat per aquesta distància com a màxima). D'aquesta manera, els vectors es guardaran en una matriu o imatge de 100×100.

A continuació, es realitza una discretització del temps i rang mesurat als valors fixos que es guardaran en la imatge (Codi 4.1). Mitjançant un bucle, es recorren totes les posicions del vector de rang mesurat amb el RADAR i per cada mostra de rang i temps mesurades se li resten les auxiliars, respectivament, i es troba el mínim.

```
[m, IndexR]=min(abs(r-Range(i)));
[m, IndexT]=min(abs(temps-Time(i)));
```

Codi 4.1. Discretització de temps i rang.

De l'índex corresponent que en surti, tant pel temps com rang, se li assigna a la imatge el valor de l'SNR (Codi 4.2).

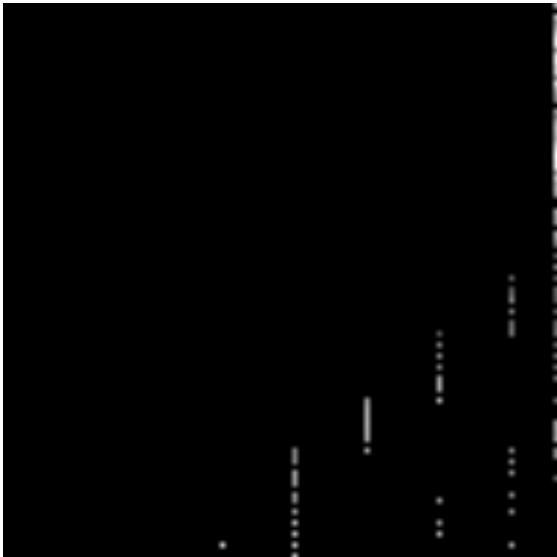
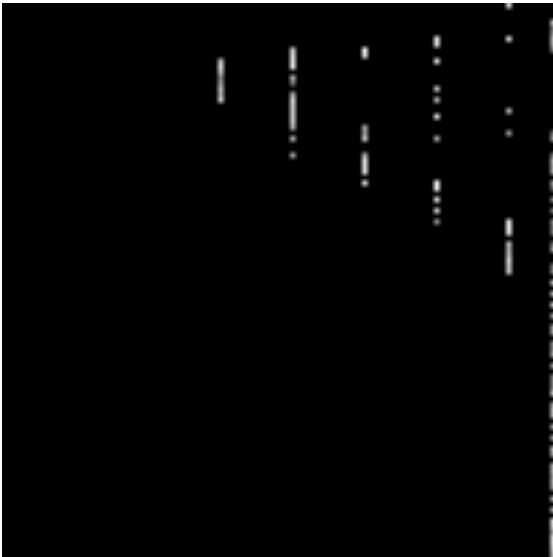
```
Imatge(IndexT, IndexR)=SNR(i);
```

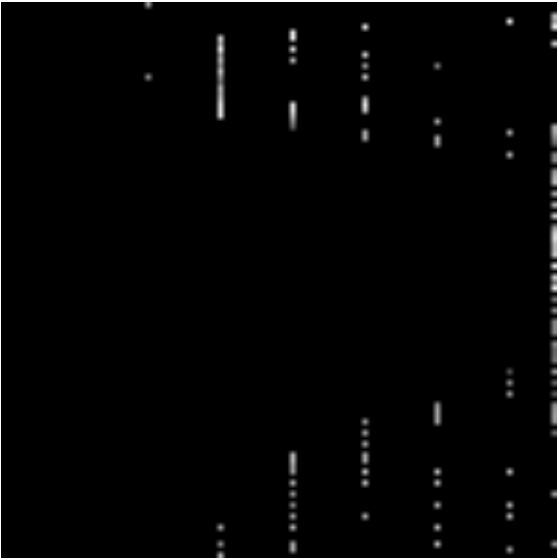
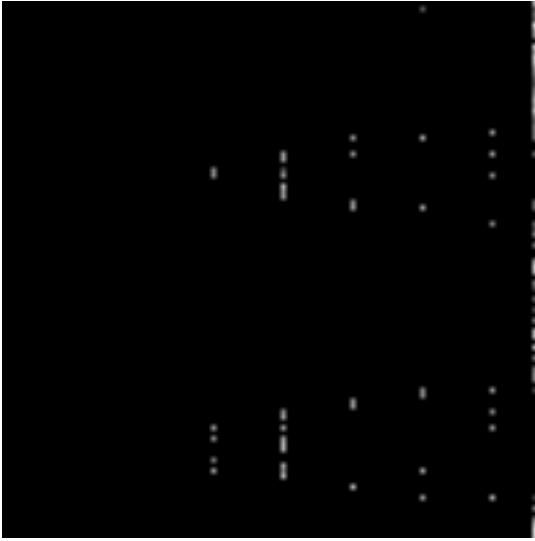
Codi 4.2. Assignació del valor de l'SNR a la imatge.

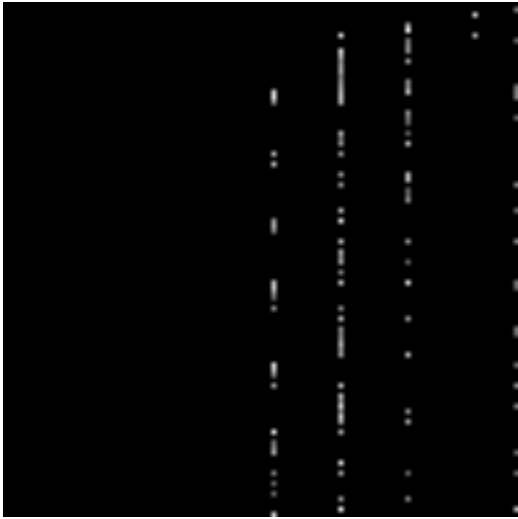
D'aquesta manera, mitjançant la funció *mat2gray* es converteix a escala de grisos i, amb la funció *imwrite*, es pot guardar a la carpeta que s'hi especifiqui. Es duu a terme d'aquesta manera ja que la imatge en escala de grisos requereix de menys espai que d'altres formats.

Matlab, al guardar i representar les imatges, les gira 90° a la dreta. Per tant, l'eix de temps acaba sent l'Y (de dalt a baix) i, el de rang, l'X (d'esquerra a dreta).

A continuació, es poden veure les figures amb les imatges de cada gest dins la Taula 4.2. Pel cas de 'Res', simplement és la mateixa imatge repetida 50 vegades. Serveix per casos en què es pot produir un moviment prop del RADAR per equivocació i no s'hi vol realitzar cap gest.

DarrDav	DavDarr
 Figura 4.9. Captura del moviment <i>DarrDav</i>. El rang comença lluny (dalt dreta) i es va apropant (baix esquerra), seguint el moviment de la mà en el temps.	 Figura 4.10. Captura del moviment <i>DavDarr</i>. El rang comença a prop i es va allunyant, seguint el moviment de la mà en el temps.

DarrDavDarr	DavDarrDav
	
Figura 4.11. Captura del moviment <i>DarrDavDarr</i>. El rang comença lluny, s'apropa i torna a allunyar-se, seguint el moviment de la mà en el temps.	Figura 4.12. Captura del moviment <i>DavDarrDav</i>. El rang comença a prop, s'allunya i torna a apropar-se, seguint el moviment de la mà en el temps.
DarrDavDarrDavDarr	ObrirTancarObrirMa
	
Figura 4.13. Captura del moviment <i>DarrDavDarrDavDarr</i>. El rang comença lluny, s'apropa, s'allunya, es torna a apropar i finalment s'allunya, seguint el moviment de la mà en el temps.	Figura 4.14. Captura del moviment <i>ObrirTancarObrirMa</i>.

PinçaDits	Res
 Figura 4.15. Captura del moviment <i>PinçaDits</i>. Per fer aquest moviment, cal fer entre 4 i 5 vegades la pinça amb els dits.	 Figura 4.16. Captura del moviment <i>Res</i>. No hi ha cap punt ja que es tracta d'un gest per classificar aquells moviments erronis.

Taula 4.2. Imatges de tots els gestos duts a terme per l'aplicació de detecció de gestos.

4.2.2. Fase d'entrenament i prova.

Seguidament, cal aplicar els algorismes d'entrenament i classificació. Caldrà utilitzar el programa *RADARclasificacnn.m*. Entrena i crea una Xarxa Neural Convolucional (CNN) per classificació *deep learning*. Aquest tipus de xarxes són eines essencials per *deep learning* i especialment per reconeixement d'imatge [5].

Les fases que es desenvolupen són:

- 4.2.2.1. Càrrega de les imatges.
- 4.2.2.2. Definició de l'arquitectura de xarxa.
- 4.2.2.3. Especificació de les opcions de *training*.
- 4.2.2.4. Entrenament i obtenció de la xarxa.
- 4.2.2.5. Testeig de la xarxa.

La Figura 4.17 il·lustra un diagrama de flux sobre el procediment seguit en la fase d'entrenament i prova de la CNN.

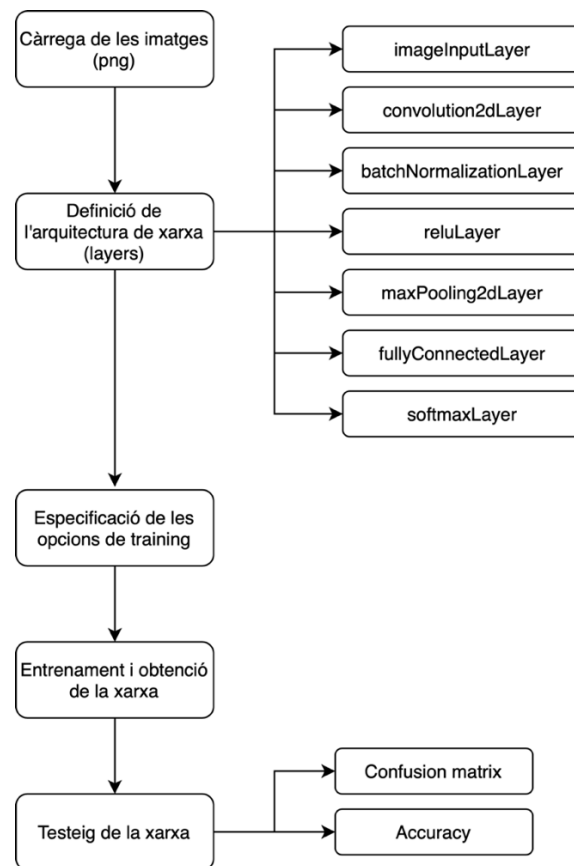


Figura 4.17. Diagrama de flux de la fase d'entrenament i prova de la CNN.

4.2.2.1. Càrrega de les imatges.

Per carregar les imatges s'utilitza un simple *image datastore* (funció *imageDatastore*, Codi 4.3). Aquesta funció etiqueta automàticament les imatges en base als noms dels seus fitxers. Per tant, dins el projecte, hi ha una carpeta i una subcarpeta anomenades 'Dataset' > 'Images' i, en aquesta última, hi ha 8 carpetes amb les imatges de cada gest i el seu nom corresponent.

```

digitDatasetPath = fullfile(pwd, 'DataSet4/Images');
imds = imageDatastore(digitDatasetPath, ...
    'IncludeSubfolders', true, ...
    'LabelSource', 'foldernames');
  
```

Codi 4.3. Càrrega de les imatges amb *imageDatastore*.

Seguidament, es divideixen les dades en *sets* d'entrenament i validació (Codi 4.4), de manera que la categoria d'entrenament tindrà 30 imatges i la de validació en tindrà les restants de cada etiqueta (és a dir, 20 imatges). És una relació 60-40% entrenament-validació. La funció *splitEachLabel* divideix la *image datastore* en dos nous *datastores* (un per entrenament i l'altre per validació). A més, també s'hi indiquen el nombre de gestos que s'han realitzat a *NumberOfClasses*.

```

numTrainFiles = 30;
NumberOfClasses=8;
[imdsTrain,imdsValidation] =
splitEachLabel(imds,numTrainFiles,'randomize');
  
```

Codi 4.4. Separació dels sets d'entrenament i validació.

4.2.2.2. Definició de l'arquitectura de xarxa.

Es defineix l'arquitectura de la xarxa neuronal convolucional (Codi 4.5). Cal especificar-ne la mida de les imatges a la capa d'input de la xarxa i el nombre de classes a la capa *fully-connected* abans de la capa de classificació. Cada imatge, com s'ha explicat, és de 100×100 píxels.

```
layers = [
    imageInputLayer([100 100 1])
    convolution2dLayer(3,8,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,16,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,32,'Padding','same')
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(NumberOfClasses)
    softmaxLayer
    classificationLayer];
```

Codi 4.5. Definició de les capes de la CNN.

Com podem observar, la xarxa es compon de 7 capes, algunes d'elles en repetició. La descripció de les diferents capes la podem trobar a la Taula 4.3 [6].

Capa	Descripció
<i>imageInputLayer</i> Input Layer	La <i>image input layer</i> introdueix una imatge 2-D a una xarxa i aplica una normalització de les dades.
<i>convolution2dLayer</i> Convolution and Fully Connected Layer	Una <i>convolution layer 2-D</i> aplica filtres convolucionals d' <i>sliding</i> a l'entrada.
<i>batchNormalizationLayer</i> Normalization, Dropout and Cropping Layers	La <i>batch normalization layer</i> normalitza un mini-lot de dades en totes les observacions per cada canal de forma independent. Per accelerar l'entrenament de la xarxa neuronal convolucional i reduir la sensibilitat a la inicialització de la xarxa, s'utilitzen capes de normalització per lots entre les capes

	convolucional i les no lineals, com les capes ReLU.
<i>reluLayer</i> Activation Layers	Una capa ReLU (<i>Rectified Linear Unit</i>) realitza una operació de llindar en cada element de l'entrada, on qualsevol valor menor a zero es posa a zero.
<i>maxPooling2dLayer</i> Pooling and Unpooling Layers	Una <i>max pooling layer</i> realitza el <i>downsampling</i> dividint l'entrada en regions de <i>pooling</i> rectangulars i calculant el màxim en cada regió.
<i>fullyConnectedLayer</i> Convolution and Fully Connected Layers	La <i>fully connected layer</i> multiplica l'entrada per una matriu de pesos i després afegeix un <i>bias vector</i> . Un <i>bias vector</i> és un conjunt addicional de pesos en una xarxa neuronal que no requereix cap entrada, i això correspon a la sortida d'una xarxa neuronal artificial quan té una entrada zero. El <i>bias</i> representa una neurona addicional inclosa en cada capa de pre-sortida i emmagatzema el valor d'1 per cada acció.
<i>softmaxLayer</i> Output Layers	La <i>softmax layer</i> aplica la funció <i>softmax</i> a l'entrada d'aquesta capa. Aquesta funció és el mateix que, a partir d'una matriu n , fer: $\exp(n)/\sum(\exp(n))$
<i>classificationLayer</i> Output Layers	La <i>classification layer</i> calcula la <i>cross-entropy loss</i> (s'utilitza per ajustar els pesos del model durant l'entrenament) per tasques de classificació i classificació ponderada amb classes mútuament excloents.

Taula 4.3. Descripció de totes les capes utilitzades en la CNN.

4.2.2.3. Especificació de les opcions de *training*.

Aquesta fase consisteix en especificar els paràmetres d'entrenament (Codi 4.6).

```
options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.01, ...
    'MaxEpochs',40, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',10, ...
    'Verbose',false, ...
    'Plots','training-progress');
net = trainNetwork(imdsTrain, layers, options);
```

Codi 4.6. Especificació dels paràmetres d'entrenament i obtenció de la xarxa *net*.

4.2.2.4. Entrenament i obtenció de la xarxa.

A continuació, es pot observar la fase de *training*. Es representa mitjançant la Figura 4.18 i consta de dos plots: un d'*accuracy* (precisió) i un altre de *loss* (pèrdues), que van en consonància. Es duen a terme 40 iteracions o cicles de *training* i, cada 10 cicles, es produeix la validació (ratlla negra). Mitjançant la validació es pot calcular l'*accuracy*, que pot observar-se que acaba sent del 97.47 %.

Per tant, es pot determinar que la xarxa neuronal ha estat molt ben entrenada i pot realitzar prediccions amb molta precisió.

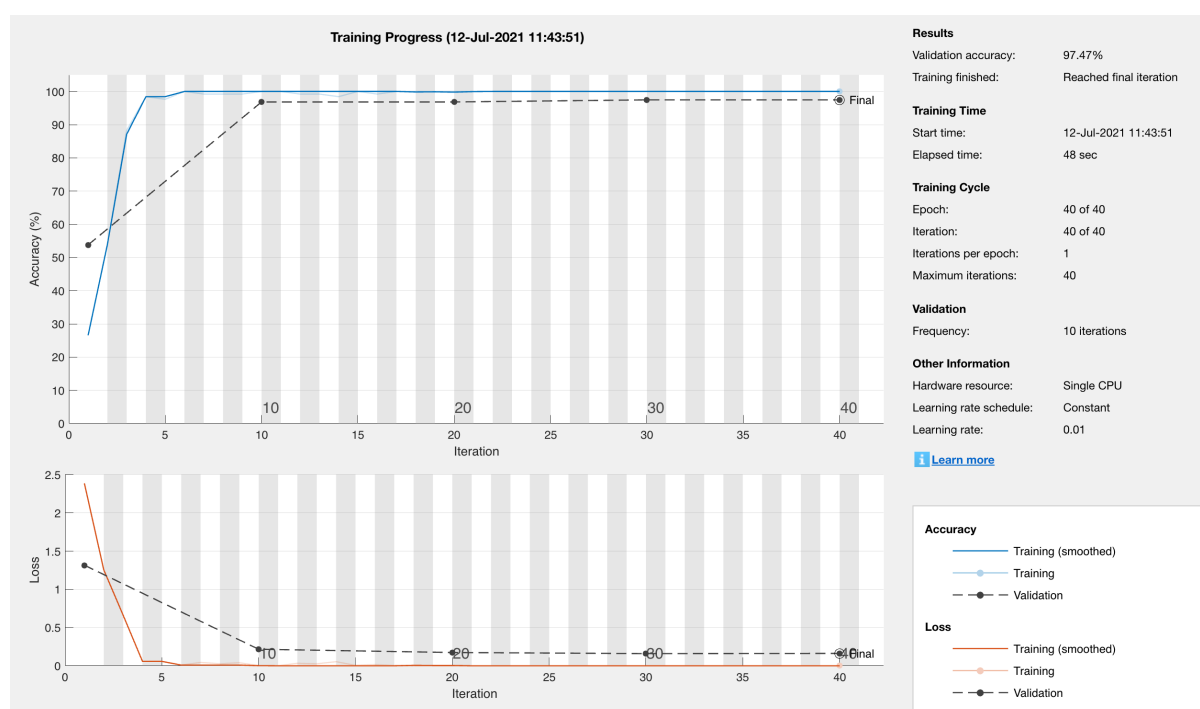


Figura 4.18. Interfície mostrada durant la fase de *training*.

S'obté la xarxa neuronal en una variable que rep el nom de '*net*' (provinent de '*network*').

4.2.2.5. Testeig de la xarxa.

En aquesta fase es fa una predicció de les etiquetes (*labels*) de les noves dades i càlcul de la precisió de la classificació (*accuracy*). S'agafen les imatges no utilitzades en el *training* i se'n fa una predicció (Codi 4.7).

```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

Codi 4.7. Classificació i validació de les imatges. Càlcul de la precisió de la xarxa.

Es pot observar la matriu de confusió en la Figura 4.19. L'eix X representa la classe predita i l'eix Y, la classe verdadera. Com pot veure's, la majoria dels casos estan en la diagonal. És el resultat esperat. Les prediccions en tots els gestos estan entre el 90% i el 100%. El cas més confós ha estat el *DarrDavDarrDavDarr*, on hi ha hagut dues equivocacions. També hi ha hagut una equivocació tant amb el gest *DarrDavDarr* i *DavDarrDav*. Això és degut a què són gestos força semblants.

L'*accuracy* calculada dona el mateix que es pot observar en la Figura 4.18, un 97.47 %.

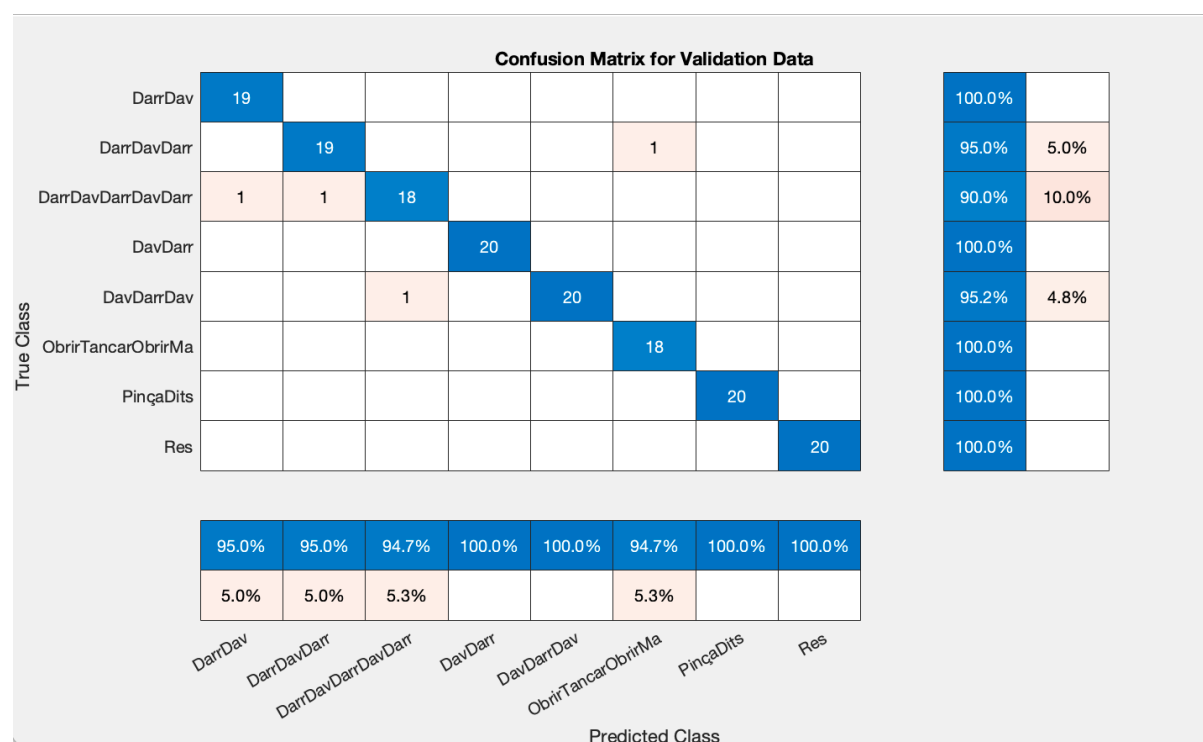


Figura 4.19. Matriu de confusió per l'aplicació de detecció de gestos.

Una vegada es té la xarxa neuronal entrenada, i emmagatzemada en la variable *net*, es guarda en un fitxer *.mat* per poder ser utilitzada en altres programes.

S'ha creat un últim programa, semblant al *prendreMesuresCasos.m*, anomenat *gestosTempsReal.m*. Aquest programa consta d'un bucle que segueix els següents passos:

1. Detecta si hi ha cap moviment a 30 cm del RADAR.
2. En cas afirmatiu, espera 1.5 segons i comença la captura del gest durant 6 segons. S'ha optat per un temps d'espera ja que és aproximadament el temps que passa entre que la mà s'aproxima al RADAR i es col·loca correctament per començar a realitzar el gest.
3. Una vegada acabada la captura, es converteix a imatge en blanc i negre, tal com s'ha explicat anteriorment, i s'emmagatzema en una carpeta del projecte.
4. Finalment, es classifica la imatge (I) mitjançant el Codi 4.8.

```
load net
label = classify(net,I);
resposta=['El gest correspon a ' char(label)];
```

Codi 4.8. Classificació de la imatge.

En primer lloc, es carrega la xarxa neuronal net del fitxer .mat. A la variable label s'hi assigna, arran de la classificació, el nom del gest al qual s'ha detectat que correspon la imatge i es mostra la resposta tant a la gràfica de detecció com pel terminal.

Per exemple, si es realitza un exemple amb el gest *PinçaDits*, el resultat mostrat és el de la Figura 4.20.

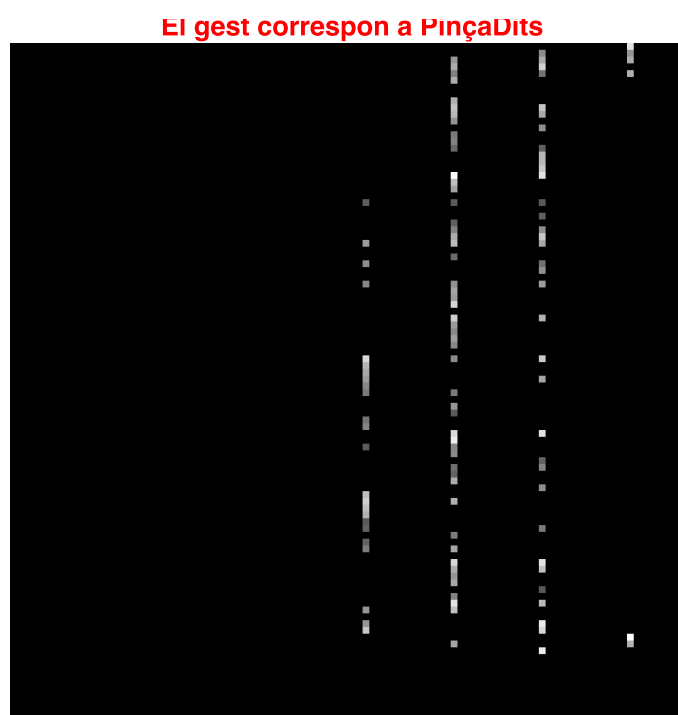


Figura 4.20. Exemple de detecció pel gest *PinçaDits*.

Es mostra tant el nom del gest com la imatge del gest capturat.

5. Es deixen passar 4 segons i es torna al punt 1.

El procediment detallat en paraules pot veure's també reflectit en el diagrama de flux de la Figura 4.21.

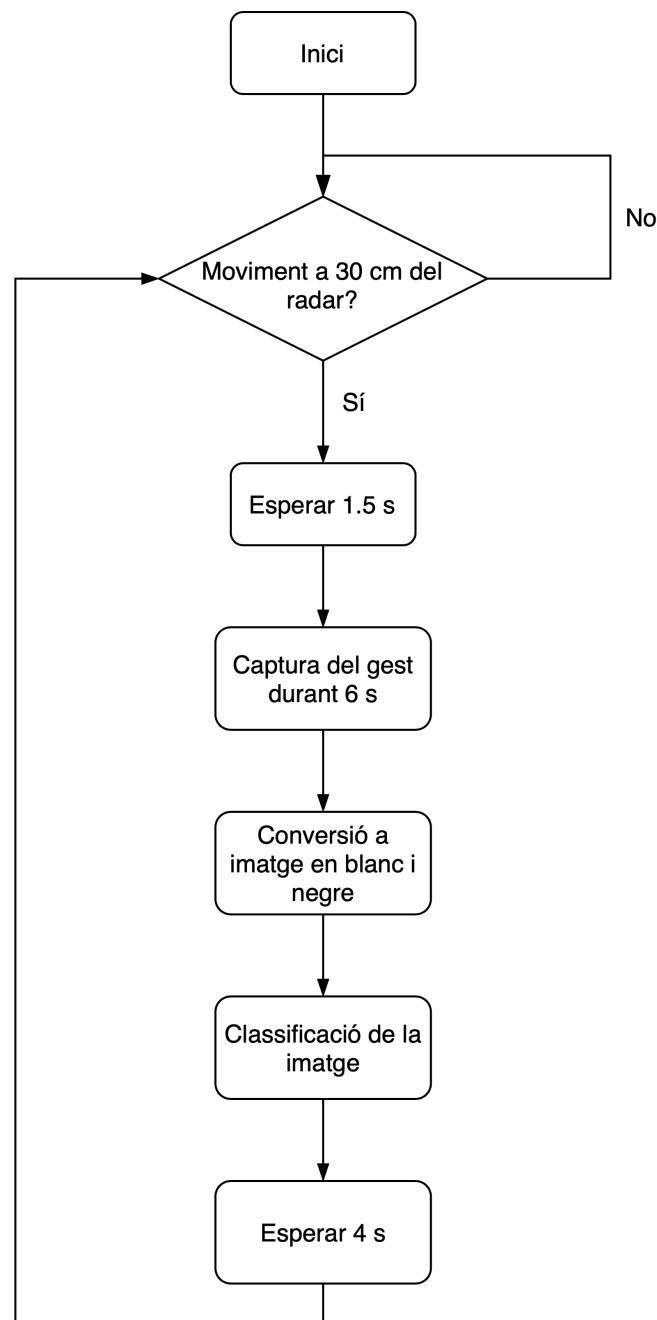


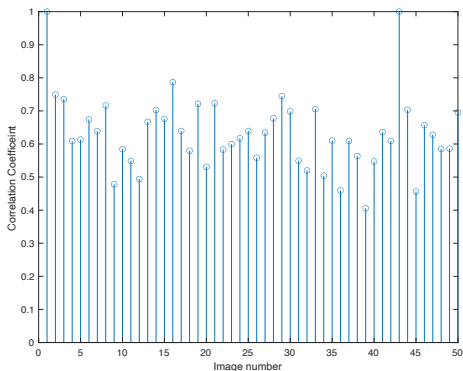
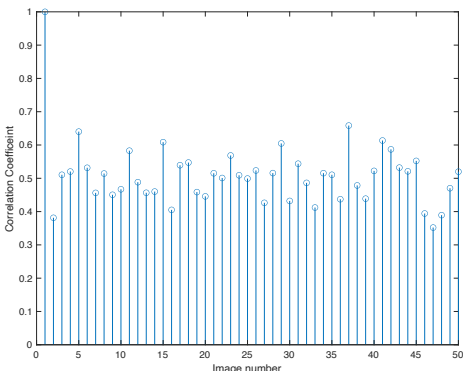
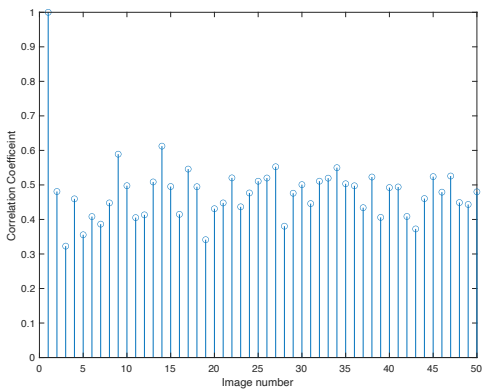
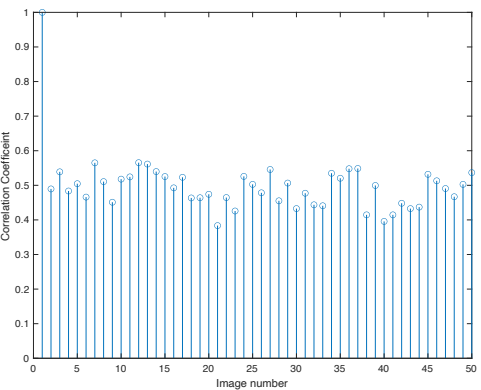
Figura 4.21. Diagrama de flux sobre el funcionament del programa *gestosTempsReal.m*.

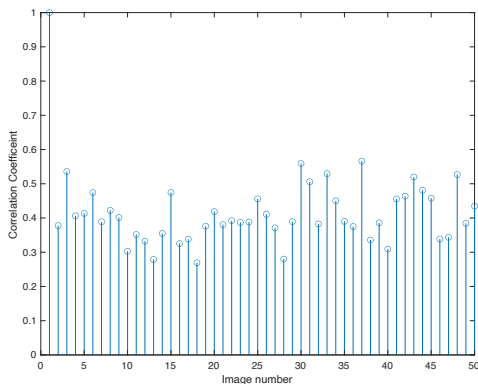
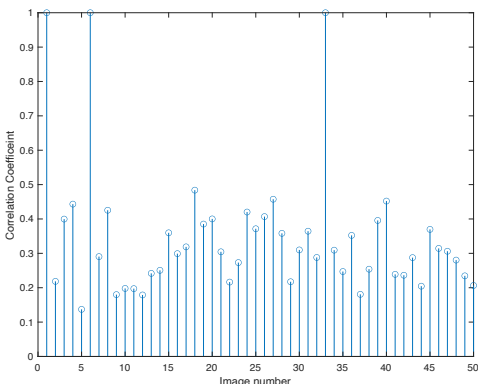
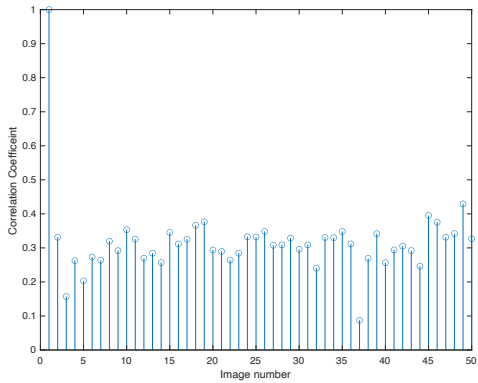
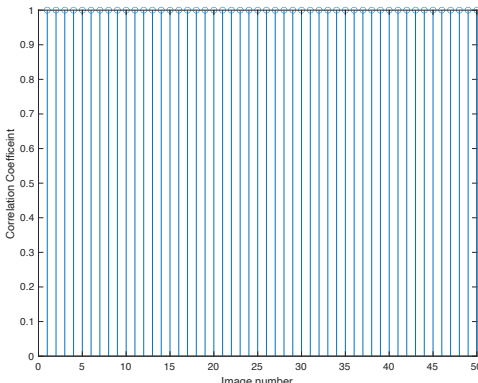
Amb aquesta explicació finalitza la part de detecció de gestos mitjançant el RADAR i una xarxa neuronal entrenada.

En la referència [7] s'hi pot veure un vídeo demostratiu sobre el funcionament de l'aplicació de reconeixement de gestos a temps real.

4.2.3. Correlació de les imatges dels gestos realitzats.

A continuació es farà una correlació entre les imatges del mateix tipus. Una correlació és un càlcul que estableix la semblança entre les imatges en format numèric entre el '0' (gens semblants) i l'1' (idèntiques). D'aquesta manera, es podrà observar quina relació mantenen entre elles (Taula 4.4).

DarrDav. Correlació mitja = 0.6330	DarrDavDarr. Correlació mitja = 0.5099
 Figura 4.21. Correlació pel gest <i>DarrDav</i>.	 Figura 4.22. Correlació pel gest <i>DarrDavDarr</i>.
DavDarrDav. Correlació mitja = 0.4791	DavDarr. Correlació mitja = 0.4996
 Figura 4.23. Correlació pel gest <i>DavDarrDav</i>.	 Figura 4.24. Correlació pel gest <i>DavDarr</i>.

DarrDavDarrDavDarr. Correlació mitja = 0.4179	ObrirTancarObrirMa. Correlació mitja = 0.3453
 Figura 4.25. Correlació pel gest <i>DarrDavDarrDav</i>.	 Figura 4.26. Correlació pel gest <i>ObrirTancarObrirMa</i>.
PinçaDits. Correlació mitja = 0.3173	Res. Correlació mitja = 1
 Figura 4.27. Correlació pel gest <i>PinçaDits</i>.	 Figura 4.28. Correlació pel gest <i>Res</i>.

Taula 4.4.

Pot observar-se com la correlació entre les imatges no és molt elevada però és suficient per concloure que tenen una semblança entre elles i puguin ser correctament classificades mitjançant la xarxa neuronal.

4.3. Referències.

- [1] SMITH, Karly A., et al. Gesture recognition using mm-wave sensor for human-car interface. IEEE sensors letters, 2018, vol. 2, no 2, p. 1-4.
- [2] FHAGER, Lars Ohlsson, et al. Pulsed millimeter wave RADAR for hand gesture sensing and classification. IEEE Sensors Letters, 2019, vol. 3, no 12, p. 1-4.
- [3] WANG, Yong, et al. A novel detection and recognition method for continuous hand gesture using fmcw RADAR. IEEE Access, 2020, vol. 8, p. 167264-167275.
- [4] AHMED, Shahzad, et al. Hand gestures recognition using RADAR sensors for human-computer-interaction: A review. Remote Sensing, 2021, vol. 13, no 3, p. 527.
- [5] MathWorks. Create Simple Image Classification Network.
Enllaç web: <https://ja.cat/ImageClassificationNetwork> (consulta: 16 de juny de 2021).
- [6] MathWorks. List of Deep Learning Layers.
Enllaç web:
<https://www.mathworks.com/help/deeplearning/ug/list-of-deep-learning-layers.html>
(consulta: 16 de juny de 2021).
- [7] Nil Munté Guerrero. Simulació de l'aplicació de gestos amb la mà. Data de creació: 13 de juliol de 2021. Data de pujada: 2 de setembre de 2021.
Enllaç web: <https://youtu.be/TX15-d1m26U>.

CAPÍTOL V

CONCLUSIONS I LÍNIES FUTURES

En aquest últim capítol de la memòria es resumiran les tasques dutes a terme en aquest projecte, s'analitzaran els resultats obtinguts i es detallaran les conclusions extretes per cada aplicació realitzada. A més, es descriuran les possibles línies d'investigació futures partint de la feina realitzada en aquest Treball de Fi de Grau.

5.1. Detecció de l'ocupació en vehicles.

En aquesta part del projecte s'ha desenvolupat una aplicació per la detecció d'ocupació en un vehicle de quatre i cinc places. Per una banda, s'ha desenvolupat un programa capaç de detectar el nombre d'ocupants a l'interior del vehicle i, partint d'aquest, s'ha desenvolupat una altra aplicació capaç de detectar la localització d'aquests ocupants.

Desenvolupar aquesta aplicació no ha estat una tasca fàcil degut als motius que s'exposaran a continuació. En primer lloc, capturar els punts. Es pot realitzar de dues maneres, tal com s'ha comentat en el Capítol II: la primera és retornar tots els objectes detectats. Aquesta configuració permet tenir una percepció de tot l'interior del vehicle però, partint d'aquesta, és molt complicat distingir d'entre seients, objectes i persones a l'interior del vehicle. Per tant, es va optar perquè el radar retornés només els punts d'aquells cossos en moviment.

En segon lloc, l'elaboració dels algorismes de decisió. Es van plantejar diverses tècniques per prendre la decisió sobre si el núvol de punts es tractava d'una persona o d'un objecte (clústers, algorisme *k-means*, etc.) i, finalment, es va acabar optant per l'algorisme explicat en el Capítol III. A més, el fet que només es retornessin punts d'objectes en moviment va ser un punt clau perquè l'algorisme funcionés amb èxit.

Pel que fa a l'APP amb *Matlab* i a la utilització de l'aplicació amb *Raspberry Pi*, són dues eines molt útils tant per facilitar l'ús de l'aplicació a usuaris sense coneixements de programació (podent seleccionar els paràmetres bàsics del vehicle en una *GUI*) com per poder adaptar l'aplicació a l'interior del vehicle sense necessitat d'haver d'utilitzar un ordinador portàtil, que ocupa un espai considerable.

Per acabar, l'aplicació tant d'ocupació com localització dels ocupants realitzada té una precisió alta, però és un sistema millorable. El fet que només es captin punts quan es detecta moviment fa que, si un usuari està totalment quiet, l'algorisme no hi detecti la seva presència i ho marqui com un espai buit. A més, si durant l'entrada dels usuaris al vehicle hi ha moviment en una zona on finalment no hi haurà ningú assegut, pugui confondre's com un espai ocupat durant les primeres deteccions.

Tot i això, s'han de tenir en compte dos aspectes: en primer lloc, deixar passar un marge de temps fins que la detecció es pugui considerar com a correcta (poden ser entre tres i quatre deteccions). En segon lloc que, en un cas real, una vegada el cotxe es posa en marxa els usuaris no es mouen a l'interior del vehicle. Per tant, deixant aquest marge de detecció i mantenint la detecció una vegada s'ha estabilitzat el moviment a l'interior del vehicle i el cotxe està en marxa (establint un mecanisme de comunicació entre el RADAR i el vehicle), la precisió en la detecció podria ser encara més alta. A més, el sistema encara podria millorar més si es fa una combinació de diverses deteccions passat el marge de temps inicial, establint un criteri a aplicar una vegada el cotxe es posa en marxa.

L'aplicació realitzada és una molt bona alternativa als sistemes de detecció d'ocupació actuals i és un sistema d'alta fiabilitat i precisió.

5.2. Sistema de detecció de gestos amb la mà basat en RADAR FMCW en la banda mil·limètrica.

En aquesta segona part del projecte s'ha desenvolupat una aplicació per la detecció de gestos amb la mà, tot utilitzant el mateix radar que en la detecció de l'ocupació en un vehicle.

La captura dels punts es realitza de la mateixa manera i només es fa per objectius en moviment. L'aplicació és integrable també a l'interior d'un vehicle i presenta una alta precisió en les proves realitzades. El temps de reconeixement és força ràpid i, per aquest motiu, l'aplicació és força integrable en una aplicació real. Tot i no haver-se realitzat en aquest projecte, podria ser de gran utilitat a l'hora de realitzar diverses accions sobre el vehicle sense produir cap mena de distracció al conductor: obrir i tancar finestres, obrir i tancar l'aire condicionat, canviar el canal de música, etc.

Tot i així, també presenta alguns aspectes que degut a la limitació del RADAR, no són del tot òptims. En primer lloc, com ja s'ha comentat, el RADAR només retorna punts en el pla horitzontal, és a dir, en la direcció de l'agrupació d'antenes virtual. Això fa que els gestos hagin de realitzar-se en aquest pla i siguin molt limitats i repetitius (amunt i avall successivament i en diverses combinacions). Aquest fet pot produir confusió al conductor, en certes ocasions, i portar-lo a realitzar un gest equivocat. L'ideal, per tant, seria haver pogut realitzar gestos en tots els plans i que el RADAR hagués estat capaç de convertir a imatge, ben diferenciada, de totes les combinacions.

Un altre aspecte a tenir en compte és que els gestos han de ser reproduïts exactament de la mateixa manera que es va fer durant l'entrenament i, per aquest motiu, s'ha establert un temps d'espera abans de començar la detecció i un temps en el qual es produeix aquesta detecció. En un entorn de proves, com el que s'ha fet en aquest projecte, es poden reproduir els gestos de manera correcta, tot i que en un entorn real s'hauria de millorar el sistema tenint en compte les condicions a l'interior del vehicle i del conductor (vibracions del vehicle, comoditat a l'hora de realitzar el gest, etc.).

Finalment, també s'hauria de tenir en compte la localització del RADAR de manera que els altres usuaris del vehicle no hi puguin intervenir ja que, com s'ha explicat, a l'aproximar-se al RADAR s'activa un temps d'espera per poder començar la captura i, en certes ocasions, podria arribar a ocasionar algun accident no desitjat i involuntari. Per tant, cal limitar-ho només a l'ús del conductor del vehicle.

5.3. Conclusions generals i línies futures.

Els objectius principals d'aquest projecte han estat la utilització d'un RADAR FMCW en la banda mil·limètrica per dues aplicacions a l'interior d'un vehicle. En aquest punt es presentarà el punt de partida a partir de la qual es poden millorar aquestes aplicacions i com es podria estendre a d'altres aplicacions fora de l'entorn del vehicle.

En primer lloc, es comentarà l'aplicació de detecció de l'ocupació en vehicles. Com s'ha explicat, és un sistema que funciona correctament però millorable. En primer lloc, s'hauria d'optimitzar per realitzar la detecció en un temps menor. A més, en un entorn real es podria implementar la comunicació amb el vehicle i un sistema optimitzat per tal de poder determinar l'ocupació de forma més precisa i efectiva.

Una possible solució per tal d'augmentar la precisió, sobretot per les places del darrere on, degut a la presència d'obstacles (seients, parets) l'encert és més baix, seria la utilització de dos RADARS. Un pels seients del davant i, un altre, pels de darrere. D'aquesta manera, es reduiria l'efecte rebot en un alt grau.

Per acabar, una línia futura d'aquesta aplicació seria l'extensió de la detecció d'ocupació en altres entorns com, per exemple, aules d'escola o, fins i tot, teatres. Serviria per poder realitzar un comptatge ràpid del nombre de persones i, fins i tot, saber de quines persones es tracten si aquestes disposen d'un seient associat. Per fer-ho, es podria fer de dues maneres. En cas de conèixer-se la localització dels seients, establir una matriu que els localitzés tots i així es podria utilitzar el mateix algorisme que en aquest projecte. En cas contrari, s'hauria de construir un algorisme capaç de detectar un núvol de punts i determinar si es tracta d'una persona. Una possible solució seria la utilització de l'algorisme *k-means*, un mètode de clústers i agrupaments de punts, per tal de fer el comptatge de persones [1].

Pel que fa a l'aplicació de detecció de gestos amb la mà, com també s'ha comentat s'haurien de poder ampliar el nombre de gestos als altres plans per tal de no realitzar gestos repetitius i que puguin portar a confusions. Una possible solució, abans de canviar de RADAR, seria també la utilització de dos RADARS en combinació: un pel pla x-y i un altre pel pla x-z, tot creant així un gest en 3-D.

Un altre aspecte a millorar és el sistema d'inici de la detecció. El mecanisme establert és correcte, però en un entorn real és molt probable que hi hagi moviments al voltant del radar i que per tant s'activi la detecció del gest. Per tant, s'hauria de trobar un mecanisme d'activació del sistema més precís i amb menys probabilitat de ser activat de forma fàcil per altres usuaris que no siguin el conductor.

Una línia de futur en la continuació d'aquesta aplicació seria establir un mecanisme perquè, una vegada reconegut el gest, es pogués determinar una escala de control dels elements del vehicle. Per exemple, la regulació del grau d'obertura o tancament de la finestra, el grau de calor o frescor de l'aire condicionat o la rapidesa en el moviment del parabrises. Una possible solució seria establir un límit inferior i un límit superior i, en funció de la situació de la mà durant un temps determinat, establir el grau en forma de percentatge del 0 a 100 %.

Per acabar, en un entorn d'integració real s'hauria de buscar la manera d'integrar el RADAR (o RADARS, en cas que l'aplicació requereix de l'ús de més d'un) a l'interior del vehicle de manera que a simple vista no fossin vistos i es poguessin utilitzar amb normalitat: el RADAR per la detecció d'ocupació situat frontalment i enfocant tot el vehicle i el RADAR pel reconeixement de gestos al costat del conductor i de fàcil accés per poder ser utilitzat en el moment necessari i sense obstacles entremig.

5.4. Referències.

- [1] ANTIC, Borislav, et al. K-means based segmentation for real-time zenithal people counting. En 2009 16th IEEE International Conference on Image Processing (ICIP). IEEE, 2009. p. 2565-2568.