

Treball de Fi de Grau  
Grau en Enginyeria Sistemes de Telecomunicacions

Implementació de la modulació direccional  
amb el SDR AD-FMComms5-EBZ.

David Sardà Martin

Director: Josep Parrón Granados

Departament de Telecomunicació i Eng. de Sistemes  
Universitat Autònoma de Barcelona (UAB)  
setembre 2, 2022

## *Agraïments*

---

Vull agrair al meu tutor del treball de fi de grau Josep Parrón per la seva paciència i suport, per l'ajuda i les seves recomanacions per poder acabar el treball.

També vull agrair a la meva família i a la meva novia pel suport que m'han donat al llarg del treball.

## *Resum*

---

La seguretat en la transmissió de dades ha sigut un camp d'estudi al qual se li ha donat molta importància recentment, inicialment se li donava una gran importància per l'ús militar de les comunicacions, però amb l'increment d'ús de tecnologies també se li ha donat més importància com per exemple amb les dades enviades a les aplicacions mòbils dels bancs. Per millorar la seguretat neixen esquemes de seguretat com la modulació direccional que en aquest projecte estudiarem. Aquests increments de la complexitat a la comunicació per millorar la seguretat o altres propòsits ajuda en el naixement de sistemes Software Defined Radio, en aquest projecte treballarem amb un SDR amb l'objectiu de crear un esquema de comunicació més segur amb la modulació direccional i estudiarem el seu comportament.

## *Abstract*

---

Security in the transmission of data has been a field of study which has been given a lot of importance recently, initially it was given great importance due to the military use of communications but with the increase of places to use technologies, it has become more important for example with the data sent to the mobile applications of the banks. To improve security, security schemes such as directional modulation are born, which we will study in this project. Due to the increases in the complexity of communication to improve security or other purposes Software Defined Radio Systems will be born, in this project we work with an SDR with the aim of creating a more secure communication scheme with directional modulation.

# *Índex de contingut*

---

1.	Introducció .....	1
1.1.	Estat de l'art.....	1
1.2.	Objectius.....	2
1.3.	Metodologia de treball.....	3
1.4.	Estructura del document .....	5
2.	Plataforma experimental .....	7
2.1.	Els Software Defined Radio .....	7
2.1.1.	Components d'un sistema SDR .....	8
2.1.2.	Avantatges i inconvenients d'un SDR .....	11
2.1.3.	Aspectes a considerar .....	12
2.1.4.	Aplicacions .....	13
2.2.	AD-FMComms5-EBZ .....	13
2.2.1.	AD9361 .....	14
2.3.	Muntatge experimental .....	16
3.	Anàlisi de les eines existents.....	20
3.1.	Conceptes teòrics previs.....	20
3.1.1.	Modulació QPSK .....	20
3.1.2.	Calibració .....	21
3.2.	IIO Oscilloscope .....	21
3.2.1.	Finestra de configuració .....	21
3.2.2.	Finestra de representació de dades .....	25
3.2.3.	Interacció amb el SDR .....	28
3.3.	Matlab i Simulink.....	31
3.3.1.	Treball Existent.....	31
3.3.2.	Particularitats .....	34
4.	Actualització amb les noves eines de Matlab .....	38

4.1.	Analog Devices Inc. Board support Packages .....	38
4.1.1.	Instal·lació .....	39
4.1.2.	Facilitats que ens ofereix la nova Toolbox.....	43
4.2.	Descripció de les noves funcions .....	44
4.2.1.	Objectes, estructures de dades per configurar el SDR, funcions principals .	44
4.2.2.	Eines per visualitzar els resultats en Matlab .....	46
4.3.	Exemples de configuració .....	47
4.4.	Calibració .....	53
4.4.1.	Transmissió amb el sistema calibrat .....	60
5.	Modulació direccional .....	62
5.1.	Introducció .....	62
5.2.	Fonaments teòrics per generar DM .....	64
5.3.	Implementació teòrica de la modulació direccional.....	69
5.3.1.	Anàlisi de rendiment.....	75
5.4.	Verificació experimental .....	79
6.	Conclusions i línies futures.....	89
7.	Bibliografia .....	91
8.	Annexes .....	94
8.1.	Codi .....	94
8.1.1.	Codi anterior .....	94
8.1.2.	Codi Enviar 4 tons .....	99
8.1.3.	Codi de Matlab pel controlador.....	104
8.1.4.	Calcular vector del canal .....	105
8.1.5.	Transmissió amb el canal calibrat .....	110
8.1.6.	Codi modulació direccional teòrica .....	114
8.1.7.	Decisors.....	116
8.1.8.	Codi de la modulació direccional teòrica enviant cadena de text amb decisor centroides .....	118
8.1.9.	Codi percentatge símbols correctes teòrics .....	121
8.1.10.	Codi transmetre modulació direccional .....	123

8.1.11.	Codi percentatge símbols correctes amb algoritme centroides.....	128
8.1.12.	Codi percentatge símbols correctes amb algoritme SVM .....	129

# *Índex de figures*

---

Figura 1 Sistema Software Defined Radio. ....	8
Figura 2 Arquitectura Fonamental SDR .....	8
Figura 3 Tipus de filtres.....	10
Figura 4 AD-FMComms5-EBZ i Zynq ZC706 .....	14
Figura 5 Diagrama funcional .....	16
Figura 6 Representació dels experiments realitzats. ....	17
Figura 7 Transmissors del SDR .....	17
Figura 8 Receptor connectat al SDR .....	18
Figura 9 Transmissor SDR connectat al controlador.....	18
Figura 10 SDR integrat a un ordinador .....	19
Figura 11 Constel·lació QPSK .....	20
Figura 12 Configuració Global AD9361 .....	22
Figura 13 Configuració Receptor AD9361 .....	23
Figura 14 Configuració Transmissor AD9361 .....	23
Figura 15 Configuració FPGA del transmissor AD9361 .....	24
Figura 16 Configuració FPGA del receptor AD9361 .....	25
Figura 17 Diagrama de blocs del AD9361 .....	25
Figura 18 Representació domini temporal amb IIO Oscilloscope.....	26
Figura 19 Representació domini freqüencial amb IIO Oscilloscope.....	27
Figura 20 Representació constel·lacions amb IIO Oscilloscope .....	27
Figura 21 Representació correlació creuada amb IIO Oscilloscope .....	27
Figura 22 Controlador que permet modificar l'angle del transmissor.....	28
Figura 23 Configuració IIO Oscilloscope per encendre el SDR (I) .....	29
Figura 24 Configuració IIO Oscilloscope per encendre el SDR (II) .....	29
Figura 25 Comprovació domini freqüència encesa del dispositiu.....	30
Figura 26 Comprovació domini temporal encesa del dispositiu .....	30
Figura 27 Document configuració dispositiu (ad9361.cfg) .....	31
Figura 28 Circuit Simulink per emissió i recepció senyal .....	32
Figura 29 Esquema Simulink del senyal enviat en el transmissor .....	32
Figura 30 Polsos quadrats.....	36
Figura 31 Senyal esperada i obtinguda al passar per un filtre .....	36
Figura 32 Add-on utilitzat per aplicar noves funcionalitats.....	38
Figura 33 Configuració de la connexió de xarxa amb el dispositiu. ....	39
Figura 34 Fitxer interfaces escrit a la targeta SD .....	40
Figura 35 Pantalla de verificació de la connexió .....	41

Figura 36 Pantalla de configuració pel codissem de HW o SW .....	41
Figura 37 Requeriments per la configuració de codissem de HW/SW .....	42
Figura 38 Add-ons instal·lats o actualitzats .....	42
Figura 39 Fitxer de configuració d'exemple .....	43
Figura 40 Configuració estructura transmissor.....	44
Figura 41 Configuració estructura receptor .....	44
Figura 42 Configuració estructura visualització de les dades.....	45
Figura 43 Estructura per la visualització senyal rebut .....	47
Figura 44 Visualització dades emeten un to pel canal.....	48
Figura 45 Visualització dades emeten un to pel canal per un receptor diferent al configurat .....	48
Figura 46 Emetre un to per cada transmissor amb diferència de 25 KHz entre ells. ....	49
Figura 47 Emetre un to per cada transmissor amb diferència de 25 KHz entre ells en un receptor diferent.....	51
Figura 48 Senyal rebut en el canal 1 durant transmissió QPSK .....	52
Figura 49 Senyal rebut en el canal 2 durant transmissió QPSK .....	52
Figura 50 Convolució circular de les dades rebudes amb vector de 0s excepte dos 1s.....	54
Figura 51 Símbols rebuts amb $fs=30.72\text{ MHz}$ .....	55
Figura 52 Anàlisi dades rebudes per una seqüència controlada amb $fs=30.72\text{ MHz}$ . ....	55
Figura 53 Símbols rebuts amb $fs=15.36\text{ MHz}$ .....	56
Figura 54 Anàlisi dades rebudes per una seqüència controlada amb $fs=15.36\text{ MHz}$ . ....	57
Figura 55 Constel·lació rebuda amb $G=30\text{ dB}$ emeten senyal QPSK.....	57
Figura 56 Constel·lació rebuda amb $G=32\text{ dB}$ emeten senyal QPSK.....	58
Figura 57 Anàlisi senyal rebuda amb $G=32\text{ dB}$ emeten senyal QPSK. ....	59
Figura 58 Visualització dades sense calibrar el sistema .....	60
Figura 59 Visualització dades havent calibrat el sistema .....	61
Figura 60 Idea general del propòsit de la modulació direccional .....	63
Figura 61 Arquitectura d'un transmissor de modulació direccional.....	63
Figura 62 Esquema fórmula modulació direccional .....	64
Figura 63 Esquema vectors $r$ i $x_n$ modulació direccional .....	65
Figura 64 Constel·lació rebuda en la direcció desitjada .....	70
Figura 65 Constel·lació rebuda $80^\circ$ amb DM.....	70
Figura 66 Constel·lació rebuda $120^\circ$ amb DM .....	70
Figura 67 Mòdul del senyal rebut amb DM .....	71
Figura 68 Fase del senyal rebut amb DM. ....	71

Figura 69 Variació símbols rebuts respecte direcció d'emissió amb DM .....	72
Figura 70 Constel·lació rebuda en la direcció $\Theta_0$ amb DM .....	73
Figura 71 Constel·lació rebuda en la direcció $80^\circ$ amb DM .....	73
Figura 72 Constel·lació rebuda en la direcció $120^\circ$ amb DM .....	73
Figura 73 Mòdul del senyal rebuda segons la direcció amb DM per seqüència de símbols. ....	74
Figura 74 Fase del senyal rebuda segons la direcció amb DM per seqüència de símbols. ....	75
Figura 75 Percentatge símbols rebuts correctament segons direcció d'emissió (Pes potència=1).....	77
Figura 76 Percentatge símbols rebuts correctament segons direcció d'emissió (Pes potència=2).....	78
Figura 77 Percentatge símbols rebuts correctament segons direcció d'emissió (Pes potència=0.25). ....	78
Figura 78 Constel·lació rebuda amb DM amb el SDR direcció $\Theta_0$ .....	80
Figura 79 Constel·lació rebuda amb DM amb el SDR direcció $30^\circ$ .....	80
Figura 80 Potència senyal rebut amb DM (pes potència soroll=0.25). ....	81
Figura 81 Fase senyal rebut amb DM (pes potència soroll=0.25). ....	82
Figura 82 Potència senyal rebut amb DM (pes potència soroll=1) .....	82
Figura 83 Potència senyal rebut amb DM (pes potència soroll=0). ....	83
Figura 84 Potència senyal rebut amb DM (pes potència soroll=0). ....	83
Figura 85 Fase senyal rebut amb DM (pes potència soroll=0).....	84
Figura 86 Percentatge símbols rebuts correctament amb DM i decisor centroides (pes potència soroll=1). ....	85
Figura 87 Percentatge símbols rebuts correctament amb DM i decisor SVM (pes potència soroll=1). ....	85
Figura 88 SER pel senyal amb DM i decisor centroides (pes potència soroll=1) .....	87
Figura 89 SER pel senyal amb DM i decisor SVM (pes potència soroll=1) .....	87

## *Índex de taules*

---

Taula 1 Comparativa sistemes SDR.....	12
Taula 2 Cadena de text rebuda i símbols rebuts correctament quan s'emet "cadena de text".....	75

# 1. Introducció

## 1.1. Estat de l'art

En els darrers anys s'han vist molts atacs que comprometen la seguretat dels individus, s'han buscat moltes formes de millorar la seguretat i fer que les transmissions de dades entre dos individus sigui confidencial, íntegre i coherent i per la resta d'usuaris sigui inintel·ligible.

S'han fet nombrosos estudis per crear esquemes que permetin l'enviament segur de dades, una de les formes més habituals és amb esquemes de criptografia, aquests esquemes es basen en la idea que el receptor pugui desencriptar a les dades però que l'atacant, que ho hagi de fer amb força bruta per saber la informació que s'ha enviat, li sigui computacional impossible desxifrar la informació.

També s'han buscat esquemes d'enviament que millorin la seguretat de l'enviament de dades, com per exemple és la modulació direccional amb la qual tractarem en aquest article que comporten una dificultat afegida per saber la informació que s'està enviant i complica als atacants vulnerar la privacitat del canal de comunicació.

Per la construcció i configuració d'esquemes de transmissió de dades són necessaris molts components de Hardware: amplificadors, mescladors, moduladors i demoduladors, filtres... La qual cosa també ens condiciona a tenir un sistema rígid per obtenir certes característiques i que té una major complexitat per la construcció del circuit. Amb l'aparició dels Software Defined Radio tots aquests components es poden configurar mitjançant Software.

L'increment de l'ús d'aquests sistemes ve relacionat també amb l'interès de crear Processadors amb Propòsit General (GPP) i Field Programmable Gate Arrays (FPGAs), i dispositius capaços de treballar amb els convertidors Analògic-Digital i inversa. Aquests dispositius han permès fer experiments per la comunicació i explorar noves aplicacions i hi ha nombrosos estudis amb l'interès d'utilitzar aquests sistemes per noves possibles aplicacions per la comunicació o adaptacions de sistemes ja existents.

Per exemple en l'estudi: [1] es plantegen reptes per l'ús de SDR per aplicacions SCA, per millorar la portabilitat, per aconseguir reptes computacionals, sigui per fer-los reconfigurables ràpidament o especificacions de temps real, aspectes de seguretat en la transmissió de dades, aspectes de regulació i certificació i altres aplicacions.

També els nombrosos estudis han portat a fer estudis per recopilar l'evolució d'aquests sistemes com es pot veure en l'article [2]. Exposa les diferents necessitats per les quals va néixer i per evolucionar i resoldre nous problemes que s'han plantejat.

Aquests sistemes ens ha permès plantejar-nos noves aplicacions i ha permès crear nous esquemes de transmissió de senyals i alhora fer-los més segurs que han tingut un gran impacte, sobretot en usos militars però també en les tecnologies del dia a dia. Més endavant, gràcies a la a aquests sistemes fàcilment configurables, veurem com crear un esquema de transmissió més segur amb la modulació direccional, la qual cosa permetrà als usuaris comunicar-se amb unes garanties de privacitat.

## 1.2. Objectius

L'interès principal d'aquest projecte és crear un model de modulació direccional i implementar-lo amb un SDR, per a aconseguir aquest objectiu el dividirem en diferents subobjectius:

- Hi haurà una part més teòrica del treball que consistirà a elaborar un esquema de modulació direccional, fent els càlculs i interpretacions per a poder enviar les dades de forma segura en una direcció concreta. [3]
- Per altra banda, hi haurà una part més pràctica que consistirà a adaptar a les noves eines de Matlab i Simulink algunes funcionalitats bàsiques de transmissió i recepció per sistemes SDR i testejar que funcionin correctament utilitzant les noves Toolboxes creades per la comunitat de Matlab. Aquestes funcionalitats bàsiques amb les quals em basaré es troben explicades en "Control and evaluation of the ad-fmcomms5-ebz software-defined radio" de Guillem Foreman Campins. [4]
- Finalment, hi haurà una part que integrarà els dos objectius anteriors per tal d'implementar la modulació direccional pel dispositiu AD-FMComms5-EBZ SDR amb Matlab i Simulink i testejar en la càmera anecoica el seu comportament.

Caldrà prèviament plantejar-se uns objectius previs necessaris per aconseguir els objectius principals, que són els següents:

- Caldrà llegir i entendre el projecte "Control and evaluation of the ad-fmcomms5-ebz software-defined radio", entendre les diferents dificultats que s'han presentat en el projecte mencionat, com s'han solventat. Amb aquest subobjectiu cal remarcar que, tot i que no només gràcies a la lectura d'aquest projecte, sinó que

caldrà contrastar o buscar més informació, s'assolirà algunes competències clau [4] :

- Entendre d'una forma bàsica els sistemes Software Defined Radio, que són, perquè es van crear, quins avantatges aporta...
- Entendre el funcionament del dispositiu ad-fmcomms5-ebz.
- Nocions bàsiques d'interacció entre el Matlab i Simulink amb el dispositiu amb el que es treballarà.
- Informar-se i entendre les noves eines disponibles de Matlab i Simulink per interactuar amb el Software Defined Radio.
- I abans d'implementar el model de modulació direccional caldrà entendre el seu funcionament i els elements necessaris per aconseguir-la.

### 1.3. Metodologia de treball

La metodologia de treball per la part pràctica es basarà a treballar amb el Software Defined Radio AD-FMComms5-EBZ i les proves pertinents al sistema es faran en una sala anecoica de la qual es disposa en la Universitat, amb ella podrem comprovar que les implementacions realitzades funcionin correctament, considerant que no hi ha interferències externes al nostre sistema.

Per a dur a terme aquest projecte es treballarà amb el software de Matlab/Simulink per interactuar amb el dispositiu AD-FMComms5-EBZ SDR d'Analog Devices per implementar les funcionalitats desitjades, així com les proves per entendre el funcionament o aconseguir les funcionalitats.

Per la part més pràctica del projecte, com ja s'han comentat, ens basarem en el projecte "Control and evaluation of the ad-fmcomms5-ebz software-defined radio" de Guillem Foreman Campins per a ampliar-lo i assolir els objectius marcats en l'apartat anterior [4]. Per la qual cosa, per començar el treball, el primer pas necessari serà fer una lectura i entendre el projecte mencionat, caldrà entendre que són els Software Defined Radios d'una forma bastant general, caldrà tindre nociions del funcionament del dispositiu amb el qual treballarem AD-FMComms5-EBZ SDR d'Analog Devices, la implementació amb Matlab i Simulink de les funcionalitats bàsiques implementades per la transmissió i recepció de senyal, les diferents incidències o problemàtiques amb què es va trobar el projecte anterior.

Per altra banda, també serà important familiaritzar-se amb les noves eines que ofereix Matlab i Simulank per a interactuar amb dispositius com el AD-FMComms5-EBZ amb el

que treballarem, com treballar amb aquestes noves eines, les variacions que introduceix respecte a l'anterior implementació utilitzada.

I un cop assolit aquest objectiu en la part pràctica i assolida la part que s'ha considerat més teòrica del treball, es farà la implementació de la modulació direccional teòrica que s'ha dissenyat i es comprovarà el seu funcionament en la cambra anecoica.

Per la part més teòrica del treball, inicialment s'entendrà el funcionament de la modulació direccional, que és, perquè és útil, com funciona... Posteriorment, s'entendrà els sistemes i matemàtiques necessàries per a la seva implementació i, finalment, es dissenyarà un esquema de modulació direccional.

Per completar el projecte, s'ha decidit treballar de la següent forma. Per la part pràctica, hi haurà una part que es podrà desenvolupar sense treballar des de la Universitat, ja que no és necessari treballar amb el dispositiu, sinó que més aviat es tracta d'entendre el funcionament del dispositiu i de les noves eines disponibles es podrà avançar remotament.

Per la part que és la familiarització amb les implementacions ja existents, adaptar a les noves eines i la implementació de l'esquema de modulació direccional es treballarà des de la Universitat.

Inicialment, em vaig marcar uns objectius pel que fa a la metodologia a seguir pel treball que consistia en la següent:

- Es treballarà un mínim d'un cop a la setmana a la Universitat amb el dispositiu amb la càmera anecoica per inicialment familiaritzar-se amb la implementació i iniciar les proves per adaptar a les noves eines. Es decidirà setmanalment, els cops que es treballarà depenen de la càrrega de treball disponible, que vindrà determinada pel progrés del treball, és a dir dependrà del cost temporal que pugui suposar adaptar a les noves eines i al progrés de les etapes teòriques o que es poden fer remotament i que són objectius precedents dels pràctics.
- Per la part realizable remotament, s'anirà avançant treballant setmanalment avançant en coneixements teòrics i explicant el que s'ha avançat en el laboratori. Es treballarà en paral·lelament en el progrés de la redacció del treball i els informes i amb les tasques assignades pel progrés del projecte durant aquest període.
- Per a assegurar el progrés del projecte es farà informes cada 10 dies del progrés, notificant de les tasques assolides, els resultats obtinguts i els diferents inconvenients que hagin sorgit durant la realització de les tasques i fent una planificació de les tasques a assolir pels següents 10 dies.

Finalment, aquesta metodologia es va seguir durant algunes setmanes, però a causa de alguns problemes de caràcter personal, de salut i elevada càrrega de treball de la carrera, a causa de l'altre treball de fi de grau d'informàtica. Es van fer alguns avanços posteriorment en els plaços previstos incialment i a l'estiu es va treballar de forma intensiva per finalitzar el treball.

## 1.4. Estructura del document

En primer lloc, en aquest document trobarem una introducció del contingut del projecte i que es pot trobar en aquest document. En ell es pot trobar un estat de l'art on s'explica de forma introductòria l'existència i estudis previs sobre els temes tractats en aquest projecte, la modulació direccional per crear un esquema d'enviament de dades segur i els sistemes SDR. Seguidament, es pot trobar els objectius marcats en l'inici del projecte, posteriorment la metodologia de treball seguida per assolir aquests objectius i l'estructura del projecte on s'explica el contingut del projecte, que és l'apartat actual.

En el capítol 2 parlarem dels components necessaris per a aquest estudi, parlarem dels SDR de forma genèrica, dels avantatges que tenen, els components que el formen i les seves aplicacions. Posteriorment de forma més explícita del SDR utilitzat en aquest estudi, el AD-FMComms5-EBZ. I al final del capítol explicarem la situació experimental, com s'han connectat els dispositius per analitzar i extreure les conclusions.

En el capítol 3 analitzarem les eines que utilitzarem al llarg d'aquest projecte, parlarem inicialment d'alguns conceptes previs importants pel projecte com són la modulació QPSK i el procés de calibració. Posteriorment, parlarem de les eines com a tal utilitzades, primerament de l'aplicació IIO Oscilloscope, parlant de les diferents pestanyes de configuració existents i com utilitzar-la. Finalment, informació sobre Matlab i Simulink, el treball realitzat amb Matlab previ analitzat sobre el que es basarà part del projecte i les particularitats d'aquest.

El projecte anterior s'ha adaptat a les noves eines de Matlab, parlarem de l'eina seleccionada per actualitzar-lo, com portar a cap la instal·lació i facilitats que ens ofereix aquesta nova eina en el capítol 4. També veurem com treballar amb aquestes noves eines, les funcions i objectes existents, alguns exemples d'ús d'aquestes eines i les particularitats d'aquesta nova eina.

En el capítol 5 veurem la modulació direccional, explicarem primerament amb què consisteix i com es pot crear, posteriorment hi ha unes simulacions teòriques del procés de modulació direccional i finalment el procés d'implementació amb el sistema SDR de la modulació direccional i l'estudi d'algunes qüestions.

Finalment, trobareu les conclusions i possibles línies futures de treball en el capítol 6 i en els annexos es pot trobar exemples de configuració, processament de dades i emissió i recepció de dades utilitzades al llarg del treball per interactuar amb e SDR o entendre el funcionament de la modulació direccional.

## 2. Plataforma experimental

### 2.1. Els Software Defined Radio

Els Software Defined Radio (SDR) són sistemes de comunicació on els components del sistema són implementats per software en ordinadors personals o sistemes integrats, també permeten implementar alguns processos com la modulació i demodulació. Aquests sistemes eliminan la necessitat d'amplificadors, mescladors, resistències i altres components hardware que anteriorment eren necessaris per a crear un sistema de telecomunicacions. [5]

Els SDR permeten implementar protocols de comunicació a partir de software. L'avenç de poder fer via software la modulació, demodulació, codificació i descodificació va simplificar molt els sistemes de telecomunicació, ja que anteriorment realitzar-les amb hardware resultava molt més complicat. Addicionalment, aquests processos i els components dels SDR poden ser modificats amb més facilitat per aconseguir noves propietats.

Aquests sistemes tenen molts avantatges respecte als sistemes no programables. La primera aparició dels Software Defined Radio data del 1984 per un equip de E-Systems, el concepte de digitalitzar els components que anteriorment estaven en hardware va suposar una gran millora en la simplicitat del sistema, a més aquests sistemes permeten poder testejar més ràpidament senyals i adaptar els components per obtenir el senyal desitjada. [6]

Avui en dia, els sistemes de processament de senyals han evolucionat de tal manera que la gran majoria de la funcionalitat de banda base s'implementa en programari i el maquinari de RF és prou reconfigurable perquè un front-end de ràdio gestioni la majoria dels sistemes de RF. La combinació de la flexibilitat de RF i la implementació del processament del senyal en programari ha conduït a la creació de ràdio definida per programari (SDR). [4]

Els SDR moderns s'utilitzen per a una àmplia gamma d'aplicacions com ara el monitoratge de l'espectre, les xarxes mmWave (5G), les comunicacions per satèl·lit (SatCom), la detecció i el rang d'automòbils i aplicacions de baixa latència. [7]

A continuació, en la figura 1 es pot veure un esquema de com són els sistemes SDR.

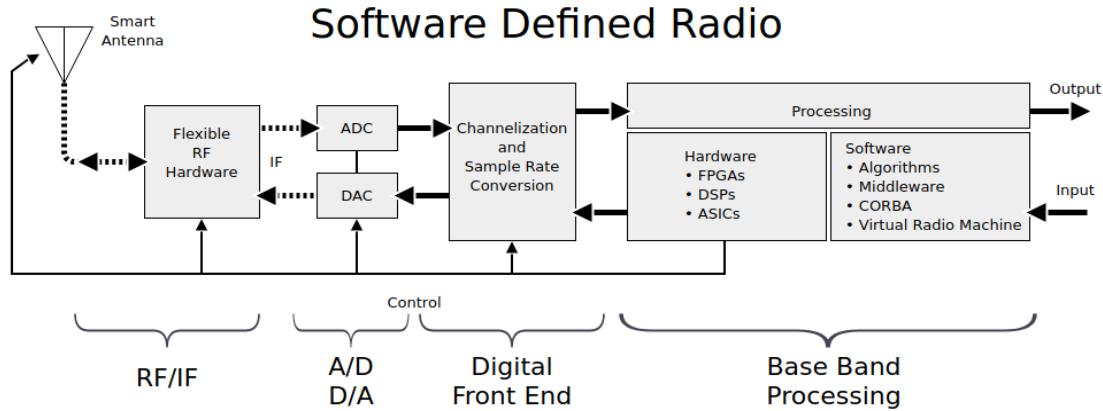


Figura 1. Sistema Software Defined Radio. [8]

Aquests sistemes són capaços de treballar a diferents freqüències, així com analitzar l'espectre per analitzar les transmissions que s'estan realitzant. La introducció dels SDR va suposar un gran avanç per les tecnologies IoT (Internet of Things) i altres camps on s'utilitza la comunicació sense fils fent que el processament i tractament de senyals fos més accessible. [8]

### 2.1.1. Components d'un sistema SDR

En la figura 2 podem observar l'arquitectura fonamental de un sistema Software Defined Radio:

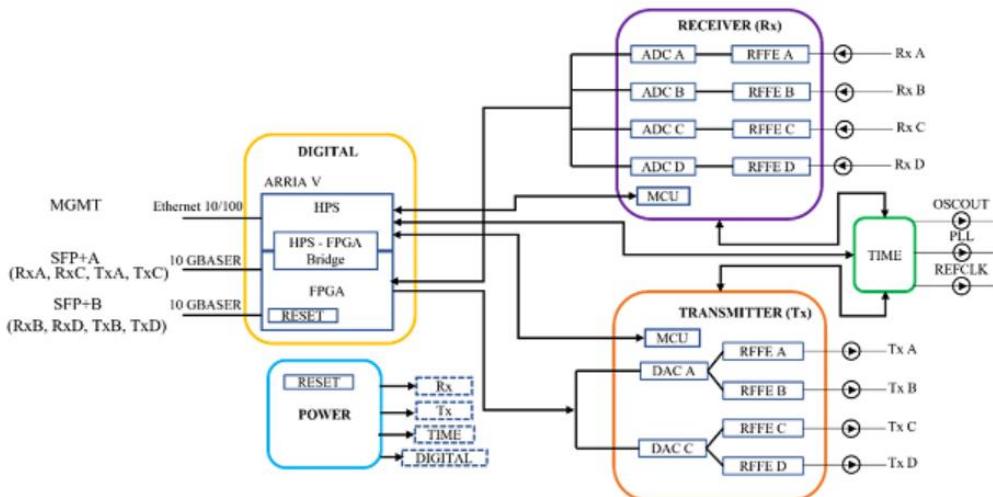


Figura 2. Arquitectura Fonamental SDR. [7]

Podem distingir en l'arquitectura dels sistema les següents parts fonamentals:

- **Digital:** s'encarrega de convertir a digital quan rebem un senyal i posteriorment s'envia al receptor. També s'encarrega del procés invers de rebre el senyal digital i convertir-lo en analògic per emetre'l a través del transmissor.
- **Power:** dona energia a les diferents parts del sistema.
- **Transmissor:** modela el comportament del transmissor.
- **Receptor:** modela el comportament del receptor.
- **Time:** controla el temps, on hi ha el rellotge del sistema, permet controlar que els diferents elements del sistema funcionin de forma sincronitzada.

Si analitzem amb més detall l'arquitectura podem observar alguns components que són molts importants pel funcionament del SDR [7] [9]:

- **ADC** (Analog-to-Digital Converter): és un dispositiu que mostreja un senyal continu i genera paraules de codi (bits digitals) amb una resolució que és igual al nombre de bits de l'ADC. El mostreig es fa a la freqüència de rellotge. Alguns paràmetres rellevants de l'ADC són: resolució (bits), freqüència de mostreig màxima, relació senyal-soroll (SNR), rang dinàmic sense espuris (SFDR), temps de serialització.
- **DAC** (Digital-to-Analog Converter): fa el procés invers a l'ADC, converteix els bits en un senyal continu i ens permet recuperar el senyal transmesa en el receptor.
- **Filtres digitals i analògics:** els filtres serveixen per separar bandes de freqüència, poden separar-se en baixes, mitjanes i altes. Els filtres serveixen per eliminar soroll o senyals no desitjades. Els filtres analògics poden eliminar tot el que està per sobre o sota una freqüència i un filtre digital pot ser programat per a ser més precís. Hi ha diferents tipus de filtre digitals:
  - FIR (Finite impulse response): Necessiten molta memòria i són més complexes, són més estables que els filtres IIR.
  - IIR (Infinite impulse response): Necessiten menys memòria però són menys complexes.

Els filtres també poden ser classificats en els següent tipus, on també podem veure una descripció gràfica en la Figura 3, per entendre la diferència entre aquest tipus de filtres:

- Filtres passa baix: Deixen passar freqüències baixes. (Figura 3a)
- Filtres passa alt: Deixen passar freqüències altes. (Figura 3b)

- Filtres passa banda: Deixen passar freqüències contingudes en un determinat rang de freqüències. (Figura 3c)
- Filtre elimina banda: Elimina el pas de freqüències contingudes en un determinat rang de freqüències. (Figura 3d)

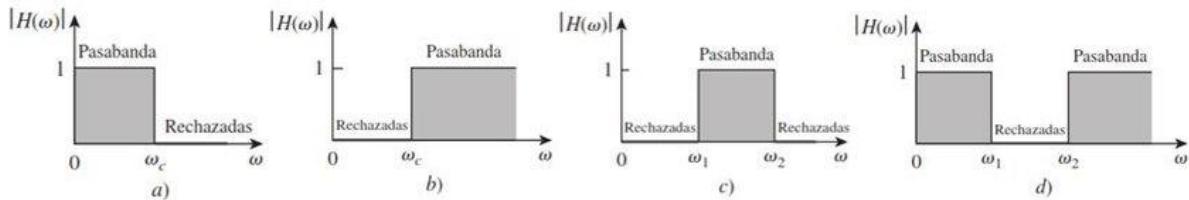


Figura 3. Tipus de filtres. [10]

- **Rellotge de cristall:** el rellotge genera un pols repetitiu i continu als diferents components del SDR, són importants degut a que la transmissió de senyals i modulació es fa utilitzant com a referència un rellotge, també altres funcionalitats com la sincronització, cronometrar... Ens interessa que el rellotge generi pulsos estables, precisos i amb valors baixos dels paràmetres de jitter i drift (que són paràmetres que indiquen la variabilitat dels temps entre diferents cicles del rellotge o respecte a un altre rellotge amb el mateix temps de cicle teòric) entre altres. Normalment, els rellotges utilitzats per als SDR són rellotges de cristall.
- **Amplificadors:** l'amplificador permet augmentar la potència del senyal quan la potència del senyal és molt baixa com per poder treballar amb ella en el sistema. Això pot donar-se en diferents situacions com per exemple: la potència és molt similar o inferior al soroll i no es pot apreciar, o bé, perquè algun component necessita operar en uns certs nivells mínims de potència o SNR.

Hi ha diferents tipus d'amplificadors com són: LNA (Low Noise Amplifier), amplificadors de potència, amplificadors de guany variable... que ens seran útils en diferents posicions del sistema, l'ordre en que es col·loquen els amplificadors i la resta de components pot tenir un gran impacte en la SNR.

- **Interfícies de Xarxa:** si volem obtenir un bon rendiment caldrà que les interfícies de xarxa del dispositiu es corresponguin amb el rendiment desitjat, normalment s'utilitzen fibres òptiques Ethernet amb port SFP+ al dispositiu. Caldrà analitzar que el rendiment de la interfície de xarxa en quan a latència, taxes de dades i altres paràmetres desitjats es correspongui.

- **Transformador:** proporciona i distribueix l'alimentació entre els diferents elements del circuit. Cal tenir en compte el rang de potències d'entrada i sortida, el ripple, la precisió, eficiència... També són necessàries les proteccions contra sobrecàrregues per protegir el sistema de danys en cas de fallades, com un curtcircuit.

## 2.1.2. Avantatges i inconvenients d'un SDR

Si intentem analitzar els diferents avantatges de inconvenients del SDR respecte els sistemes de hardware ens podem adonar de la importància i l'impacte que tenen i perquè avui en dia són tan utilitzats. Primer ens fixarem amb els avantatges que ens aporten aquest tipus de sistemes [7], [9], [11], [12] i [13]:

- Reconfigurables ràpidament: permeten que es puguin canviar components del sistema o configuracions del sistema molt ràpidament i amb gran facilitat via software.
- Actualitzables i millorables ràpidament: Es poden actualitzar amb nou programari a nous protocols o actualitzacions de components o altres ràpid i senzillament.
- Poden parlar i escoltar diferents canals alhora
- Es poden crear i implementar nous sistemes de radio
- Molt flexibles: la majoria poden adaptar-se a treballar en rangs de freqüència bastant grans. això permet utilitzar aquests dispositius ser utilitzats en tecnologies com LTE (Long Term Evolution).
- Sistemes robustos: Són sistemes fiables i efectius de telecomunicacions.
- Portabilitat: els seus pesos i dimensions són petits, per tant es fàcil traslladar de lloc i permet provar-lo en diferents àmbients.
- Comporten menors costos de desenvolupament, ja que es pot utilitzar el mateix prototip i posteriorment ja s'adaptarà a les necessitats específiques.

Seguidament ens fixarem amb els inconvenients:

- Obren a la possibilitat de nous tipus d'atacs, com són per exemple:
  - Atac de criptoanàlisi
  - Atac de canal lateral com Tempest
  - Atacs a la xarxa mòbil o mòbil
  - Anàlisi de comunicacions SAT
- Consum d'energia: poden consumir una gran quantitat d'energia.
- Gran quantitat de llibreries per utilitzar: pot complicar el trobar quina és la millor llibreria o l'adequada per una funcionalitat concreta. A vegades és difícil entendre com alguns paràmetres afecten el sistema o com funciona el sistema.

- ADC limita la freqüència màxima que es pot utilitzar amb sistemes SDR .
- El desenvolupament de SDR requereix enginyers de programari i maquinari.
- La plataforma SDR pot ser molt cara.
- Rang dinàmic petit en alguns dissenys SDR

### 2.1.3. Aspectes a considerar

Els aspectes més importants que cal considerar quan triem un Software Defined Radio són els següents [7], [8], [9]:

- Rang de freqüències en les que opera: ens interessa que aquest sigui gran, però serà prioritari que pugui treballar en un rang en les freqüències que ens interessen.
- Ampla de banda del receptor i transmissor: quan més gran més dades podrà recopilar però alhora també hi haurà més soroll.
- Resolució ADC: és important triar una resolució adequada, perquè a part de la resolució del senyal també serà important rebre a temps el senyal encara que es sacrifici una mica la resolució.
- Capacitats de transmissió i recepció: cal identificar quin tipus de dispositiu volem, es poden classificar en: semidúplex, full-duplex, etc.
- Cost: aquest és un factor bastant rellevant i que molt sovint ens limita, no és necessari un molt car però si un que satisfaci les nostres necessitats.

Hi ha diferents formes per dissenyar un sistema Software Defined Radio, n'hi ha de diferents tipus com són entre d'altres [9] [14]:

- GPP (General Purpose Processor)
- DSP (Digital Signal Processor)
- FPGA (Field Programmable Gate Array)

En la taula 1 podem veure una comparativa d'aquests diferents sistemes SDR:

Parameters	GPP	DSP	FPGA
Computation	Fixed Arithmetic engine	Fixed Arithmetic Engine	User Configurable Logic
Power Efficiency	Low	Moderate	High
Throughput	Low	Medium	High
Cost	Moderate	Low	Moderate
Input/Output	Dedicated ports	Dedicated ports	User Configurable Ports
Execution	Sequential	Partially parallel	Highly Parallel

Form Factor	Large	Medium	Small
-------------	-------	--------	-------

Taula 1. Comparativa sistemes SDR. [14]

### 2.1.4. Aplicacions

En aquesta secció parlarem d'algunes possibles aplicacions, ja que són usats en una gran quantitat d'ambients com ara [6], [7], [9], [14], [15], [16]:

- Propòsits de IoT.
- Connexions sense fils.
- Propòsits educacionals.
- Usos Militars.
- Usos mèdics.
- Descodificació de transmissions de veu digitals, com ara P25/DMR/D-STAR.
- Seguiment de posicions de vaixells marítims com un radar amb descodificació AIS.
- Radioastronomia.
- Seguiment i recepció de dades meteorològiques de l'agència meteorològica.
- Veure la televisió analògica.
- Altres.

## 2.2. AD-FMComms5-EBZ

En aquesta secció parlarem i introduirem el Software Defined Radio utilitzat per aquest projecte, en la figura 4 es pot observar el SDR utilitzat integrat a un ordinador:



Figura 4. AD-FMComms5-EBZ i Zynq ZC706.

El AD-FMComms5 Evaluation Board és una placa que permet fer qualsevol tipus d'aplicació de sistema SDR. Aquest dispositiu disposa de 4 transmissors i 4 receptors, es tracta d'una placa FMC pel AD9361, un transmissor RF Agile altament integrat, que a continuació s'explicarà amb més detall. El AD-FMCOMMS5-EBZ proveeix una plataforma que permet connectar i sincronitzar diferents AD9361 per aconseguir aplicacions de MISO, SIMO, SU-MIMO entre altres [3].

El receptor opera en freqüències d'entre 70 MHz fins a 6 GHz, mentre que el transmissor opera entre 47 MHz i 6 GHz i els canals permeten treballar entre freqüències menors a 200 kHz i 56 MHz. Aquest dispositiu disposa de 2 dispositius AD9361 on cada un controla dos receptors i dos transmissors.

### 2.2.1. AD9361

L'AD9361 És un dispositiu d'alt rendiment de radiofreqüència i altament integrat dissenyat per usos en aplicacions d'estació de base en 3G i 4G, que, com hem dit anteriorment, controla dos dels transmissors del dispositiu FM-Comms5 amb el qual es treballa en aquest projecte.

És un dispositiu programable i amb les seves capacitats d'amplada de banda per la transmissió el fan molt útil per aplicacions de transmissió. A continuació, veurem amb més detall algunes de les característiques seves.

Cada subsistema de recepció d'aquest dispositiu disposa, entre altres, de: control automàtic de guany (AGC), correcció de quadratura i filtrat digital, així com el dispositiu també compta amb modes guany manual flexibles i controlables externament. D'aquesta forma, els dos convertidors analògic-digital d'alt rang dinàmic reben el senyal en format de components en fase i en quadratura (I i Q respectivament) es digitalitza el senyal i passats per filtres configurables de delmat i filters FIR per produir un senyal de sortida de 12 bits a la taxa de mostreig adequada. [17] [18]

Els transmissors utilitzen arquitectura de conversió directa per aconseguir alta precisió en la modulació amb un soroll molt baix. Aquesta arquitectura del transmissor produeix una magnitud en el vector d'error (EVM) de fins a -40 dB, la qual cosa permet que hi hagi marge per seleccionar amplificador de potència. El soroll de fons del transmissor és menor a -157 dBm/Hz. [18]

A més aquest dispositiu permet mesurar la potència del senyal enviada amb gran precisió a partir d'un monitor de potència integrat que disposa, que té un rang dinàmic superior a 66 dB i amb una precisió d'1 dB. Els bucles de bloqueig de fase (PLL – Phase Locked Loop) totalment integrats proporcionen una síntesi de freqüència N fraccionada de baixa potència per a tots els canals de recepció i transmissió. L'aïllament del canal, exigit pels sistemes dúplex de divisió de freqüència (FDD), està integrat en el disseny i tots els components del filtre de bucle i VCO (Voltage Controlled Oscillator) estan integrats.

El nucli d'aquest dispositiu pot ser alimentat directament per un regulador de 1.3 V. L'IC es controla mitjançant un port sèrie estàndard de 4 fils i quatre pins de control d'entrada/sortida en temps real. S'inclouen modes complets d'apagada per minimitzar el consum d'energia durant l'ús normal.

El diagrama funcional del dispositiu és mostra en la figura 5, on es poden apreciar els diferents transmissors i receptors connectats als convertidors analògics-digital o digital-analògics, en el cas dels receptors reben les dades de la interfície de dades a partir d'ells i en el cas dels transmissors a la inversa i també es poden apreciar en l'arquitectura diferents controladors com el Clock.

El diagrama funcional del dispositiu és mostra en la figura 5, on es poden apreciar els components del dispositiu.

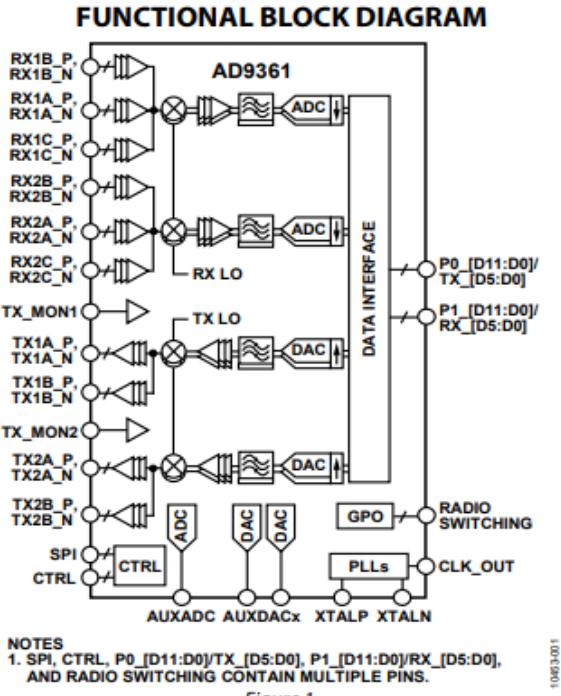


Figure 1.

Figura 5. Diagrama funcional. [18]

Algunes de les aplicacions més habituals pel dispositiu AD9361 són:

- Sistemes de comunicació punt a punt
- Estacions base Femtocell/picocell/microcell
- Sistemes de ràdio d'ús general

Es pot trobar més informació en el datasheet en el següent enllaç [18] i també es pot trobar més informació sobre l'ús en el manual de referència en el següent enllaç [19].

## 2.3. Muntatge experimental

Per entendre-ho ens fixarem en com són els experiments que estem fent, en la figura 6 podem veure una representació gràfica de la situació del nostre experiment:

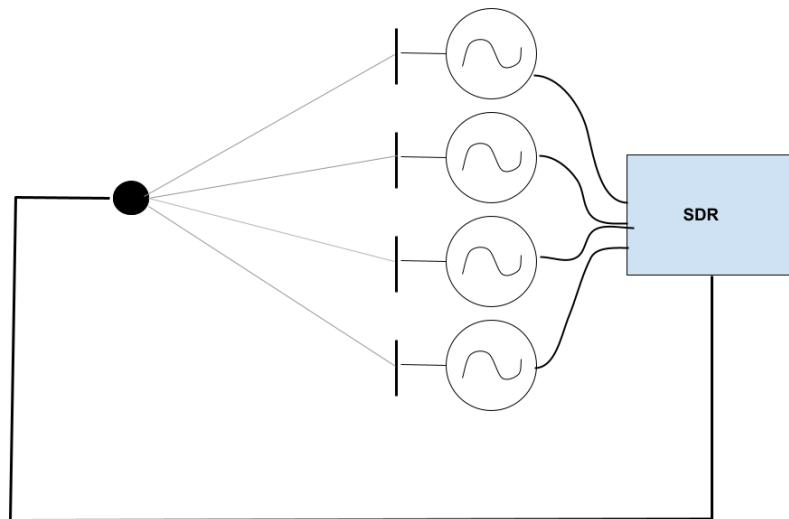


Figura 6. Representació dels experiments realitzats.

El nostre sistema SDR envia els senyals a través de 4 antenes transmissores, o 4 canals, amb el seu respectiu oscil·lador, que es poden observar a la figura 7. Aquestes 4 antenes transmetran el senyal per l'aire i aquesta serà captada per un receptor, el receptor tindrà una connexió amb cable fins al sistema SDR amb la qual enviarà les dades rebudes, el receptor es pot observar a la figura 8.

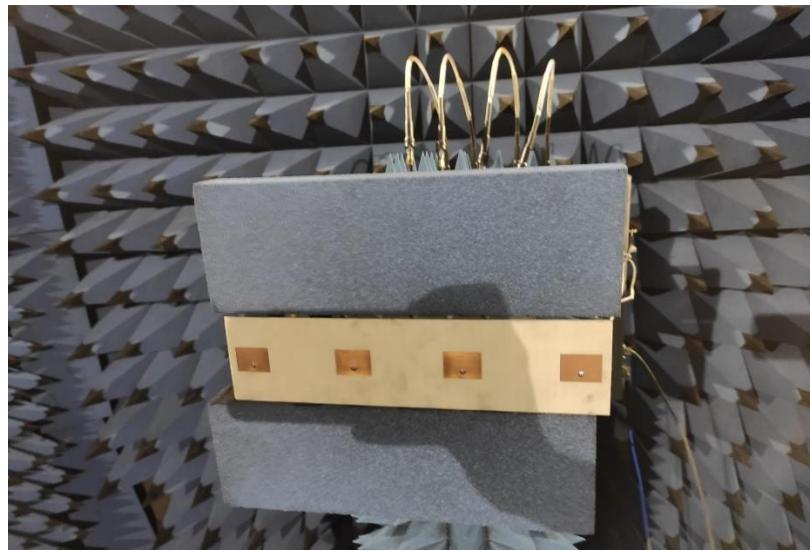
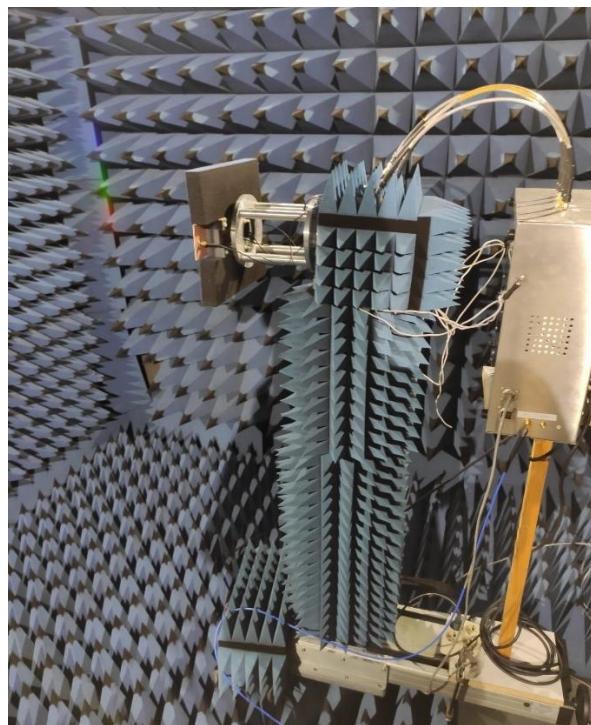


Figura 7. Transmissors del SDR.



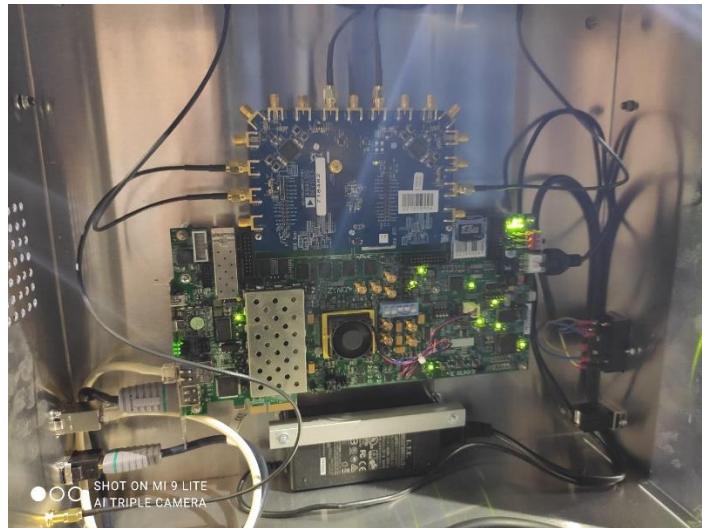
*Figura 8. Receptor connectat SDR.*

El sistema SDR i les antenes transmissores a les qual està connectat està situat sobre un controlador que permet controlar la direcció de les antenes, permeten canviar l'angle de recepció, que es pot observar en la figura 9.



*Figura 9. Transmissor SDR connectat al controlador.*

El sistema SDR es troba integrat a un ordinador de forma que ens hi podrem connectar via IP a través de Matlab i controlar els diferents experiments que vulguem realitzar, es pot observar a la figura 10. El sistema SDR s'ha definit perquè treballi a una freqüència central de 3.6 GHz.



*Figura 10. SDR integrat a un ordinador.*

Tot això es troba a l'interior d'una sala anecoica. Una cambra anecoica és una sala dissenyada per absorbir les refraccions i reflexions de les ones i aïllar el sistema de sorolls exteriors. Això permetrà que els experiments que es duguin a terme al seu interior simulin el comportament quan es transmet l'ona en un espai lliure sense interferències exteriors al mateix sistema.

### 3. Anàlisi de les eines existents

#### 3.1. Conceptes teòrics previs

##### 3.1.1. Modulació QPSK

La modulació per desplaçament de fase o PSK (Phase Shift Keying) consisteix a variar la fase entre un nombre determinat de valors discrets per poder enviar més informació pel canal per segon. En el cas de fer-ho amb 4 valors diferents QPSK (Quadriphase PSK) de forma que 2 bits codificant els símbols. Aquesta modulació digital es representa en el diagrama de constel·lació amb 4 punts equidistants de l'origen de coordenades amb una forma com la que es pot veure en la figura 11. [20]

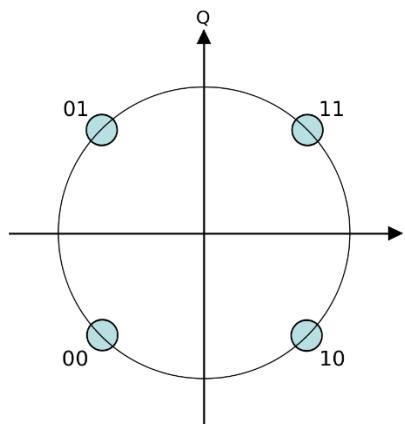


Figura 11. Constel·lació QPSK.

Això permet augmentar el nombre de bits per segon en el canal i d'aquesta manera la informació que es transmet pel canal per segon, però també provoca que el procés previ a l'emissió sigui més complex.

El senyal modulada en una PSK respon a l'expressió:

$$A_p \cdot \cos(2\pi ft + \theta) \quad (1)$$

On:

- $A_p$  es l'amplitud del símbol.
- $f$  és la freqüència del senyal.
- $t$  temps.

- $\theta$  representa els possibles valors de fase que pot tenir el senyal, seran tants com símbols possibles es puguin enviar.

D'una forma més particular per la QPSK, el senyal enviada respon a una equació com la següent:

$$S_n(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(2\pi f_c t + (2n - 1)\frac{\pi}{4}\right), \quad n \in \{1,2,3,4\} \quad (2)$$

### 3.1.2. Calibració

La calibració és un procés que compara els valors obtinguts per un instrument de medició amb un patró o uns valors de referència. En el SDR es compararan els valors transmesos pel canal amb els valors rebuts, concretament ens centraren amb les diferències de fases entre els senyals. D'aquesta forma es sabran les diferències de fase entre els diferents transmissors, el que permetrà fer processos com aconseguir interferències constructives, fer la modulació direccional o altres, i també permetrà mesurar la diferència de fase que hi ha durant l'emissió del senyal.

## 3.2. IIO Oscilloscope

La IIO Oscilloscope [21] es tracta d'una aplicació GUI multiplataforma que permet connectar-hi dispositius per avaluar el seu funcionament. Ens permet veure la seva sortida en diferents formats (temporal, freqüència, constel·lacions i correlació) i també permet modificar configuracions del dispositiu que volem avaluar. Per treballar amb un dispositiu és necessari especificar la seva IP.

### 3.2.1. Finestra de configuració

El IIO Oscilloscope ens ofereix una interfície gràfica més simple per entendre els paràmetres que podem modificar del FMComms5, a més de permetre enviar un senyal i posteriorment mostrar gràficament el senyal rebut. Els atributs configurables es poden agrupar en 4 categories: AD9361 Global Settings, AD9361 Receive Chain, AD9361 Transmit Chain i FPGA settings.

A continuació, passarem a analitzar les diferents parts configurables que ens ofereix el IIO Oscilloscope:

- En el cas de AD9361 Global Settings, se'ns presenta la interfície en la figura 12:

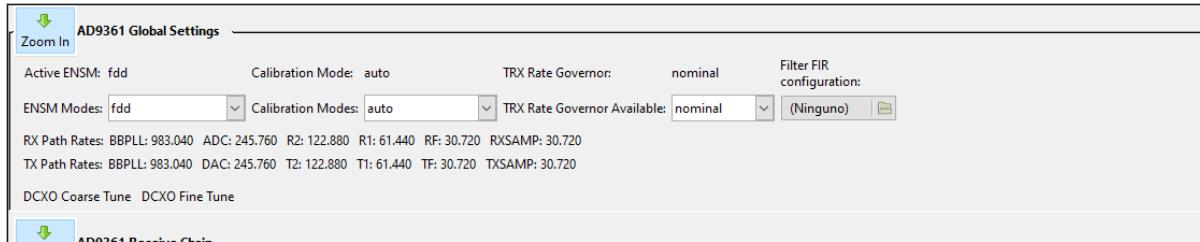


Figura 12. Configuració Global AD9361.

Els paràmetres configurables són:

- ENSM Modes (Mode ENSM): Es pot triar entre els modes FDD (Frequency Division Duplexing) i TDD (Time Division Duplexing).
- Calibration Mode (Mode de calibració): Permet triar entre el mode de calibració del dispositiu, normalment els algoritmes de calibració automàtics ja són molt bons.
- TRX Rate Governor Available: S'utilitza per configurar les tasses de decimació i interpolació dels filtres digitals en el camí de recepció i transmissió.
- Filter FIR configuration: (Configuració del filtre FIR): Permet utilitzar als usuaris filtres FIR.

Pel cas AD9361 Receive Chain la interfície és la que podem observar a la figura 13, els atributs que es poden seleccionar són:

- RF Bandwidth (Ample de banda RF): Configura l'ampla de banda de tota la cadena de recepció.
- Sampling rate (Tasa de mostreig): La màxima que permet aquest SDR és 30.72 MSPS.
- RF Port Select (Seleccionar port de RF): Les opcions disponibles en el xip són A, B i C, però en la placa FMCOMMS5-EBZ només està habilitada la opció A.
- RX LO Frequency (Freqüència LO del receptor): Configura la freqüència a la que treballa l'oscil·lador local del dispositiu.
- Hardware gain (Guany Hardware): Permet configurar el guany del receptor.
- Gain control mode (Mode control de Guany): Permet seleccionar com el AD9361 controla el guany del receptor. El mode "slow\_attack" controla el guany del receptor depenen del senyal rebuda. En el mode "manual" cal configurar el paràmetre Hardware Gain.

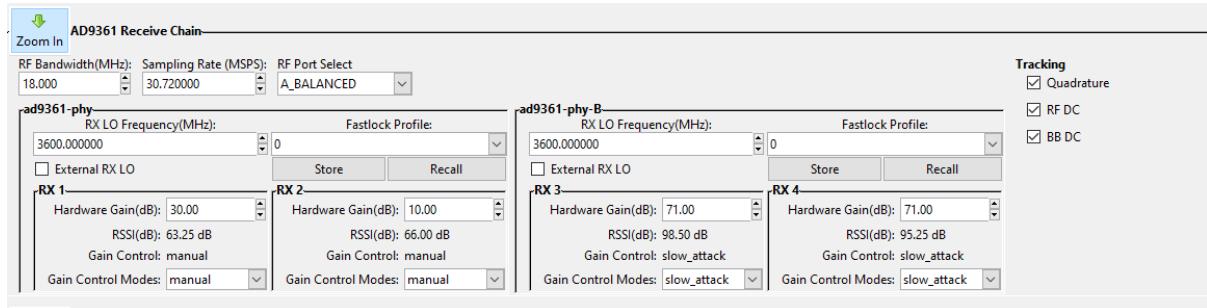


Figura 13. Configuració Receptor AD9361.

Pel cas AD9361 Transmit Chain la interfície és la que podem observar a la figura 14, és molt similar a l'anterior, els atributs que es poden seleccionar són:

- RF Bandwidth (Ample de banda RF): Configura l'ampla de banda de tota la cadena de transmissió, molt sovint és el mateix que en el receptor.
- Sampling rate (Tasa de mostreig): Similarment a l'anterior, és el mateix que en el receptor.
- RF Port Select (Seleccionar port de RF): Les opcions seleccionables són A i B.
- TX LO Frequency (Freqüència LO del transmissor): Configura una freqüència de l'oscil·lador local en transmissió.
- Hardware attenuation (Atenuació del Hardware): Permet configurar una atenuació per cada transmissor.
- Fastlock Profile (perfil Fastlock): Hi ha una opció per carregar un perfil de configuració emmagatzemat prèviament.

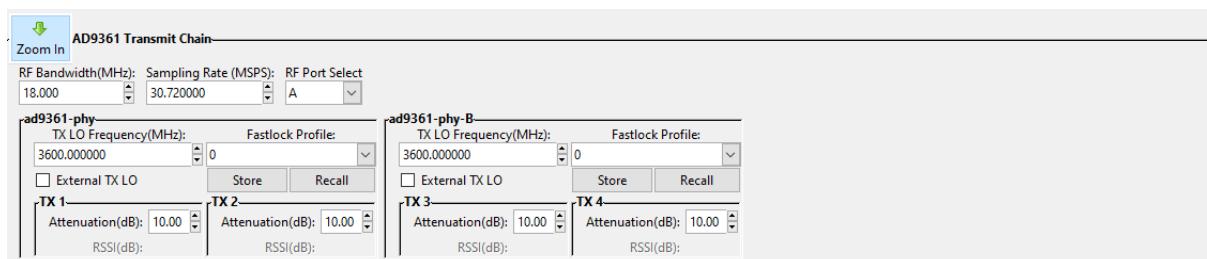


Figura 14. Configuració Transmissor AD9361.

Finalment, es troben els atributs pertanyents a la categoria FPGA Settings, aquests es poden separar en dos bàsicament Transmit/DDS (figura 15) i Receive (figura 16). Els atributs que trobem en el Transmit/DDS són els següents:

- DDS Mode (Mode DDS), permet triar el mode Direct Digital Synthesizer (DDS), que permet triar com es diferents tons:

- One CW Tone: Envia un to a través d'un transmissor seleccionat.
- Two CW Tones: Envia dos tons.
- Independent I/Q Control: Permet una configuració independent de dos o més tons a través dels canals pels components I i Q.
- DAC Buffer Output: Permet a l'usuari afegir un fitxer amb les mostres del senyal que desitja enviar. Aquestes dades es transmeten d'una forma cíclica fins que no es canvia la configuració per enviar tons o es carregui un altre fitxer amb mostres.
- Frequency (Freqüència): Permet triar la freqüència del to que s'enviarà, com es desplaçarà respecte TX LO Frequency.
- Scale (Escala): L'atenuació (quan es negatiu) que tindrà el senyal.
- Phase (Fase): La fase que afegeix al senyal.

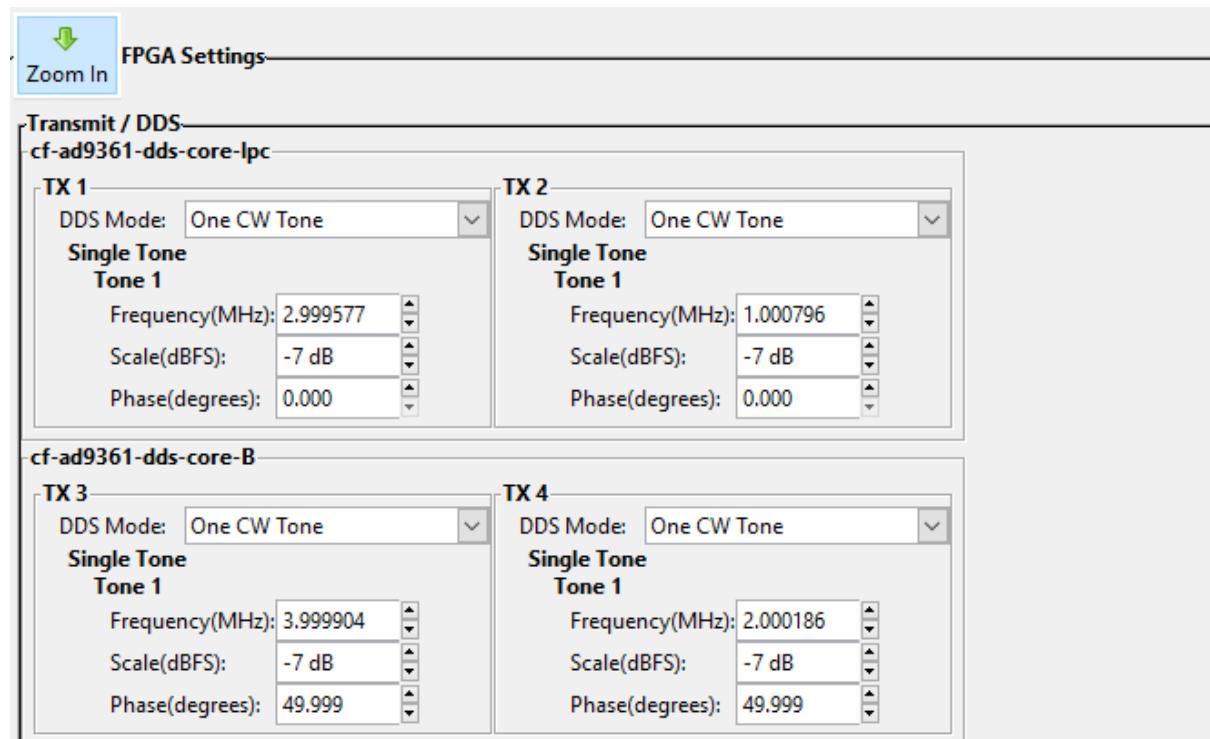


Figura 15. Configuració FPGA del transmissor AD9361.

En el cas del receptor tenim el paràmetre:

- Phase Rotation (Rotació de la fase): afegeix una fase al senyal rebut.

Cadascun d'aquests atributs són configurables per cada un dels transmissors i receptors independentment entre ells.

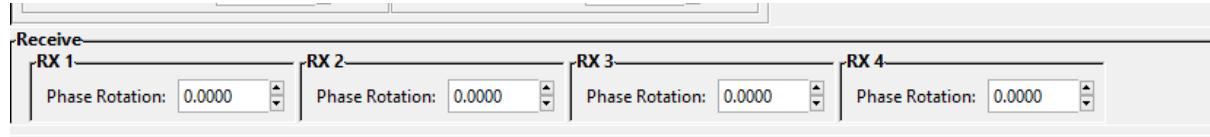


Figura 16. Configuració FPGA del receptor AD9361.

Per altra banda, també podem trobar una pestanya on es pot observar el diagrama de blocs del dispositiu AD9361, on es poden veure els components que formen el sistema (figura 17). Recordem que no totes les sortides disponibles del AD9361 es troben connectades a sortides de la placa FMCOMMS5-EBZ.

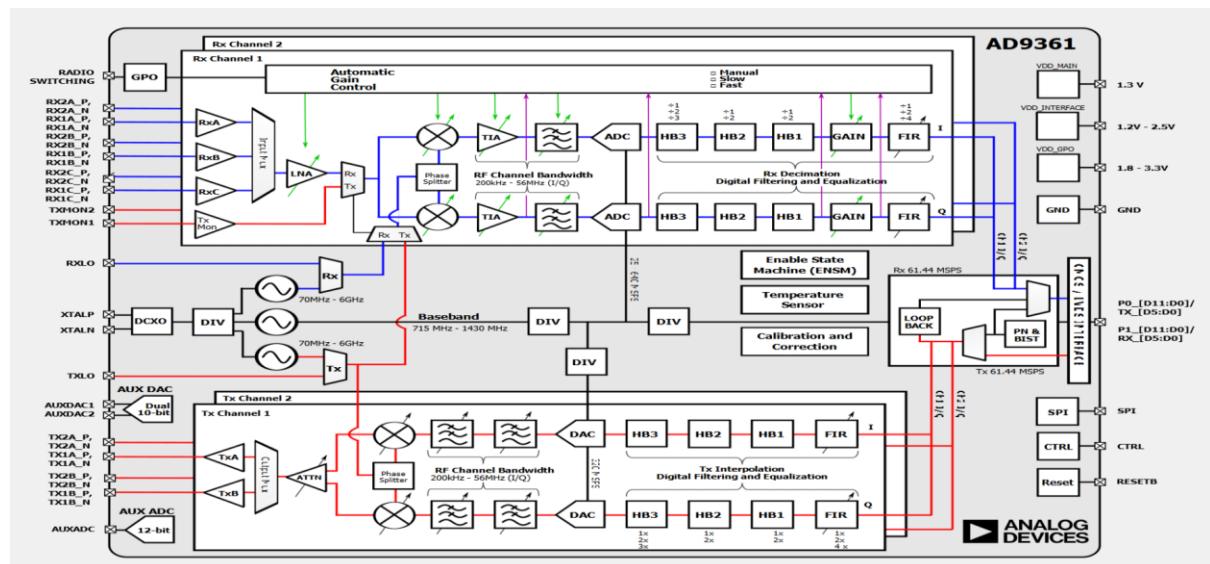


Figura 17. Diagrama de blocs del AD9361.

### 3.2.2. Finestra de representació de dades

La finestra de representació de dades és on es mostren les dades rebudes pel dispositiu. Es poden seleccionar diferents formes de mostrar les dades, cadascuna amb opcions per configurar:

- Domini del temps (figura 18):
  - Comptador de mostres: selecciona el número de mostres capturades que es representen en el domini del temps.
  - Tipus de graf: línies o punts.
- Domini de la freqüència (figura 19):
  - Mida FFT: selecciona el nombre de mostres que es fan servir en la FFT.

- Finestra: selecciona la finestra amb la que es processa el senyal abans de fer la FFT. Cada finestra té les seves característiques, que són més o menys adequades per una situació donada.
- Mitja FFT: selecciona la mitja de pesos per aplicar a les mostres de la FFT.
- PWR Offset: Selecciona el offset del graf FFT.
- Constel·lacions (figura 20):
  - Comptador de mostres: selecciona el número de mostres que es representen en forma de constel·lació (part real vs part imaginària).
  - Tipus de graf: línies o punts.
- Correlació Creuada (figura 21):
  - Comptador de mostres: selecciona el número de mostres que farà servir en la correlació creuada.
  - Tipus de graf: línies o punts.

També es poden seleccionar, com es pot veure en les figures 18 a 21 a dalt a l'esquerra, quines entrades de la placa es representaran (voltatge X). Aquestes 8 sortides corresponen als 4 receptors, per cada un d'ells es pot representar la component I (component en fase) i la component Q (component en quadratura).

En cadascuna de les figures, polsant el botó dret, es poden afegir marcadors que ens permetran un millor anàlisi de la sortida, com es pot veure en el cas de del domini freqüencial (figura 19).

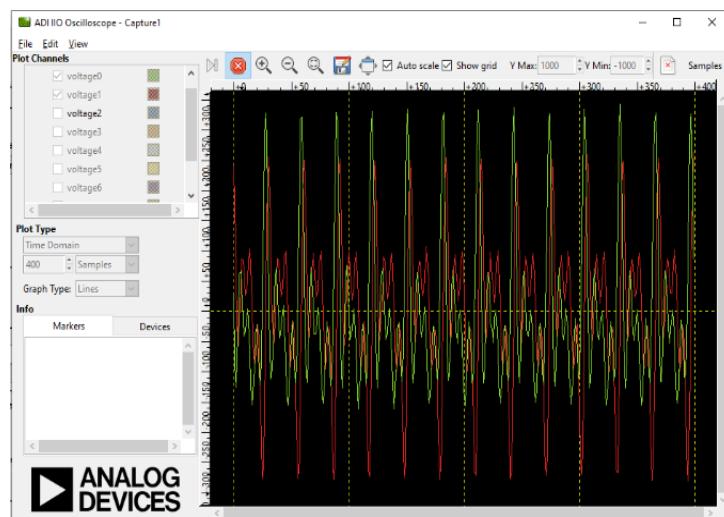


Figura 18. Representació domini temporal amb IIO Oscilloscope.

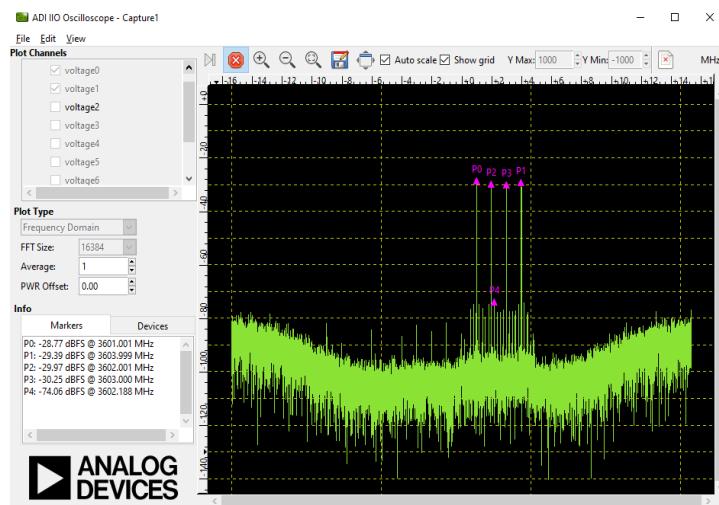


Figura 19. Representació domini freqüencial amb IIO Oscilloscope.

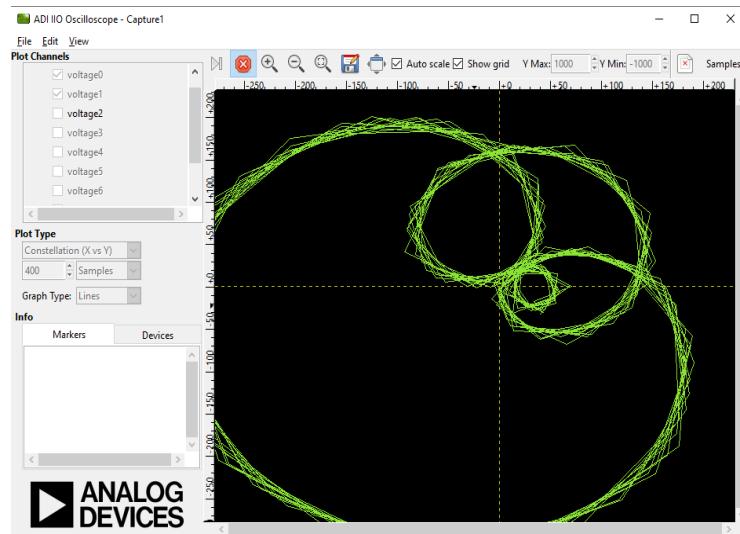


Figura 20. Representació constel·lacions amb IIO Oscilloscope.

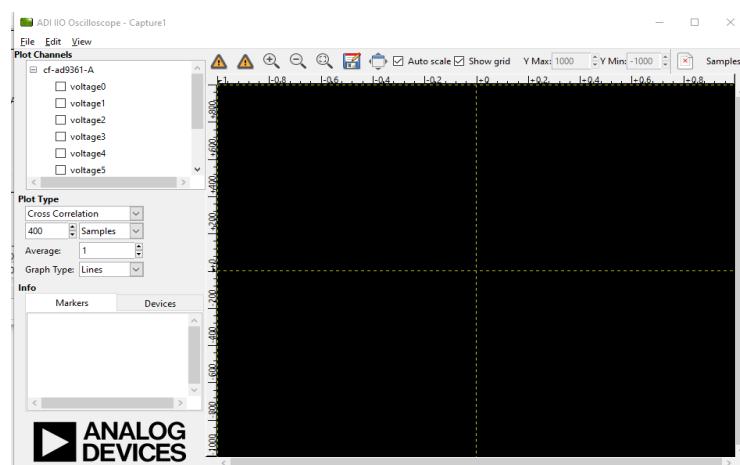


Figura 21. Representació correlació creuada amb IIO Oscilloscope.

### 3.2.3. Interacció amb el SDR

Per treballar amb el dispositiu AD-FMComms5-EBZ és important fer correctament la inicialització i per aconseguir-ho s'han de seguir els següents passos:

1. Engregar el dispositiu amb l'interruptor.
2. Posicionar el dispositiu com volem, en la posició i l'angle desitjat, això ho podem fer mitjançant el controlador de la figura 22.

(Nota vigilar amb la velocitat, la direcció i com es modifica la posició per evitar que els cables es pugui fer malbé).



Figura 22. Controlador que permet modificar l'angle del transmissor.

3. Iniciar l'aplicació IIO Oscilloscope.
4. Configurar en la pestanya FMComms5.
  - a. Transmissor i Receptors per treballar amb la mateixa freqüència (3600 MHz).
  - b. Per comprovar funcionament DDS Mode One CW Tone,
  - c. En l'apartat Transmit DDS per cada transmissor col·locar diferents Freqüències i una Scala (dBFS) que representarà l'atenuació amb la que es pugui apreciar (p.e: tots -7dB), d'aquesta forma podrem veure clarament al representar el senyal en freqüència les diferents senyals transmeses.

A les figures 23 i 24 podem veure un exemple de la configuració en el IIO Oscilloscope per comprovar el bon funcionament del dispositiu AD-FMComms5-EBZ.

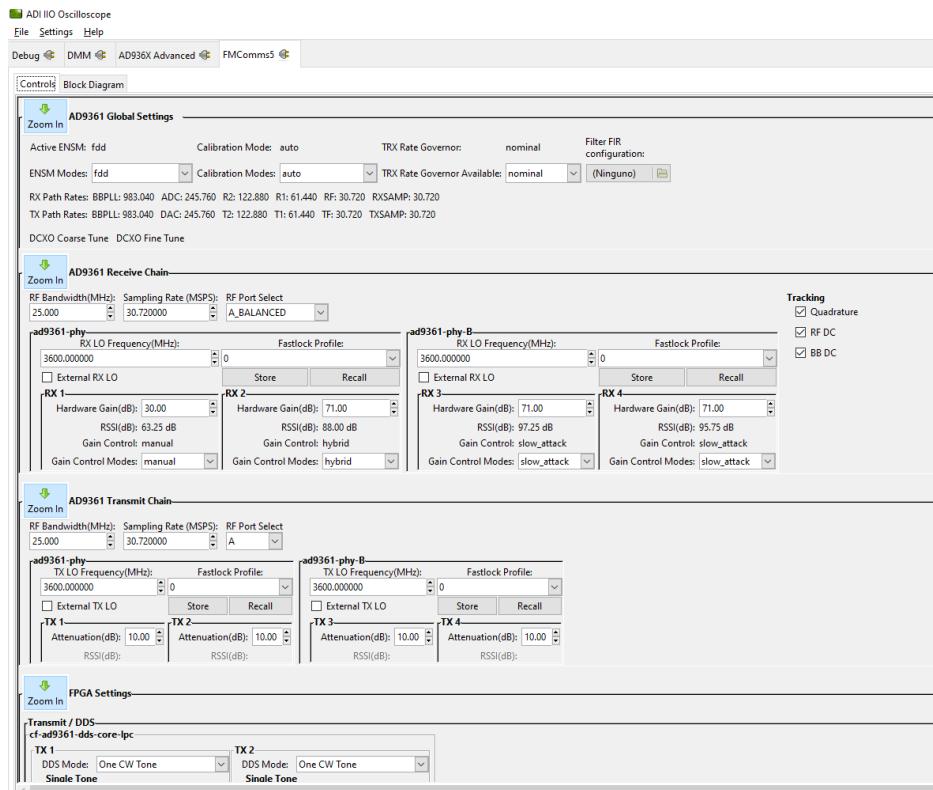


Figura 23. Configuració IIO Oscilloscope per encendre el SDR (I).

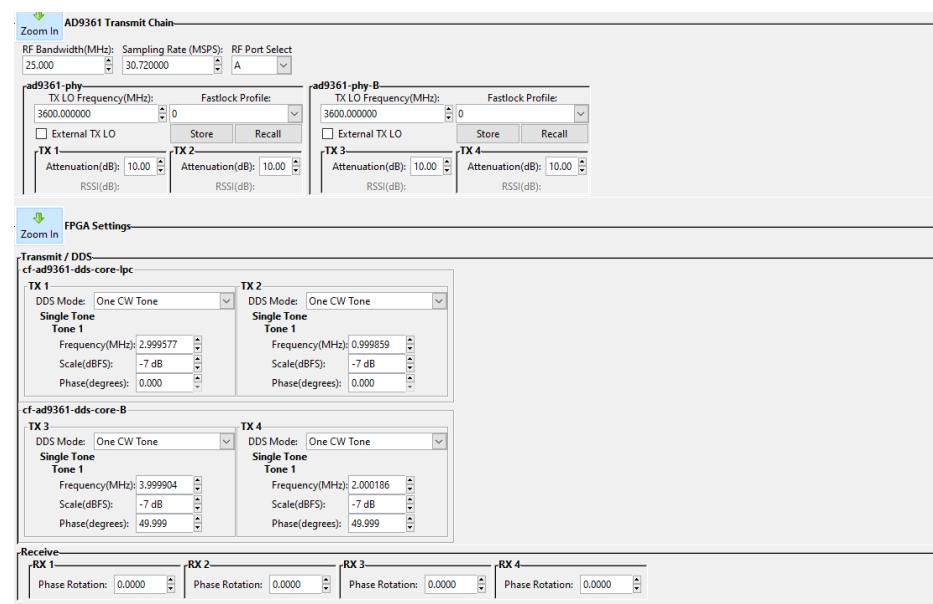


Figura 24. Configuració IIO Oscilloscope per encendre el SDR (II).

- En l'altra pestanya seleccionar Voltage 0 i 1 (correspondents als canals I i Q del primer receptor).

6. Comprovar que apareixen les 4 senyals correctament en el domini de la freqüència (es poden posar marcadors per comprovar la intensitat dels senyals).

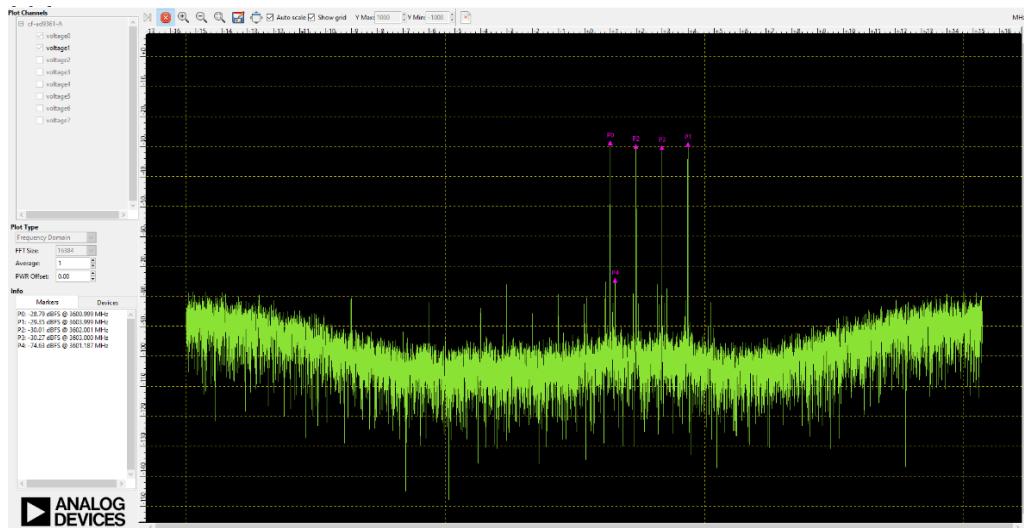


Figura 25. Comprovació domini freqüència encesa del dispositiu.

També es pot mostrar el senyal rebut en el domini del temps, però no es veu de forma tan clara el funcionament del dispositiu:

7. Apagar el IIO Oscilloscope, si es treballa amb el dispositiu fent servir dos interfícies diferents, es poden rebre ordres contradictòries i el dispositiu no funcionarà com esperem.
8. Ara ja podem obrir el Matlab i treballar des d'ell, a partir d'ell podrem crear implementacions per interactuar amb el dispositiu AD-FMComms5-EBZ.

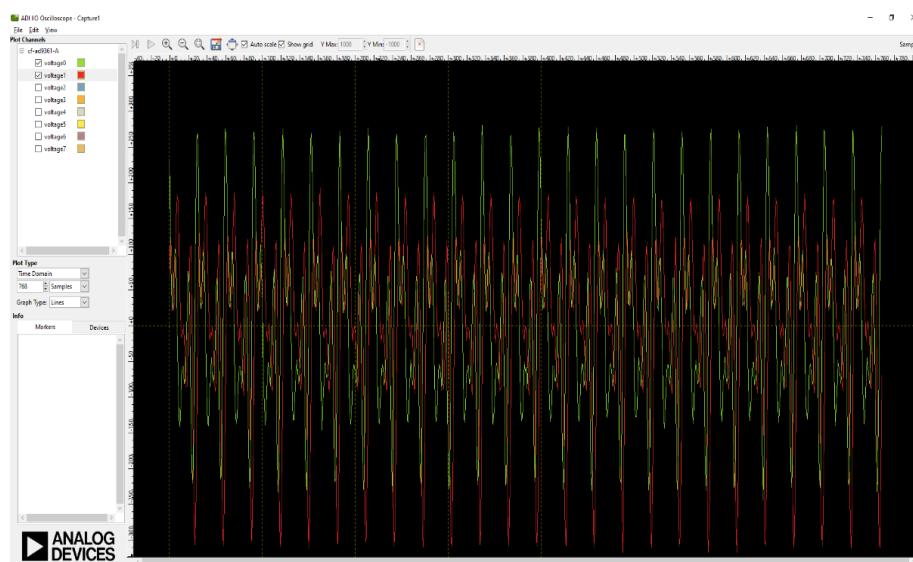


Figura 26. Comprovació domini temporal encesa del dispositiu.

## 3.3. Matlab i Simulink

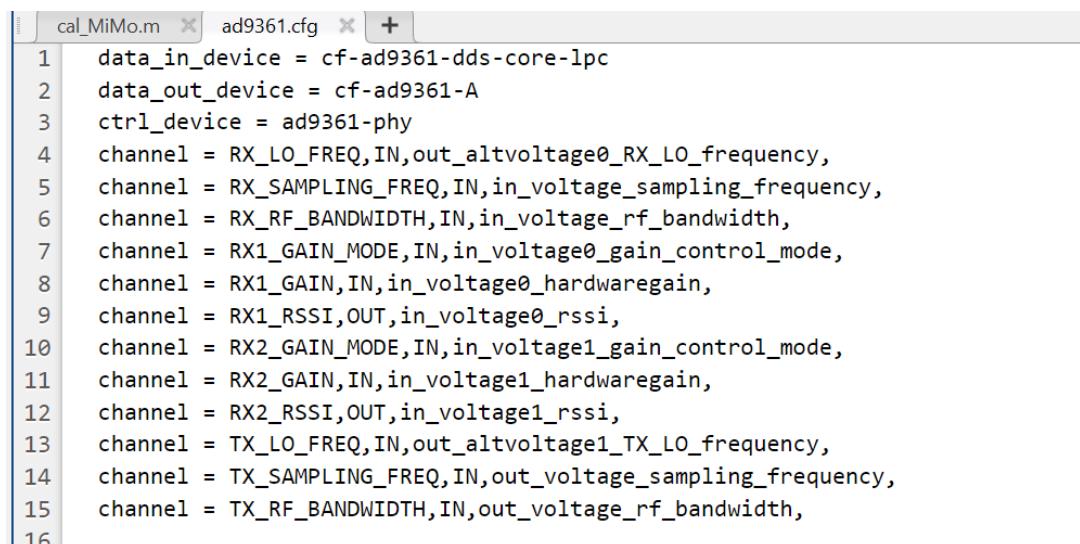
Al llarg del treball s'ha utilitzat el Matlab per els diferents processos, s'ha utilitzat el Matlab 2019b per interactuar amb el SDR. En el següent apartat parlarem del codi en el qual ens hem basat del treball d'en Guillem [4] per a poder interactuar amb el sistema SDR per la transmissió, recepció i pel procés de calibració.

### 3.3.1. Treball Existente

#### 3.3.1.1. Entorn i configuració prèvia

Si mirem la implementació ja existent sobre la qual es treballarà inicialment i servirà de guia inicial per entendre el funcionament de la interacció entre Matlab i el dispositiu AD-FMComms5-EBZ podem veure que es tracta de modulació QPSK, concretament aquestes primeres operacions serviran per a la calibració inicial dels diferents transmissors del dispositiu. Si mirem en detall la implementació podem veure algunes particularitats que s'han tingut en compte per la correcta transmissió i recepció de senyals.

Si revisem la implementació, veiem que es tracta d'un circuit Simulink on un dels blocs representa el dispositiu AD-FMComms5-EBZ, que funciona gràcies a les llibreries instal·lades: "iio\_sys\_obj.m", "iio\_sys\_obj\_matlab.m", "libiio\_if.m" i "libiio\_if\_daq2.m" i al fitxer de configuració del dispositiu anomenat "ad9361.cfg", que es mostra a la figura 27.



```
cal_MiMo.m ad9361.cfg +  
1 data_in_device = cf-ad9361-dds-core-lpc  
2 data_out_device = cf-ad9361-A  
3 ctrl_device = ad9361-phy  
4 channel = RX_LO_FREQ,IN,out_altvoltage0_RX_LO_frequency,  
5 channel = RX_SAMPLING_FREQ,IN,in_voltage_sampling_frequency,  
6 channel = RX_RF_BANDWIDTH,IN,in_voltage_rf_bandwidth,  
7 channel = RX1_GAIN_MODE,IN,in_voltage0_gain_control_mode,  
8 channel = RX1_GAIN,IN,in_voltage0_hardwaregain,  
9 channel = RX1_RSSI,OUT,in_voltage0_rssi,  
10 channel = RX2_GAIN_MODE,IN,in_voltage1_gain_control_mode,  
11 channel = RX2_GAIN,IN,in_voltage1_hardwaregain,  
12 channel = RX2_RSSI,OUT,in_voltage1_rssi,  
13 channel = TX_LO_FREQ,IN,out_altvoltage1_TX_LO_frequency,  
14 channel = TX_SAMPLING_FREQ,IN,out_voltage_sampling_frequency,  
15 channel = TX_RF_BANDWIDTH,IN,out_voltage_rf_bandwidth,  
16
```

Figura 27. Document configuració dispositiu (ad9361.cfg).

A la figura 28 podem observar el circuit Simulink:

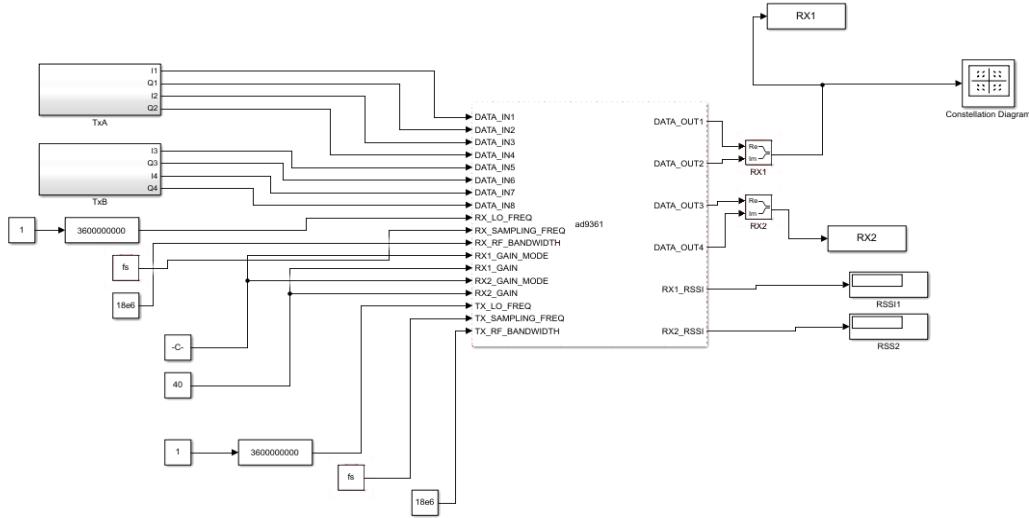


Figura 28. Circuit Simulink per emissió i recepció senyal.

En aquesta figura podem observar com es relacionen les entrades i sortides del model de Simulink amb els paràmetres de configuració del SDR.

Podem veure en la figura 28 com algunes de les entrades que s'han definit anteriorment en el fitxer de configuració (figura 27), són entrades del bloc que representa el dispositiu AD-9361. Principalment, veiem que hi ha les entrades de guanys dels receptors i transmissors, freqüències, ampla de banda i els senyals d'entrada.

Podem veure amb més detall els transmissors a la figura 29.

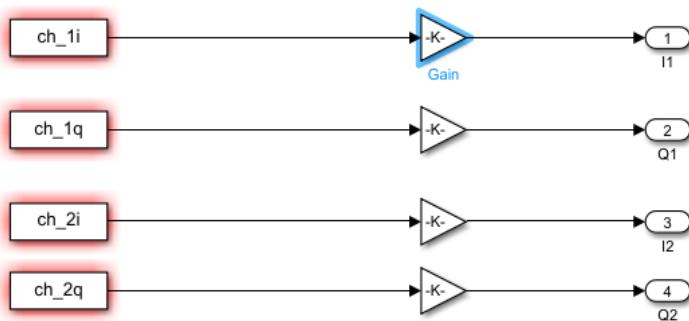


Figura 29. Esquema Simulink del senyal enviada en el transmissor.

Si ens fixem veiem que hi ha dos blocs d'entrada amb 4 entrades cada un, per tant hi ha un total de 8 entrades, que representen realment 4 senyals d'entrada ja que per cada senyal d'entrada es defineixen les components "i", que representa la component en fase, i "q", que representa la component en quadratura.

També s'ha vist que amb aquesta configuració, per obtenir resultats apreciables cal definir un guany inicial de  $2^{14}$ , la qual cosa sorprèn ja que és molt elevat i tampoc es sap el motiu pel qual ha de ser tan elevat per poder obtenir senyals apreciables.

### 3.3.1.2. Anàlisi del codi existent

L'objectiu del codi existent anteriorment, que servirà per entendre les bases per la interacció entre el Matlab i el dispositiu, era realitzar un sistema MIMO (Multiple Input Multiple Output) amb una modulació QPSK. En els sistemes amb aplicacions MIMO entre altres permet ser molt eficients en la transmissió, un exemple pot ser que per cada un dels transmissors s'envii el mateix senyal, tenint en compte les diferències de fase entre ells, d'aquesta forma, a causa de les interferències constructives, el receptor rebrà el mateix senyal amplificat, és a dir amb una major amplitud.

Per treballar en aplicacions en MIMO i DM és important tenir en compte les fases que alimenten les antenes, per poder fer això cal caracteritzar abans amb quina fase comença cada transmissor. Per fer això, el primer pas és sincronitzar els diferents transmissors i per tant fer un procés de calibració del sistema.

El codi sobre el que treballarem, que es pot trobar en l'annex 1, farà aquesta funció de calcular les diferències de fase entre els diferents transmissors quan s'envia un senyal amb modulació QPSK, per poder posteriorment fer una calibració del sistema. Per a fer això el codi funciona de la següent manera:

1. Definim diferents variables com són: freqüència de mostreig, període mostreig, número de frames que s'envien, numero de símbols per cada frame, número de bits per cada símbol...
2. Definim els diferents símbols possibles que es poden enviar.
3. Definim els frames que s'enviaran, així com els instants de temps en que s'enviaran.
4. Inicialitzem els vectors que contindran les dades que s'enviaran pels transmissors (Tx A i Tx B en la figura 28), guardem cada dada enviada en dos vectors, un per la component "i" i l'altre per la component "q".
5. Inicialitzem un vector on es guarda el que es va enviant, tenir en compte que per distingir entre frames cal algun codi que ens avisa, en el nostre cas s'ha triat quan s'enviïn dos zeros seguits que representa l'inici d'un nou frame.

Posteriorment per cada canal que voldrem testejar es seguirà el següents passos:

1. Creem i inicialitzem les seqüències de dades enviar, en aquest cas com aquest procés es per calibrar el sistema, s'enviaran símbols de forma aleatori, els dos primers símbols enviats seran 0, que indicaran l'inici d'un nou frame. Posteriorment es compararà la cadena enviada amb la rebuda per buscar la distància que hi ha entre les mateixes mostres. Per aquest exercici en tots els canals s'ha enviat la mateixa seqüència, d'aquesta forma no caldrà generar nous símbols innecessàriament. Posteriorment s'allargaran els polsos per evitar distorsió, d'aquesta forma es podrà detectar amb les facilitat el símbol enviat.
2. Es transmeten 15 frames generats pel canal que volem testejar a través del transmissor del dispositiu SDR.
3. Es convertirà el senyal que volem enviar en format Timeseries, que és el format amb que el dispositiu SDR pot treballar amb Matlab.
4. S'envien les dades des del transmissor fins al receptor amb el bloc de Simulink, de forma que s'enviaran les dades pel dispositiu SDR.
5. Es recopilaran les dades rebudes pel receptor del dispositiu en format Timeseries.
6. Es compararà les dades rebudes amb les dades transmeses per obtenir la diferència de fase entre ells i així també identificar la diferència de fase per cada transmissor.

Per calcular la diferència de fase s'utilitza el següent mètode:

1. Posar un 0 als dos primers símbols.
2. Definim els símbols final i inicials amb un zero i per cada frame enviat:
  - a. El primer frame enviat correspon amb el 4rt rebut, guardo els valors de sortida del receptor 1.
  - b. Faig una convolució circular i busco el punt on sigui mínim que representarà la posició d'inici de cada frame.
  - c. Desplaço circularment per tenir l'inici al principi.
  - d. Llegeixo el punt mig, per evitar la distorsió.
3. Finalment, sabent la diferencia es pot calcular la diferència en fase.

### 3.3.2. Particularitats

Hi ha alguns conceptes que en aquest programa que poden cridar l'atenció i a continuació es mencionen i si és possible s'intentarà explicar el motiu de la seva importància.

### 3.3.2.1. Diferència de 4 frames

Com hem comentat anteriorment, a vegades és difícil treballar amb els Software Defined Radio degut a la gran quantitat de llibreries existents i que a vegades no es pot saber exactament el que està succeint, aquest aspecte és un exemple d'ells. Per algun motiu desconegut hi ha una diferència de 4 frames entre el frame enviat i el rebut, és a dir el 1r frame enviat es correspon amb el 5è frame rebut. Un altre aspecte que no es pot explicar, ja que és complicat d'entendre degut a les llibreries és el guany en el transmissor en la construcció del circuit Simulink.

La diferència de 4 frames es dona en part degut a la situació de l'experiment que hem vist anteriorment, ja que hi ha un retard entre que s'envia el senyal pel receptor del SDR i es processa en el receptor, la distància fins l'antena receptora i posteriorment de l'antena fins al Software Defined Radio, cal tenir en compte també els temps de processament dels diferents components del sistema. Tot i així, el retard de 4 frames no s'explica del tot, ja que és un retard molt gran per la distància que hi ha entre els components en la realitat.

### 3.3.2.2. Lectura del punt mig per evitar distorsió

Quan s'envia un símbol pel canal, anteriorment s'ha allargat el senyal d'aquest símbol. Això permetrà evitar la distorsió, la distorsió es defineix com l'alteració de la forma de un senyal quan passa a través d'un sistema, en l'àmbit de la freqüència es donarà per alteracions en les freqüències, per exemple quan el senyal passa per un filtre, el senyal es veurà alterada degut a que no es poden construir filtres perfectes.

Generar polsos quadrats és impossible, ja que els senyals es poden presentar com combinació de sinus i cosinus, i per generar un quadrat perfecte caldria tenir un nombre infinit de combinacions. A més, quan més polinomis afegim més complex serà el sistema i més costos. És per això que s'envien senyals que s'aproximen a ell, però no tenen una certa pendent.

A la figura 30 tenim un exemple d'aquesta situació quan es vol generar un pols quadrangular (vermell), s'envia un senyal com la verda.

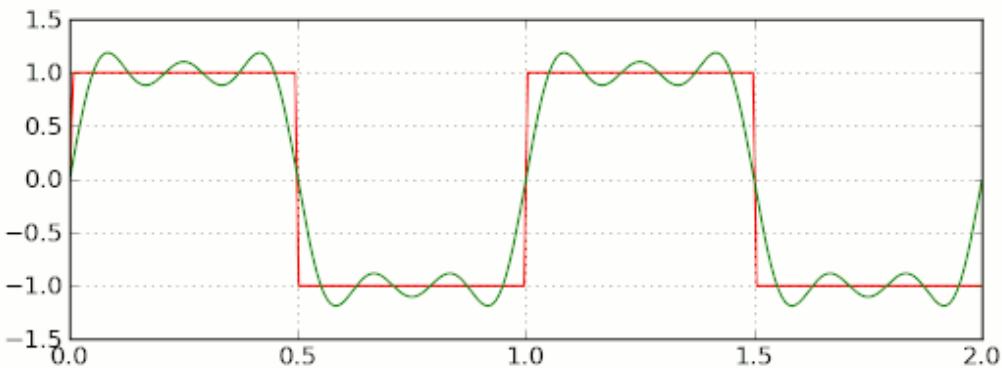


Figura 30. Polsos quadrats.

Per això, quan el senyal passi per exemple per un filtre, si comparem el senyal esperat amb el senyal que realment tenim tindrem una situació similar a la que es representa en la figura 31, on en vermell hi ha el senyal que obtenim i en negre el senyal esperat.

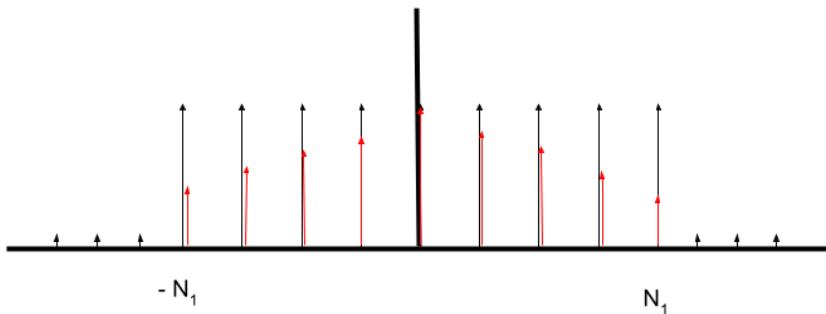


Figura 31. Senyal esperada i obtinguda al passar per un filtre.

Es pot observar que quan més el centre, l'efecte és menor, és per això que per evitar efectes de distorsió de filters i altres components, els símbols enviats pel canal es repeteixen i posteriorment es llegeix el punt mig, ja que és més estable que la resta i ens permetrà conservar i obtenir en el receptor a el senyal enviat pel transmissor amb menys distorsió.

### 3.3.2.3. Trama circular

Les trames rebudes estan rotades circularment, de forma que les primeres dades enviades en la recepció es poden trobar en un punt diferent. Això provoca que per identificar el símbol rebut corresponent a un símbol enviat sigui necessari ordenar les dades.

Per trobar l'inici els dos primers símbols enviat en cada frame es corresponen a dos zeros, això permet que en la recepció convolucionant circularment amb una cadena formada per 0s excepte en dues posicions corresponen als dos símbols 0 en els que es col·locaran 1s. D'aquesta forma al fer la convolució circular la posició que dona un valor mínim marcarà la posició d'inici de la seqüència en la recepció.

### 3.3.2.4. Fases d'inici dels oscil·ladors

Cada vegada s'encén el dispositiu SDR els oscil·ladors dels transmissors inicialitzen les seves fases de forma aleatòria. Això provoca que la diferència de fase entre el transmissor i el receptor sigui diferent cada vegada que s'inicialitza, aquesta s'afegirà a la variació fase introduïda per la distància entre l'emissor i el receptor.

En el projecte considerarem el vector de canal per la suma dels dos components, ja que mirarem la variació de fase entre unes dades enviades i unes rebudes les quals s'hauran vist afectades pels dos elements. És important ser conscient d'aquestes diferències de fase ja que afecten directament sobre el senyal rebut i depenen de l'aplicació, si transmetem amb més d'un transmissor a l'hora la voldrem corregir.

## 4. Actualització amb les noves eines de Matlab

Per a adaptar a les noves Toolboxes de Matlab, es va trobar un Add-on de Matlab que servia per interactuar amb AD-FMComms5-EBZ SDR, però calia instal·lar alguns requeriments previs. Per això és necessari actualitzar i instal·lar els diferents Add-ons requerits.

Per buscar la Toolbox adequada pel propòsit del treball, es van buscar Toolboxes que treballessin amb ad9361 i que tinguessin actualitzacions recents. Després es van explorar entre les possibilitats, observant alguns exemples així com tenint en compte les descàrregues i ressenyes d'altres usuaris i es va triar la Toolbox anomenada "Analog Devices Inc. Board Support Packages".

### 4.1. Analog Devices Inc. Board support Packages

En aquesta secció parlarem de l'add-on seleccionat per adaptar a les noves eines de Matlab, el Software Defined Radio que utilitzem, veurem com funciona i alguns exemples d'ús. L'add-on que hem seleccionat és el "Analog Devices Inc. Board Support Packages", que es pot observar a la figura 32.

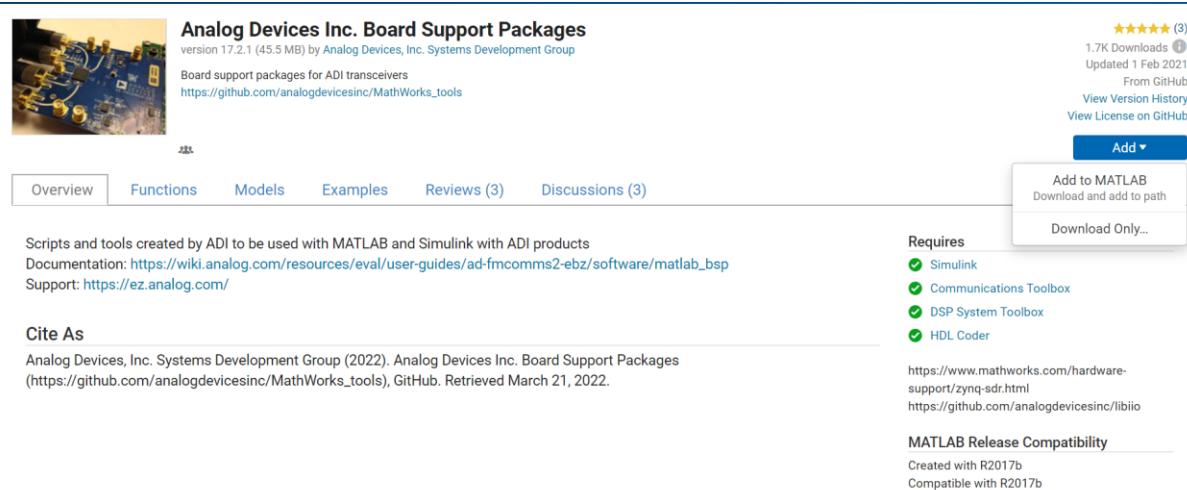


Figura 32. Add-on utilitzat per aplicar noves funcionalitats.

### 4.1.1. Instal·lació

Abans de instal·lar-la calia mirar entre els requeriments previs, es troava l'Add-On anomenat "Communications Toolbox Support Package for Xilinx Zynq-Based Radio", que realment ja teníem instal·lat però calia actualitzar i tornar a fer la configuració inicial. Bàsicament s'han d'anar seguint els passos que esmenta, els quals són bastants senzills amb l'explicació que dona el propi Matlab. A continuació, parlarem d'un possible error que ens pot sortir durant la configuració inicial.

Durant l'actualització un dels passos és actualitzar programari que hi havia en un targeta SD que es connecta amb el dispositiu, el qual quan el fem s'esborraran tots els arxius anteriors existents en la targeta SD, és per això que és recomanable fer una còpia de seguretat. Al fer l'actualització del contingut de la targeta SD col·locant la configuració correctament del dispositiu, aquest no funciona correctament. Això era degut a les configuracions de les interfícies i les IPs que provocava que perdés la connexió, que es trobava en una pantalla prèvia a l'escriptura en la targeta SD, es pot veure a la figura 33.

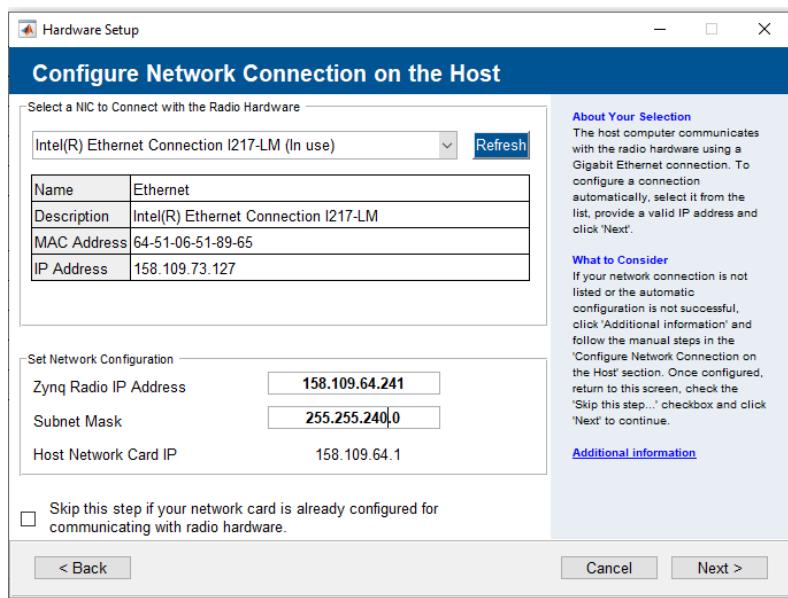
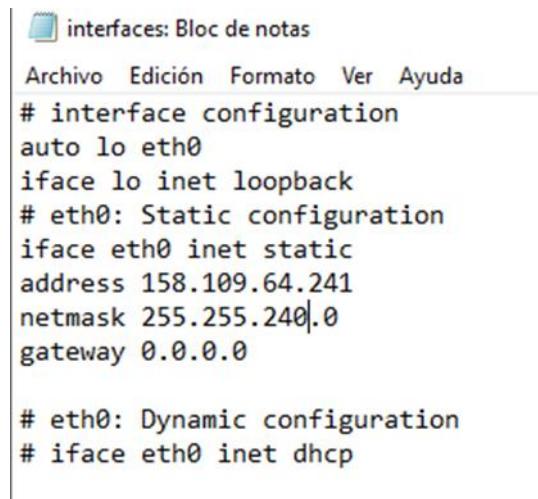


Figura 33. Configuració de la connexió de xarxa amb el dispositiu.

Tot i configurar correctament les IPs, aquestes no s'escriuen correctament en la targeta SD, a més pot ser que al fer aquest pas l'ordinador se'n disconnecti de la xarxa un moment. Per això, un cop seguits els passos de l'actualització va caldre reconnectar l'ordinador amb que configuràvem la targeta SD a la xarxa i modificar el fitxer "interfaces" de la targeta SD, que es trobava en l'emmagatzematge F: sobre la

configuració de BOOT, el qual no s'escrivia correctament. Calia configurar la màscara de xarxa i la IP del Gateway correctament d'acord al nostre entorn, el fitxer a modificar és el següent, i s'ha de modificar la netmask per la nostra màscara de xarxa en concret, en el nostre cas 255.255.240.0 i el Gateway s'ha modificat pel nostre Gateway que és 158.109.64.1.

Es pot observar el fitxer que s'ha de modificar d'acord a la descripció en la figura 34.



```
interfaces: Bloc de notas
Archivo Edición Formato Ver Ayuda
# interface configuration
auto lo eth0
iface lo inet loopback
# eth0: Static configuration
iface eth0 inet static
address 158.109.64.241
netmask 255.255.240.0
gateway 0.0.0.0

# eth0: Dynamic configuration
# iface eth0 inet dhcp
```

*Figura 34. Fitxer interfaces escrit a la targeta SD.*

Posteriorment es passarà a una pestanya per verificar la connexió. Si el procés de verificació funciona correctament podem veure que la verificació de la connexió amb el dispositiu que facilita Matlab es fa correctament i ens mostrarà una pantalla com la figura 35.

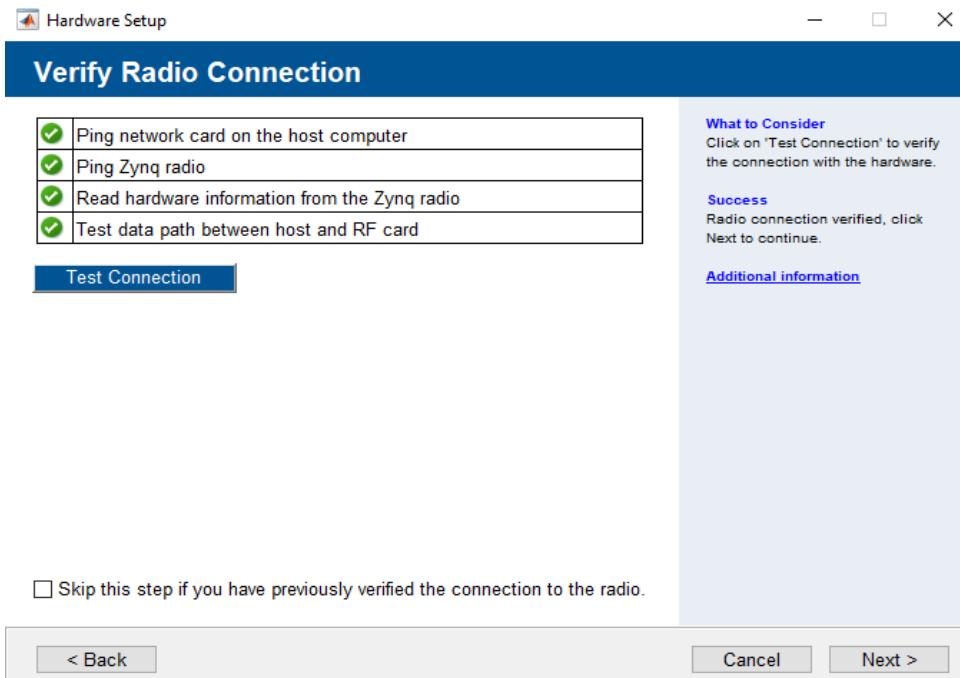


Figura 35. Pantalla de verificació de la connexió.

Posteriorment, si avancem es passarà a la pantalla de configuració pel codissey de HW o SW, que depenen del que vulguem fer amb el dispositiu amb aquesta Toolbox instal·lada caldrà instal·lar algunes altres eines de software de tercers per poder-les utilitzar. En el nostre cas, aquesta part la podem saltar.

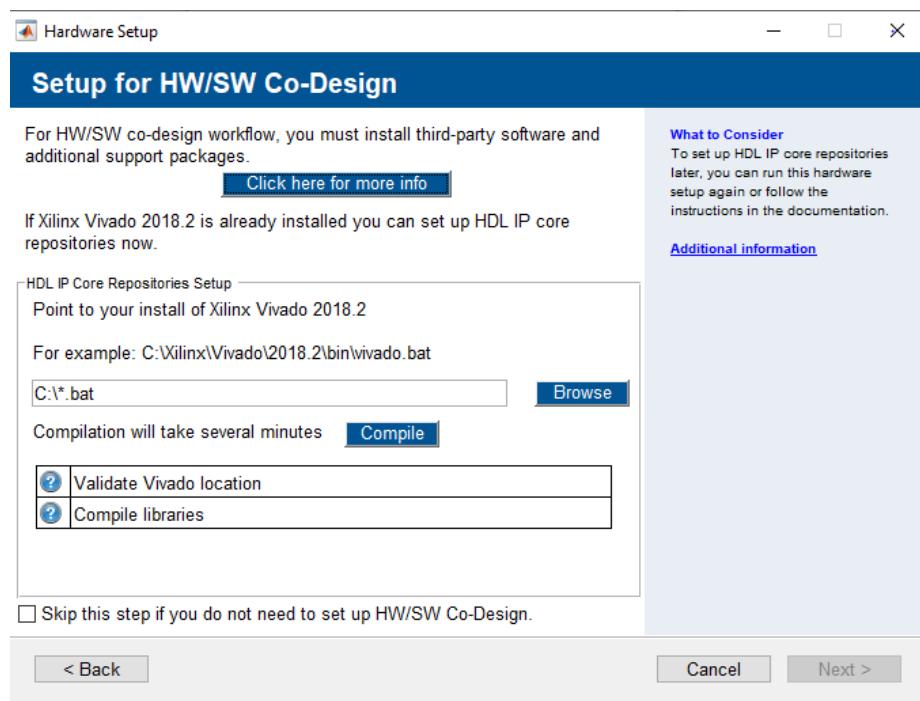


Figura 36. Pantalla de configuració pel codissey de HW o SW.

A la figura 37 es mostren els requeriments depenen del que vulguem fer d'acord a Matlab:

**Setup for HW/SW Co-Design**  
For the FPGA targeting and hardware-software co-design workflows, you must install additional MathWorks® support packages.

- To generate code for the FPGA portion of the development board, you must install HDL Coder™ Support Package for Xilinx Zynq Platform (requires HDL Coder).
- To generate code for the ARM® portion of the development board, you must install Embedded Coder® Support Package for Xilinx Zynq Platform (requires Embedded Coder, Simulink® Coder™, and Embedded Coder Support Package for ARM Cortex®-A Processors). In addition to these support packages, you must also install and configure this third-party tool:
  - Xilinx Vivado® development tools, version 2018.2If Xilinx Vivado is already installed on your system, you can set up HDL IP Core repositories in MATLAB by following the instructions on the screen.

You can perform this step later by following the instructions in [Installation for Hardware-Software Co-Design](#).

Figura 37. Requeriments per la configuració de codissey de HW/SW.

Un cop actualitzat i feta la configuració inicial de “Communications Toolbox Support Package for Xilinx Zynq-Based Radio”, ja es pot prosseguir a instal·lar l’Add-on “Analog Devices Inc. Board Support Packages” que ens servirà per actualitzar el programa a les noves Toolboxes. En el cas d’aquest Add-on, s’instal·larà correctament a la primera sense necessitat de passos especials i , en el nostre cas, no ens ha donat cap inconvenient.

A continuació es mostren els diferents Add-ons instal·lats, inclosos els dos esmentats anteriorment, a més d’alguns que es van instal·lar per en cas de que fossin necessaris que es poden trobar en la configuració per el codissey, o entre altres requeriments pels instal·lats.

	<b>Analog Devices Inc. Board Support Packages</b> version 17.2.1	Collection	Analog Devices, Inc. Systems Development Group	11 March 2022	
	<b>Embedded Coder Support Package for Xilinx Zynq Platform</b> version 19.2.5	 Hardware Support Package		11 March 2022	
	<b>Embedded Coder Support Package for ARM Cortex-A Processors</b> version 19.2.5	 Hardware Support Package		11 March 2022	
	<b>Embedded Coder Interface to QEMU Emulator</b> version 19.2.0	 Optional Feature		11 March 2022	
	<b>Simulink Coder</b> version 9.2	 MathWorks Product		11 March 2022	
	<b>Embedded Coder</b> version 7.3	 MathWorks Product		11 March 2022	
	<b>HDL Coder Support Package for Xilinx Zynq Platform</b> version 19.2.0	 Hardware Support Package		11 March 2022	
	<b>MATLAB Coder</b> version 4.3	 MathWorks Product		11 March 2022	
	<b>HDL Coder</b> version 3.15	 MathWorks Product		11 March 2022	
	<b>Fixed-Point Designer</b> version 6.4	 MathWorks Product		11 March 2022	
	<b>Communications Toolbox Support Package for Xilinx Zynq-Based Radio</b> version 19.2.3	 Hardware Support Package		8 March 2022	

Figura 38. Add-ons instal·lats o actualitzats.

## 4.1.2. Facilitats que ens ofereix la nova Toolbox

Per a entendre com treballar amb aquesta eina, es pot veure algun exemple de com interactuar amb el dispositiu per fer-nos una idea, però el que realment resulta útil per aprendre sobre com interactuar amb aquest sistema és buscar particularment una de les carpetes que instal·la aquest add-on anomenada "sdrz". A l'interior d'aquest directori hi ha un directori anomenat "sdrzdemos" on es poden veure molts exemples per interactuar amb el dispositiu, aquests permeten enviar i rebre tons, enviar i rebre modulacions QPSK...

A partir d'aquests documents podem fer una còpia del que s'adeqüi millor per tal de complir les nostres especificacions. D'aquesta forma podrem treballar sobre una base, sense modificar documents que puguin ser utilitzats en altres parts de l'add-on i sense perdre una base sobre la qual treballar en un futur. D'aquesta forma

Cal tenir en compte que en aquests fitxers caldrà modificar sempre els paràmetres de configuració per treballar amb FM-COMMS5 i especificar també la IP del SDR, això en el propi codi que estem modificant ja ve especificat on i com s'ha de modificar especificat com un comentari, a continuació en la figura 39, es pot veure com s'ha de modificar el fitxer.

---

```
%> Running the Example
% This example can be run by executing
% <matlab:edit('zynqRadioToneReceiverAD9361AD9364ML')>
% zynqRadioToneReceiverAD9361AD9364ML.m>.
if ~exist('prmToneRx','var')
    prmToneRx.SDRDeviceName = 'FMCOMMS5';
    prmToneRx.IPAddress = '158.109.64.241';
end
% To update the example for FMCOMMS5, set |prmToneRx.SDRDeviceName| to
% |'FMCOMMS5'|.
```

---

Figura 39. Fitxer de configuració d'exemple.

Aquesta configuració s'ha d'actualitzar tan en el transmissor com en el receptor del Software Defined Radio.

Una particularitat d'aquesta eina, és que quan s'emet un senyal aquest s'envia repetidament fins que s'executa un release() d'aquell sistema, en la recepció es capturen els fragments enviats que serà el que es mostrerà a l'usuari. Per tant, és molt

important fixar-se i tancar el transmissor a través d'un release al final de cada execució d'igual forma que cal parar també el receptor amb un release.

També accedint al menú del Matlab i buscant al cercador sdrx (en cas del receptor) i sdrtx es pot trobar més informació sobre les funcions que es poden utilitzar i entendre millor el codi ja existent per transmetre o rebrà i processar senyals.

## 4.2. Descripció de les noves funcions

### 4.2.1. Objectes, estructures de dades per configurar el SDR, funcions principals

Per tal d'enviar dades amb la nova llibreria prèviament cal configurar dues estructures que especifiquen les característiques del receptor i el transmissor. L'estructura per configurar el transmissor es pot observar la figura 40 i l'estructura per configurar el receptor es pot observar a la Figura 41.

```
sdrTransmitter = sdrtx(prmToneRx.SDRDeviceName, ...
    'IPAddress',      prmToneRx.IPAddress, ...
    'CenterFrequency', CenterFrequency, ...
    'BasebandSampleRate', RadioBasebandRate, ...
    'CenterFrequency', CenterFrequency, ...
    'ChannelMapping', [1 2 3 4], ...
    'ShowAdvancedProperties', true);
```

Figura 40. Configuració estructura transmissor.

```
sdrReceiver = sdrrx(prmToneRx.SDRDeviceName, ...
    'IPAddress',      prmToneRx.IPAddress, ...
    'CenterFrequency', CenterFrequency, ...
    'BasebandSampleRate', RadioBasebandRate, ...
    'GainSource', 'Manual', ...
    'Gain', 32, ...
    'SamplesPerFrame', RadioFrameLength, ...
    'ChannelMapping', 1, ...
    'OutputDataType', 'double');
```

Figura 41. Configuració estructura receptor.

En les figures 40 i 41 es pot veure la configuració que s'ha utilitzat per tal de transmetre modulació direccional amb el transmissor i receptor, els dos comparteixen alguns atributs:

- IP Address: adreça IP del dispositiu SDR.
- CenterFrequency: la freqüència central (freqüència LO) utilitzada, en el nostre cas la configurem a 3.6 GHz.
- Channel Mapping: Els canals del sistema que estem configurant (receptor o transmissor) que s'utilitzaran.
- BasebandSampleRate: Ampla de banda que donem al sistema, inicialment estava configurat a 30.72 MHz però posteriorment el modificarem a 15.36 MHz.

Pel cas del receptor s'afegeixen alguns paràmetres més:

- GainSource: es pot configurar el tipus de guany a les dades rebudes que s'aplicaran.
- Gain: un cop especificat el tipus de guany amb aquest paràmetre especificuem el valor de dB que li donem.
- SamplesPerFrame: Nombre de mostres que conté cada trama rebuda.
- OutputDataType: Tipus de dades rebudes.

Una altre estructura també molt útil que hem utilitzat al llarg del projecte és la que es pot veure a la figura 42 que mostra diverses característiques de les dades rebudes com són: la constel·lació, l'espectre i el senyal en el domini temporal.

```
spectrumScope = dsp.SpectrumAnalyzer('SampleRate', RadioBasebandRate);
timeScope = dsp.TimeScope('TimeSpan',lframe,'SampleRate',RadioBasebandRate);
constellation = comm.ConstellationDiagram('ShowReferenceConstellation', false);

% Create a container for the three scopes
scopesContainer = HelperCreateScopesContainer(...
    {spectrumScope,timeScope,constellation},...
    'Name','Zynq Radio QPSK Receiver',...
    'Layout',[2 2],...
    'ExpandToolbar',false);
scopesContainer.setColumnSpan(1,1,2);
```

*Figura 42. Configuració estructura visualització de les dades.*

També algunes de les funcions que ens ofereix aquesta nova Toolbox i que són de gran importància en el projecte són:

- sdrReceiver(): aquesta funció activa el receptor sdrReceiver perquè rebi les dades transmeses.
- transmitRepeat(sdrTransmitter, X): fa la funció de transmetre les dades X pel transmissor configurat sdrReceiver. El funcionament d'aquesta funció és que emet de forma repetida el senyal X fins que el transmissor es tenca o s'especifica una altra senyal a enviar.
- release(estructura): permet tancar estructures, és molt important per tancar els receptors i transmissor que s'activin amb les funcions anteriors, però també serveix per tancar les estructures que ens mostren algunes característiques de les dades.

## 4.2.2. Eines per visualitzar els resultats en Matlab

Per visualitzar els resultats i estudiar el comportament del sistema que s'ha creat en la transmissió i recepció de dades o els efectes del canal:

- plot(X,Y): crea una gràfica de línies en 2D de les dades en Y davant els valors corresponents en X. Ens servirà per veure l'evolució depenen de l'angle de transmissió de mòdul del senyal rebut o de la fase del senyal. [10]
- scatterplot(): crea un diagrama de dispersió per mostrar el senyal d'entrada x al pla IQ. Concretament, el pla IQ mostra els components en fase i en quadratura d'un senyal modulat en l'eix real i imaginari d'una trama xy. [11]
- Estructura explicada en l'apartat 4.2.1. en l'estructura es mostren l'espectre, el senyal en el domini temporal i la constel·lació. Tots aquests anàlisi permetrà identificar la majoria de comportaments del senyal.

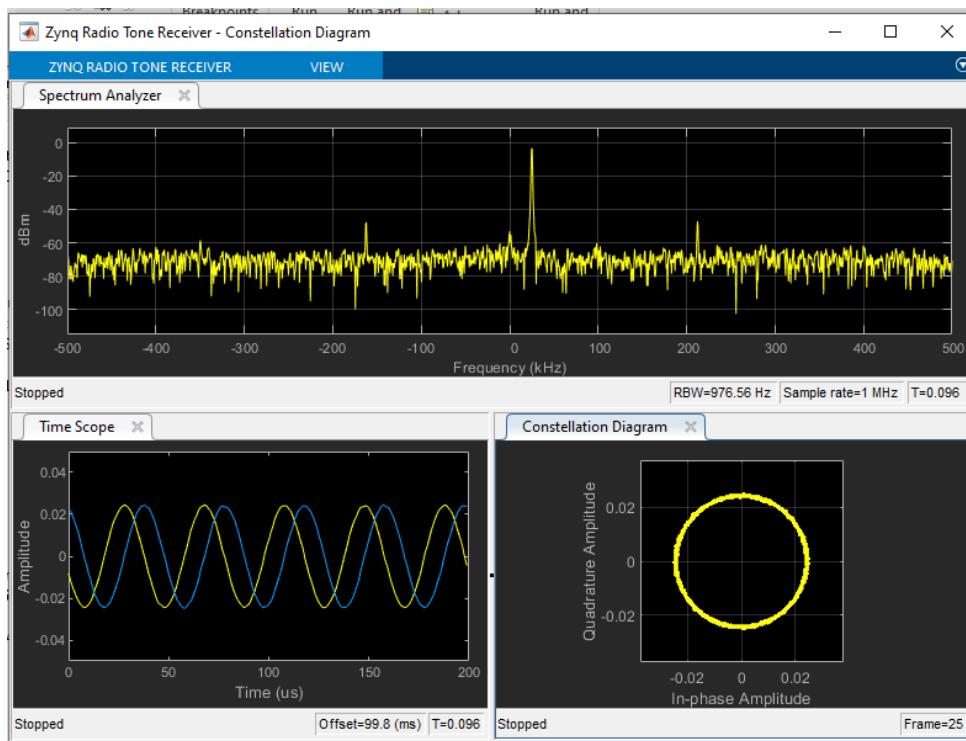


Figura 43. Estructura per la visualització senyal rebut.

### 4.3. Exemples de configuració

Per treballar amb les noves eines, primer cal entendre com funcionen aquests, així que s'ha adaptat el codi anterior per calcular les diferents fases dels transmissors, ja que cada transmissor inicia la seva fase diferent a la resta i cal un procés de calibració per poder dur a terme certs processos que depenen de la fase. Aquest procés permetrà calibrar els diferents transmissors, la qual cosa és un procés ens permetrà tindre en compte les diferents fases per dur a terme certes aplicacions.

Prèviament, per adaptar-se a la nova llibreria s'han fet algunes proves per introduir-se a la nova eina: primer s'enviarà un to pel canal, després 4 tons pels diferents 4 canals, posteriorment modulació QPSK per un canal i posteriorment pels diferents canals. D'aquesta forma podrem anar avançant progressivament i provant diferents elements de configuració.

Primerament s'ha enviat un to pel transmissor, al fer-ho podem observar el següent resultat en la figura inferior, on en freqüència es pot veure clarament el to com destaca, també podem observar el senyal en temps i la seva constel·lació.

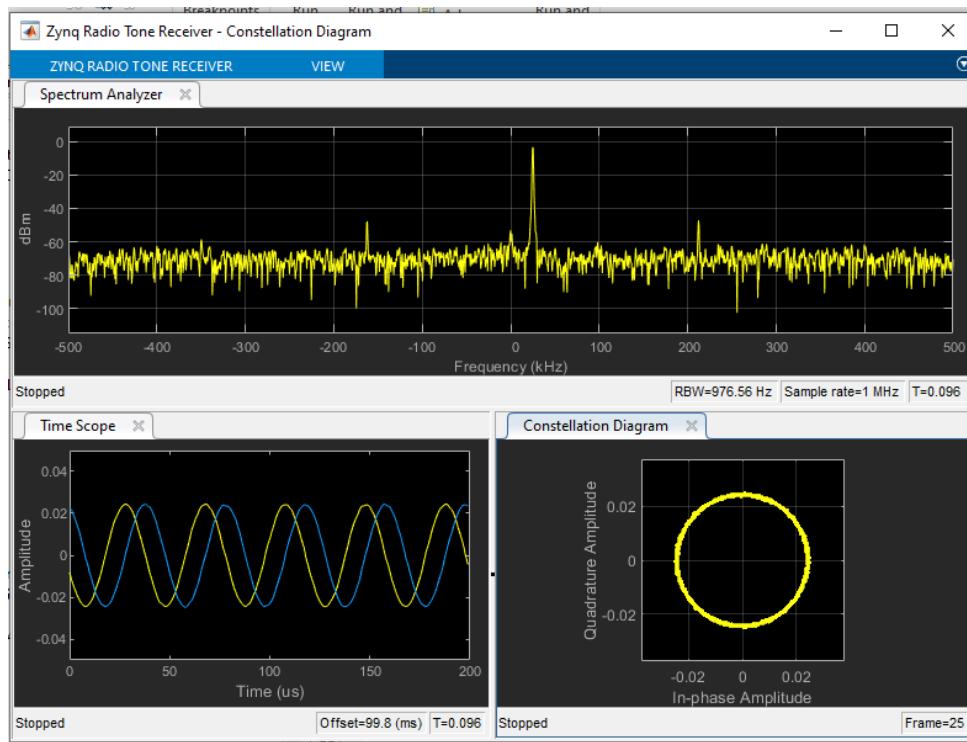


Figura 44. Visualització dades emeten un to pel canal.

Un element que podem observar es que si intentem rebre pel canal 2 l'únic que rebem és soroll, ja que el sistema està configurat per rebre només pel canal 1, de forma que al receptor del canal 2 no hi ha res connectat.

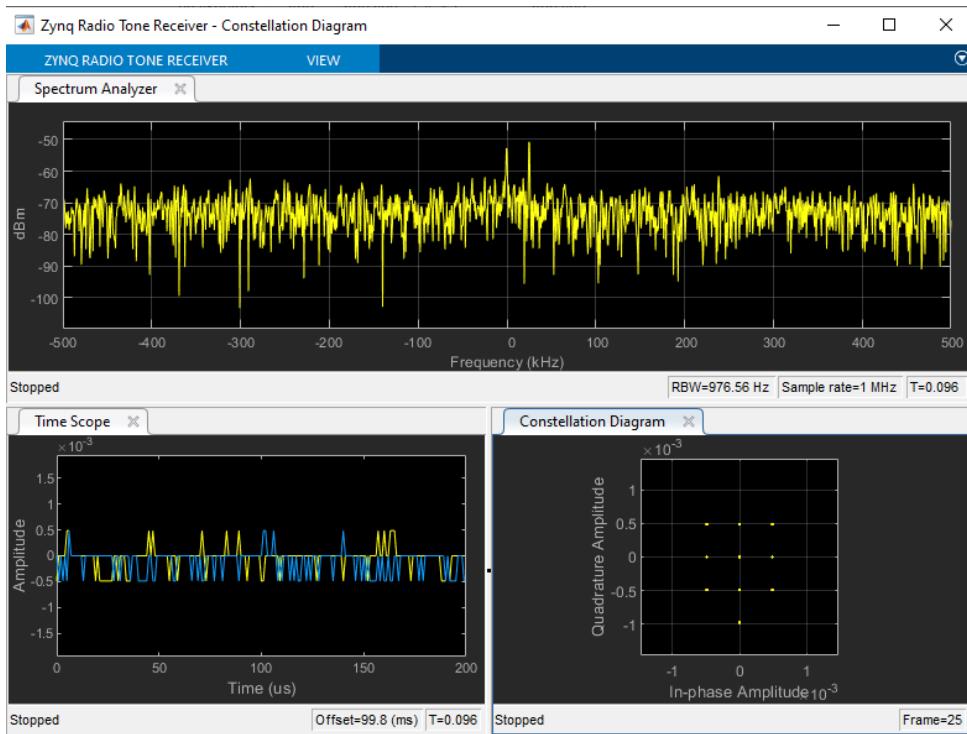


Figura 45. Visualització dades emeten un to pel canal per un receptor diferent al configurat.

A continuació configurem els transmissors per enviar per cada un dels canals un pols, cada un dels pols estarà separat per 25 KHz, a la sortida es pot veure que el senyal rebut és l'esperat, ja que es poden identificar clarament al receptor si escoltem pel canal 2, els 4 pics de les diferents senyals emeses per cada un del canals separades pels 25 KHz.

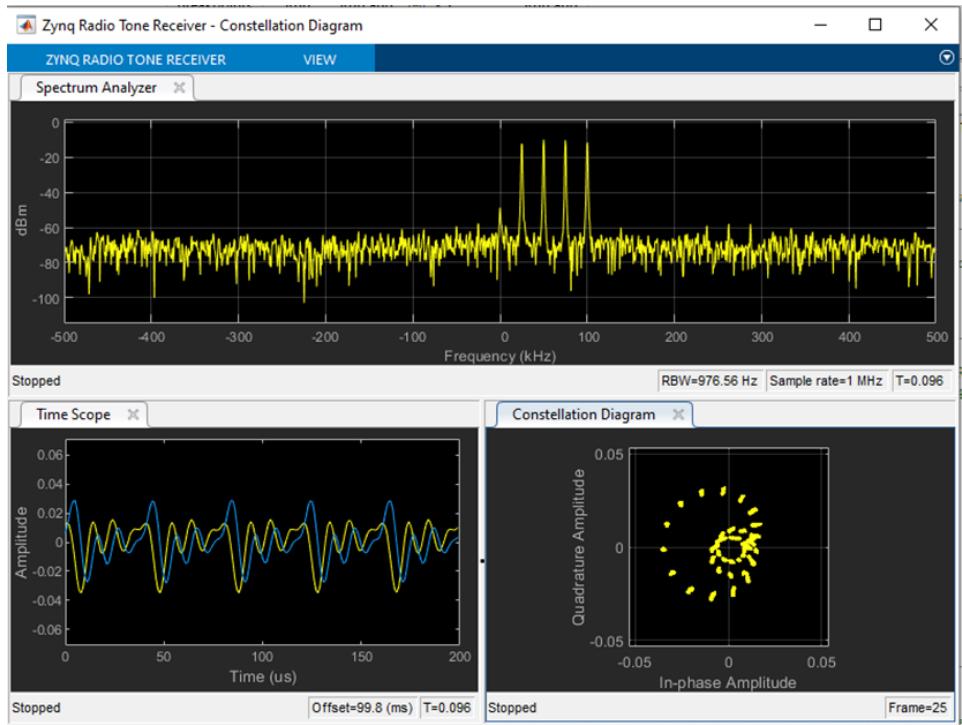


Figura 46. Emetre un to per cada transmissor amb diferència de 25 KHz entre ells.

En aquest cas, quan mirem el senyal rebut al receptor 2, es pot veure que el senyal emès pels transmissors s'està acoblant, es pot observar el resultat en la figura 47, com si s'hagués rebut una transmissió, però atenuada, però obviament això no implica que s'està rebent un senyal directament pel receptor 2, simplement que no està del tot aïllada.

Es pot apreciar en el gràfic de l'amplitud que el senyal rebut per aquest canal 2 és bastant similar a la primera on s'envia un to de freqüència, però el senyal és més feble si ens fixem en l'escala i a més la sinusoide no és tan perfecte. En la constel·lació també es pot apreciar que es rep molt més irregular i en l'anàlisi de l'espectre ens pot donar a pensar que és el senyal enviat pel segon canal del transmissor (a 50 kHz) la que s'està acoblant al canal 2, ja que es pot veure com apareixer un pic en aquesta freqüència.

El codi corresponent a la transmissió d'aquests 4 tons es pot trobar en l'annex 8.1.2, si l'anàlitzem en detall es pot entendre com funciona amb les noves Toolboxes de Matlab i els codis per altres proves que s'han realitzat en aquesta secció que són bastants similars. Els passos que s'han seguit són:

1. Configurar el SDR i la seva IP (línes 39-42).
2. Configurem el transmissor, configurant el canal de transmissió, la freqüència central i la IP (línes 58-78).
3. Configurem la senyal que es vol emetre, en aquest cas els 4 tons (línes 79-90).
4. Transmetem el senyal, en aquest cas ho hem fet com a la línia 99.
5. Configurem el receptor, especificant la IP, el guany, el tipus de dades de sortides que volem, el canal de recepció, la longitud dels frames, la freqüència central i l'ampla de banda (línes 123-138).
6. Configurem el per l'anàlisi del senyal rebut (línes 140-158).
7. Configurem la recepció de dades, entre altres l'instant de temps final de la recepció de dades (línes 157-179).
8. Fem el procés per la recepció de dades, llegint en cada iteració d'un bucle un frame.
9. En cada iteració es poden tractar les dades, emmagatzemar-les, analitzar-les o visualitzar-les per cada un dels frames.
10. Tanquem les diferents estructures creades durant el procés: el receptor, el transmissor i l'estruatura per l'anàlisi (línes 244-249).

Per la resta d'operacions, principalment es modificaran les dades que es transmetran, enviant un sol to, o enviant alguna combinació de símbols modificats amb una QPSK, però també es poden variar aspectes de configuracions per analitzar el seu comportament com són per exemple el guany en el receptor o canviar el canal receptor.

Per veure en més detall tot el procés per l'enviament de 4 tons pel canal en l'annex 8.1.2.

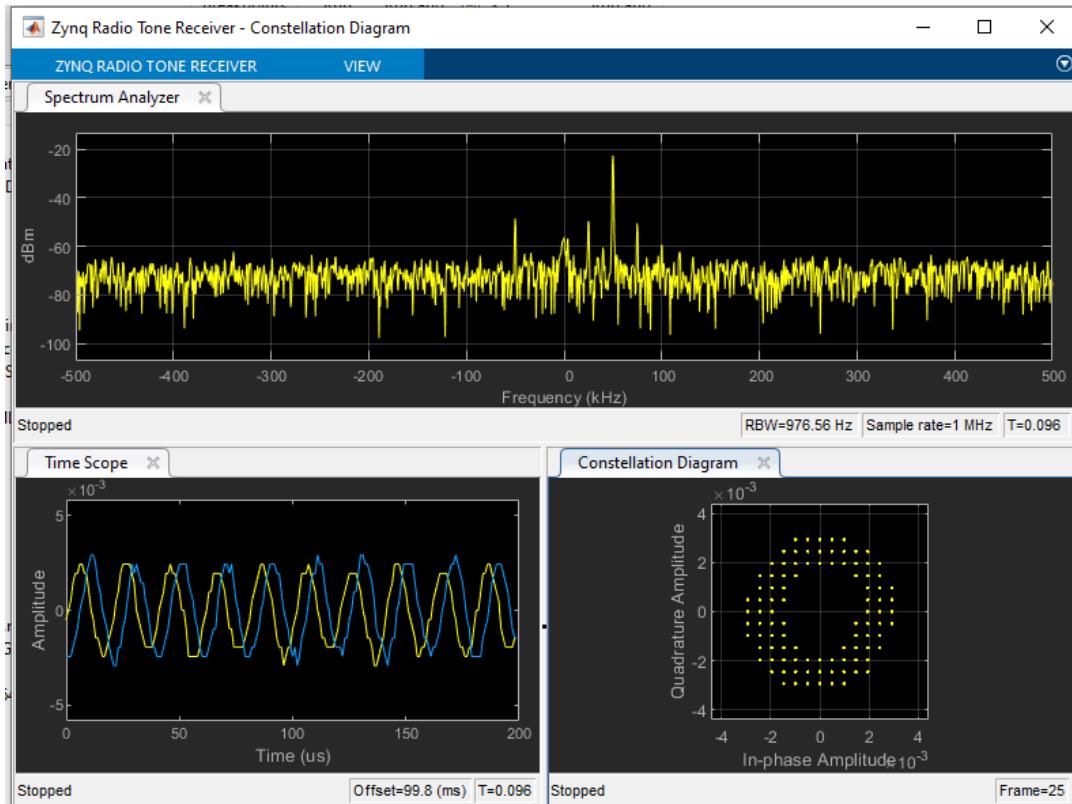


Figura 47. Emetre un to per cada transmissor amb diferència

de 25 KHz entre ells en un receptor diferent.

El següent pas que s'ha seguit és enviar un senyal QPSK, primerament s'ha fet enviant el senyal QPSK per un sol canal, de forma que utilitzant el codi del projecte en que ens hem basat, que es pot trobar en el l'annex 8.1.1, s'ha adaptat pel funcionament de les noves eines i s'ha enviat 15 frames, tots iguals entre ells, de símbols generats de forma aleatòria, excepte els dos primers de cada frame que s'enviaven 0s, d'aquesta forma es pot identificar fàcilment l'inici de cada frame. El resultat obtingut és com el de la següent figura 48.

Posteriorment, s'ha modificat per enviar una modulació QPSK per cada un dels canals de transmissió, mentre a la resta de canals no s'enviava res. Un aspecte que s'ha notat i s'ha provat és que el resultat d'enviar zeros per la resta canals de transmissió o no especificar què s'envia per la resta de canals no afecta el resultat de l'execució, és a dir que no cal especificar que s'envia per la resta de canals si no ens interessa.

El resultat d'enviar una QPSK per cada canal de forma seqüencial és similar a l'anteriorment vist quan s'enviava un senyal QPSK per un sol canal, però en aquest cas com es pot apreciar en les següents dues figures (figura 48 i 49), per cada un dels canals hi ha una variació de la fase, es pot apreciar clarament en la rotació de la constel·lació QPSK en les següents figures. Cada canal té una rotació diferent i és per això que cal una

calibració per poder controlar les diferències en fase dels diferents canals, ja que aquesta diferència en fase és diferent en cada inicialització del dispositiu AD-FMCOMMS5, pel fet que cada cop que s'inicialitza el dispositiu FM-Comms 5, cada canal s'inicialitza d'una forma aleatòria d'acord amb certs paràmetres, però de diferent forma que la resta de canals.

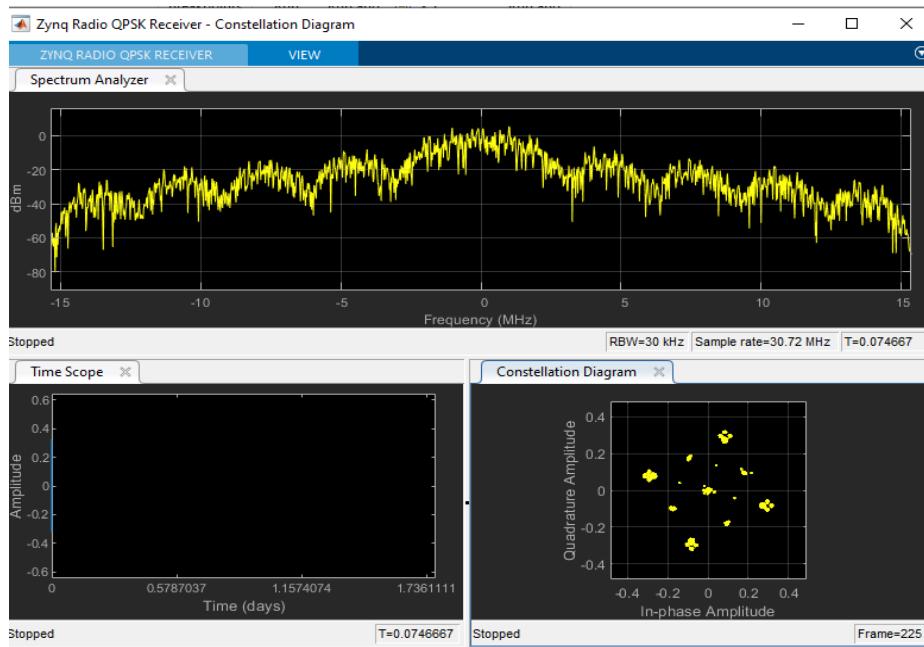


Figura 48. Senyal rebut en el canal 1 durant transmissió QPSK.

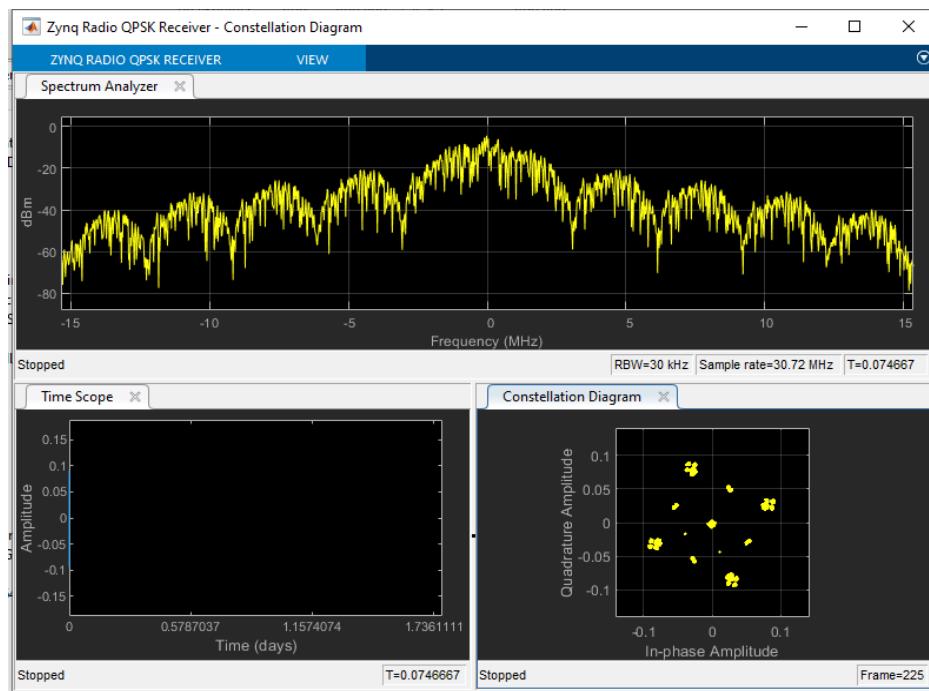


Figura 49. Senyal rebut en el canal 2 durant transmissió QPSK.

De forma que a la sortida es poden apreciar 4 diferents diagrames de constel·lacions, corresponent a cada un dels canal, el següent pas per tant serà el procés de calibració del sistema.

## 4.4. Calibració

La calibració del sistema consistirà a mesurar la diferència de fase entre cada un dels transmissors i el receptor, la diferència de fase per cada un d'ells serà diferent i cada cop que s'inicialitzi el SDR també variarà. Això és a causa del fet que la diferència de fase és la suma de la diferència de dos processos, hi ha un component degut a la variació de fase en l'enviament de dades, és a dir la variació de fase durant la propagació del senyal entre el receptor i el transmissor, aquesta variació serà similar per a tots, ja que la distància és similar per tots ells; el segon component serà depenen de la inicialització del sistema, quan el sistema s'inicialitzi cada un dels oscil·ladors del transmissor s'inicialitzarà en base a uns paràmetres diferents a la resta i això introduirà un nou desfase entre l'emissor i el receptor diferent per cada un dels transmissors.

Per fer la calibració enviarem unes dades modulades per una QPSK per un dels transmissors mentre la resta de transmissors emetran 0s. Les dades enviades per evitar la distorsió es repetiran 10 vegades (variable ls) cada una. D'aquesta forma la tassa de transmissió es veurà perjudicada, ja que per cada bit que es vol enviar, se n'hauran d'enviar 10 bits pel canal, de la mateixa manera que els símbols. El receptor rebrà les dades i les guardarà, amb aquestes dades caldrà processar-les: trobar l'inici de les dades amb la convolució circular, buscant el mínim que serà quan els dos 0s inicials per cada frame enviat estaran al principi, es rotaran circularment les dades per ordenar-les i finalment llegirem una de cada 10 dades per recuperar els símbols que es volien enviar inicialment, es llegiran les dades del punt mig ja que aquestes patiran menys distorsió.

Es pot veure el resultat d'una execució de la convolució circular en la figura 50 on s'ha fet la convolució circular entre les dades rebudes i un vector de 0s excepte 1s en dos símbols:

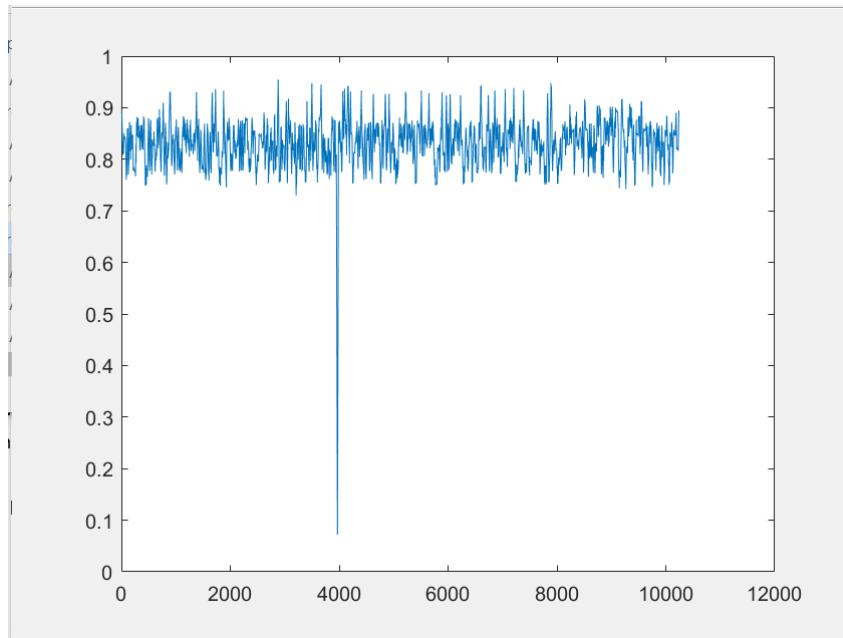


Figura 50. Convolució circular de les dades rebudes amb vector de 0s excepte dos 1s.

Es pot veure que l'inici de les dades rebudes es troben aproximadament a partir de la dada 4000.

En el procés de calibració han sorgit algunes complicacions que cal esmentar i a continuació s'exposaran.

Inicialment el procés de calibració donava resultats diferents en cada una de les execucions que es realitzaven el desfase donava un resultat diferent, això indicava que hi havia alguna cosa del procés que no funcionava correctament, es van reduir les mostres enviades de 1024 a 14 mostres conegeudes amb un patró d'aquesta forma es podria identificar si hi havia alguna problemàtica en la recepció de les dades, el patró es tractava de la següent cadena de símbols: 0, 0, 1, 2, 3, 4, 1, 1, 2, 2, 3, 3, 4, 4; també s'ha modificat el nombre de repeticions de cada una de les variables ( $ls$ ) de 10 a 100 de forma que es pugui apreciar millor els símbols enviats. El resultat rebut es pot veure en la figura 51.

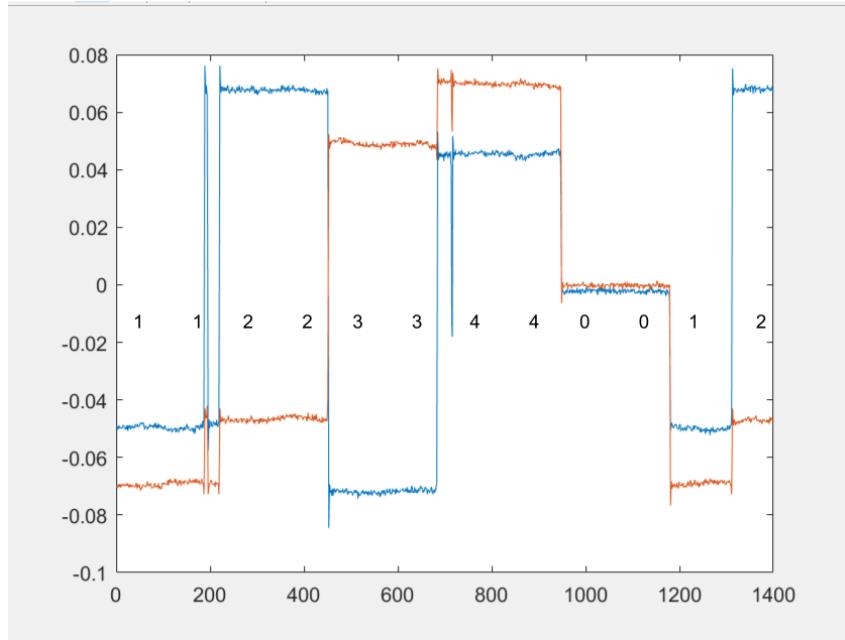


Figura 51. Símbols rebuts amb  $fs=30.72\text{ MHz}$ .

Es pot observar clarament que no tots els símbols enviats s'estan reben, de la cadena els símbols: 3, 4 no es reben, també es pot identificar que cada un dels símbols enviats es repeteix més de 100 vegades, això ens porta a pensar que alguna freqüència de mostreig no està funcionant correctament. També es pot observar un anàlisi general de les dades rebudes en la figura 52.

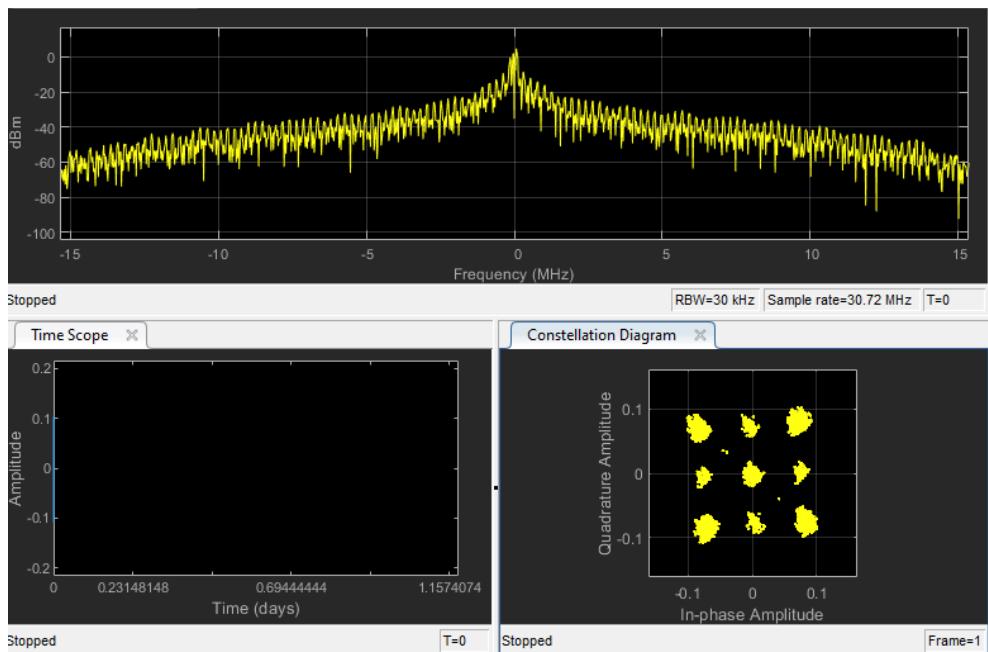


Figura 52. Anàlisi dades rebudes per una seqüència controlada amb  $fs=30.72\text{ MHz}$ .

Particularment en el mapa de constel·lació, es pot apreciar que la quantitat de mostres en les seccions de transició i en el punt 0 és molt elevada.

Per solucionar el problema esmentat anteriorment de forma arbitraria s'ha reduït la freqüència de mostreig de la meitat, de 30.72 MHz a 15.36 MHz. I el resultat ha sigut el següent:

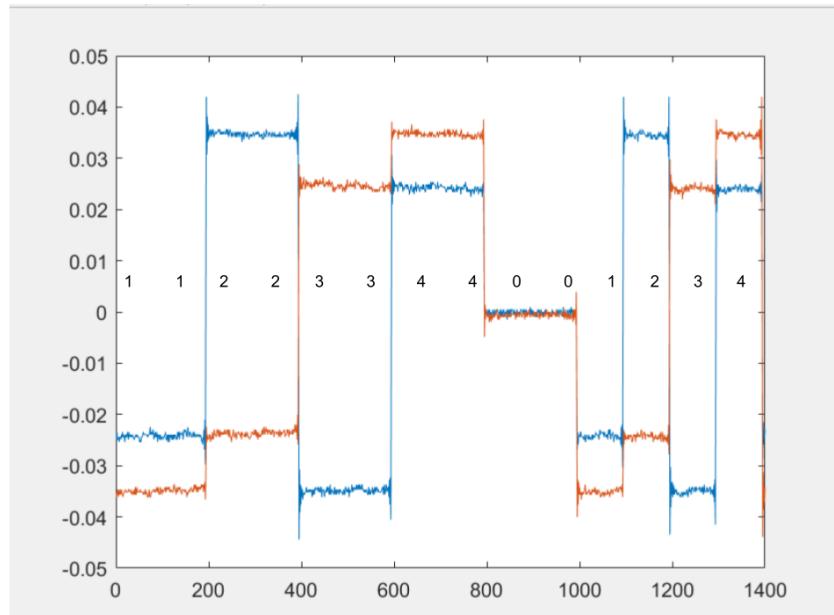


Figura 53. Símbols rebuts amb  $fs=15.36$  MHz.

Es pot apreciar que quan s'ha reduït si és possible identificar tota la cadena de símbols enviats, de forma que la freqüència a la qual s'estava mostrejant inicialment no funciona correctament i s'ha modificat a la meitat de forma arbitrària. Això comporta una gran pèrdua en la taxa de bits enviats per segon, ja que s'ha reduït a la meitat, però l'interès d'aquest estudi és en la funcionalitat del sistema per sobre de l'eficiència i no es considera rellevant pel resultat final, de forma que s'ha acceptat aquesta pèrdua d'eficiència, sense buscar el punt òptim per la freqüència de mostreig del sistema. A part d'aquesta forma per millorar la tassa d'enviament del sistema també s'ha notat que la taxa de repetició dels símbols enviats ( $I_s$ ), també es pot reduir per millorar la tassa d'enviament total del sistema, ja que observant les dades rebudes es pot notar que la diferència de la distorsió entre les dades situades en el centre de cada pols rebut és gairebé nul·la, de forma que afectaria en el rendiment del sistema.

El resultat que es pot apreciar en l'anàlisi general és el que es pot observar en la figura 54.

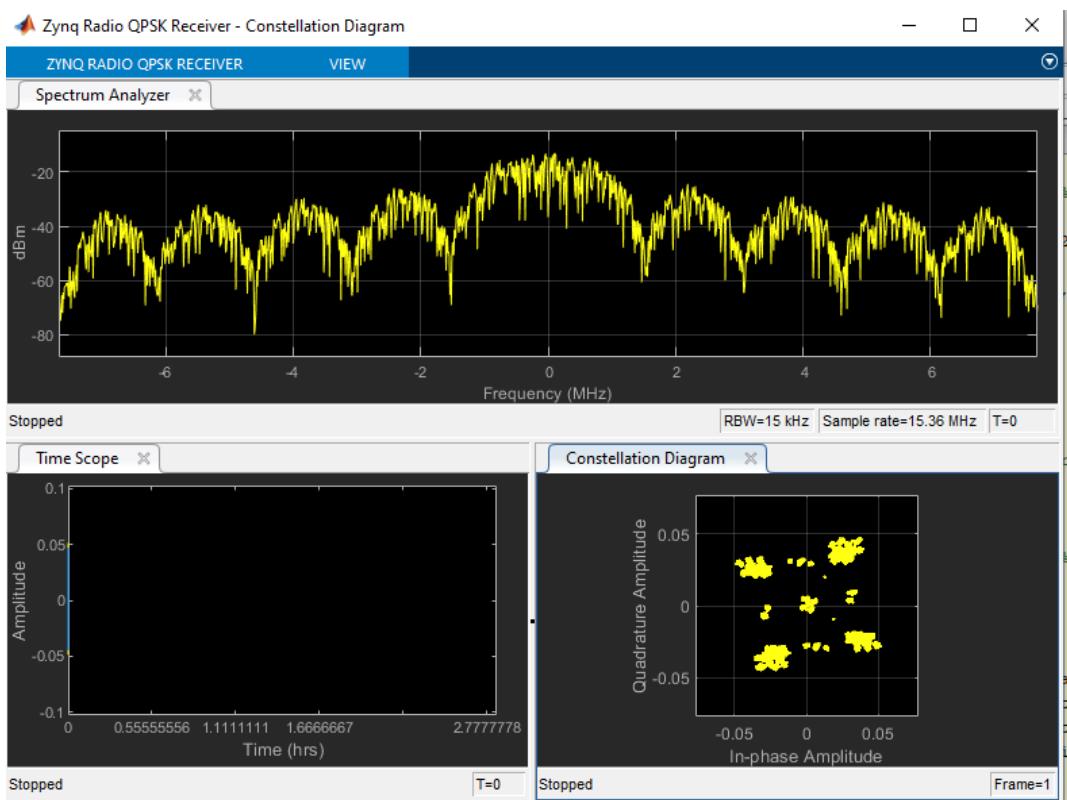


Figura 54. Anàlisi dades rebudes per una seqüència controlada amb  $f_s=15,36$  MHz.

Es pot observar que la quantitat de mostres que es troben en la transició i en el punt 0, s'ha reduït molt, de forma que es pot identificar més clarament la constel·lació de la QPSK.

Per observar-la millor s'han mostrat les dades en l'eix IQ i s'ha pogut apreciar el següent resultat:

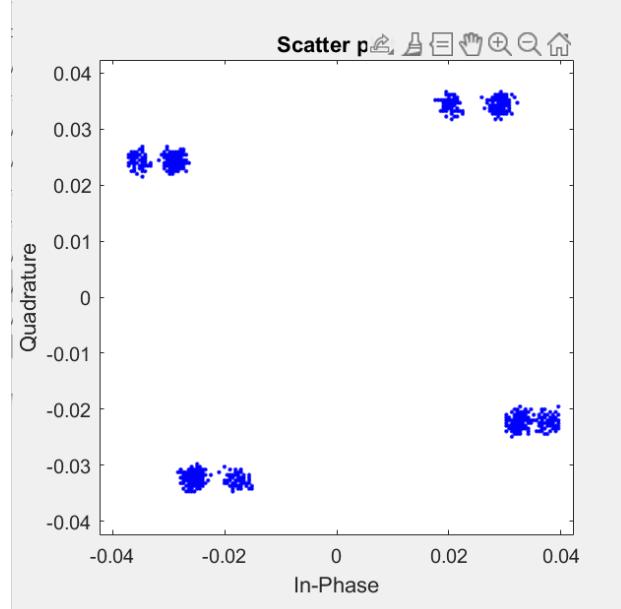


Figura 55. Constel·lació rebuda amb  $G=30$  dB emeten senyal QPSK.

Es pot observar en el gràfic com si la constel·lació de la QPSK s'hagi duplicat o els punts es concentressin en dos centroides per cada un dels punts de la constel·lació, per solucionar aquest error s'ha modificat la freqüència de recepció de 30 dB a 32 dB, essent el resultat final a l'observar la constel·lació:

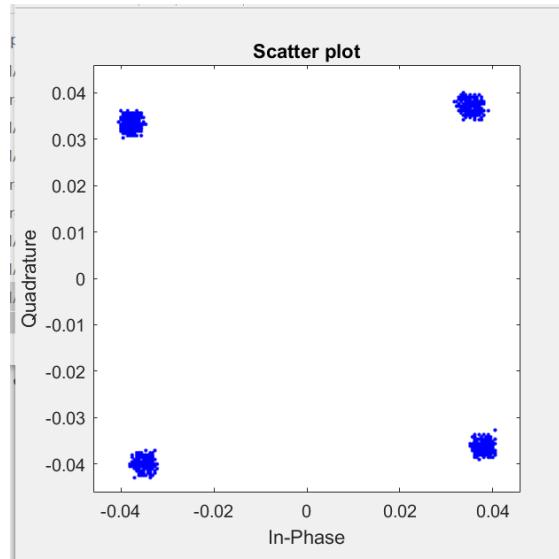


Figura 56. Constel·lació rebuda amb  $G=32$  dB emeten senyal QPSK.

Es pot observar en la figura 56 els 4 punts de la constel·lació clarament definits, sense soroll en els punts de transició, en els símbols finals obtinguts (eliminant les dades reiteratives). Particularment en aquesta execució, es pot veure que la diferència de fase entre la QPSK enviada i la rebuda és molt petita.

També es pot fer un anàlisi més general del resultat amb totes les dades rebudes:

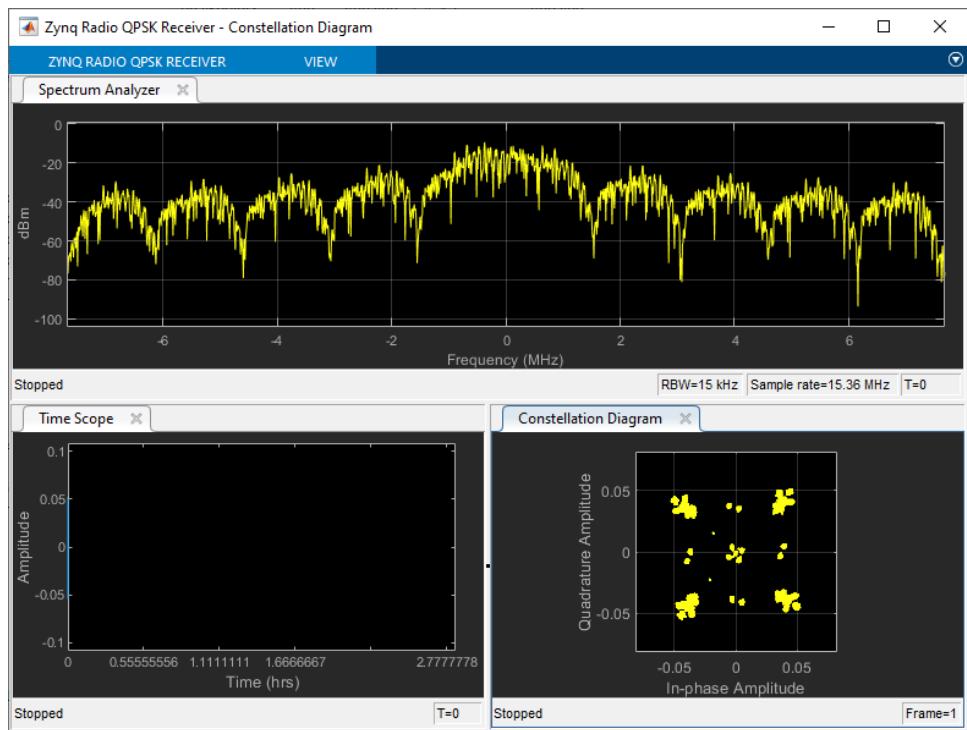


Figura 57. Anàlisi senyal rebut amb  $G=32$  dB emeten senyal QPSK.

Es pot observar que la constel·lació del total de dades rebudes no té quasi soroll degut a les transicions.

Finalment, per diferents execucions el resultat de la diferència de fase és la mateixa sempre i quan no es reiniciï el sistema SDR, això té sentit, ja que cada vegada que s'engeguen els transmissors, s'inicialitzen les seves fases de forma diferent, provocant que la fase entre els transmissors i cada vegada que s'engega sigui diferent. Analitzant amb més detall, totes les diferències de fases en aquesta execució eren molt petites de forma que la diferència de fase de la constel·lació final per cada un dels transmissors també era molt petita com s'ha pogut observar anteriorment, d'aquesta forma confirmant que la calibració funciona correctament, per a poder recuperar perfectament la constel·lació enviada, caldria rotar la fase obtinguda en cada un de les mostres rebudes.

En l'annex 8.1.4 es pot trobar el codi utilitzat per calcular les fases entre els diferents transmissors del SDR.

#### 4.4.1. Transmissió amb el sistema calibrat

Un cop s'han calculat les fases per cada un dels transmissors, corregirem les fases en cada un d'ells durant la transmissió de forma que tots arribin al destí amb la mateixa fase, fent això el que podem esperar és que els senyals tinguin interferència constructiva entre elles i al destí la potència augmenti. Per tal de comparar primer enviarem uns símbols sense calibrar el sistema que podem veure el seu resultat en la figura 58.

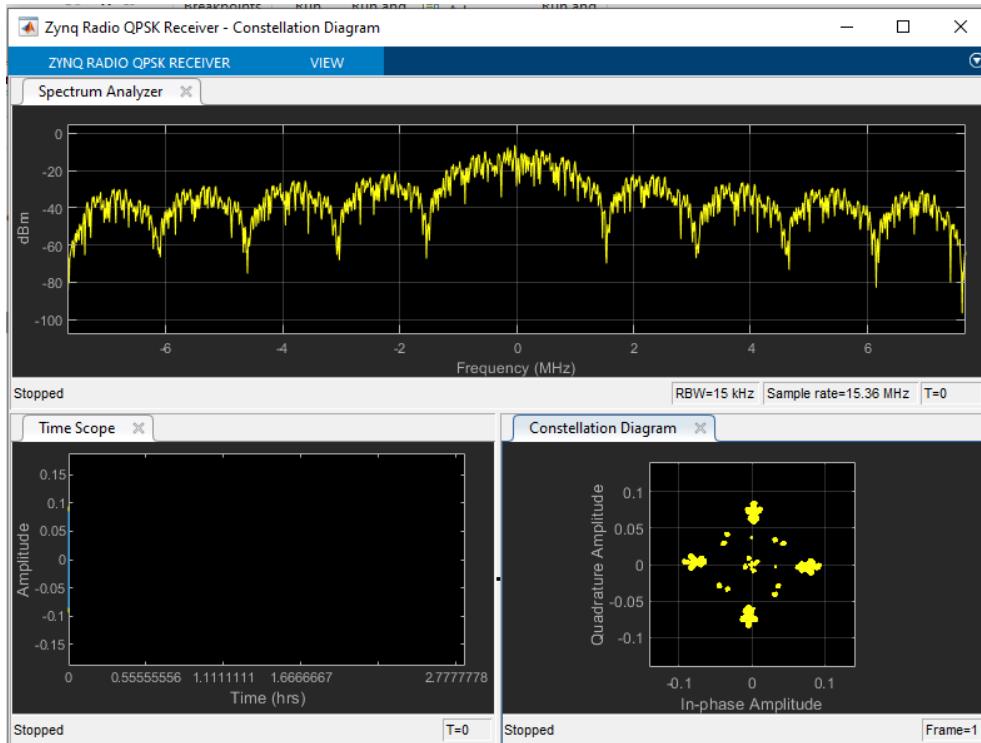


Figura 58. Visualització dades sense calibrar el sistema.

Posteriorment enviarem uns símbols calibrant el sistema, analitzarem el senyal rebut, que podem observar a la figura 59, i compararem els resultats. Per calibrar el sistema multiplicarem els símbols per la matriu hermítica del canal normalitzant-la. De forma que el senyal enviat pel canal respondrà a l'equació:

$$Y_h = \vec{H}^\dagger \vec{S}$$

On  $\vec{H}^\dagger$  està normalitzada.

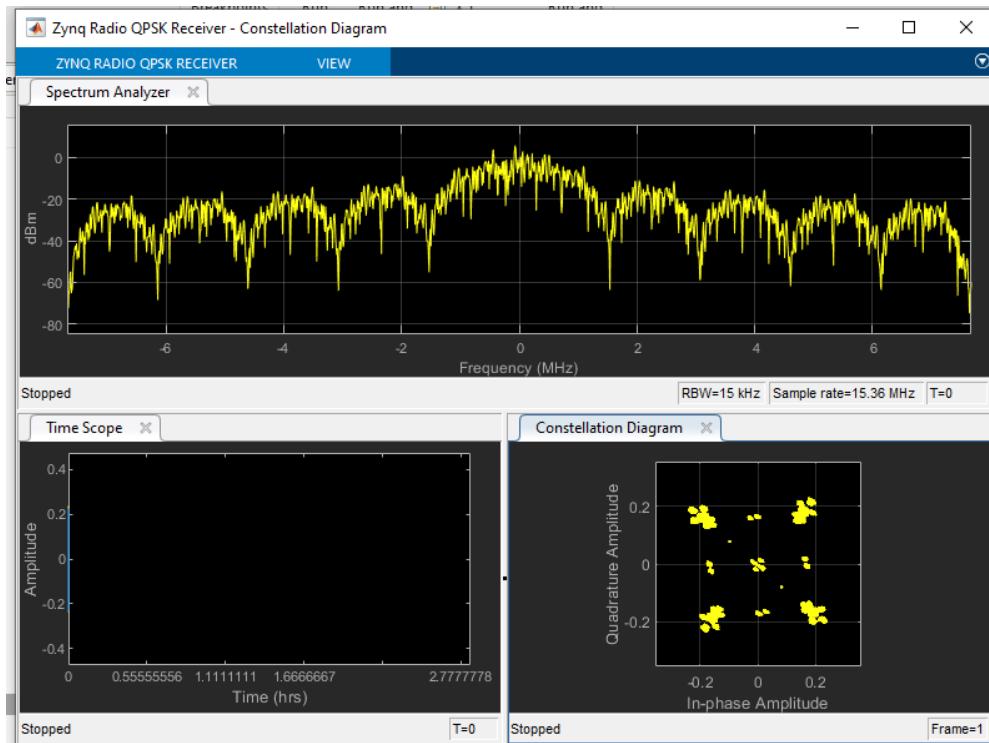


Figura 59. Visualització dades havent calibrat el sistema.

Es pot observar que la constel·lació queda rotada quan no calibrem al sistema i a més es pot observar que l'amplitud en els gràfic de la constel·lació i en el domini temporal és el doble quan el sistema és calibrat, això també es tradueix en una diferència de dBm apreciable en el domini freqüencial.

En els annexos es pot trobar el codi utilitzat per transmetre calibrant els transmissors del SDR.

## 5. Modulació direccional

### 5.1. Introducció

Com hem vist quan parlàvem dels avantatges i inconvenients dels Software Defined Radio, un dels inconvenients eren les noves possibilitats d'atacs que suposen. Al tenir un sistema de comunicació sense fils és molt més senzill pels atacants rebre o escoltar la informació, el que suposa greus problemes de la privacitat. Això suposa que no es pugui enviar informació confidencial o sensible amb aquest sistema. A més la idea d'enviar patrons destructius als receptors que volem evitar és impracticable, ja que, també degut a l'accessibilitat que s'ha donat al públic a sistemes de telecomunicacions, no es pot determinar on estan tots els receptors.

D'aquest problema neix la necessitat de buscar sistemes o algoritmes que puguin transmetre la informació sense fils de forma segura, un exemple és la modulació direccional. La modulació direccional és capaç d'enviar informació a través d'una senyal, sense fils, en una direcció en concret i en tota la resta de direccions enviar informació inintel·ligible, d'aquesta forma només podria llegir la informació aquella persona que es troba en una ubicació en l'angle que es decidirà de forma arbitrària, d'aquesta forma s'aconseguirà que hi hagi seguretat en l'enviament de dades.

La idea essencialment es basa en afegir un soroll artificial de forma que gràcies a propietats de ortogonalitat, en un direcció triada el soroll s'eliminarà i es podrà rebre el senyal correctament, mentre que en la resta de direccions el senyal no es podrà rebre correctament i per tant no es podrà llegir la informació enviada.

L'aparició de la modulació direccional es dona el 2001, a Estats Units es va presentar una patent d'una capa d'una arquitectura de comunicació segura que funcionava com un transmissor de modulació direccional [3]. Es basava en la idea d'aprofitar la ortogonalitat de les ones de Walsh per generar senyals de modulació sensibles a la direcció, posteriorment es va desenvolupar amb la idea de modulació de subconjunt d'antenes que juntament amb l'arquitectura de modulació direccional d'espectre estès entren en la categoria d'agrupacions d'antenes modulades en temps.

La figura 60 mostra la idea d'aquesta modulació, únicament la persona situada en l'angle  $\Theta_0$ , serà capaç de llegir la informació enviada correctament.

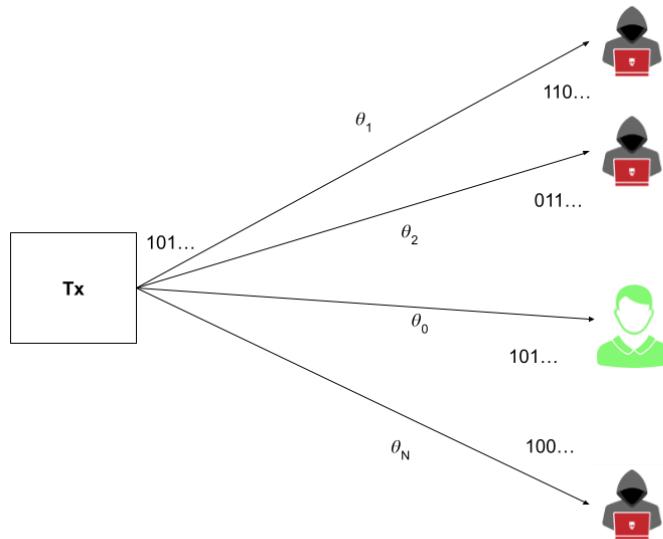


Figura 60. Idea general del propòsit de la modulació direccional.

Les arquitectures de la modulació direccional consisteixen en agrupacions d'antenes impulsades activament amb canvis de fase configurables, amb algoritmes d'optimització. En la figura 61 es pot veure l'arquitectura general d'un transmissor de modulació direccional:

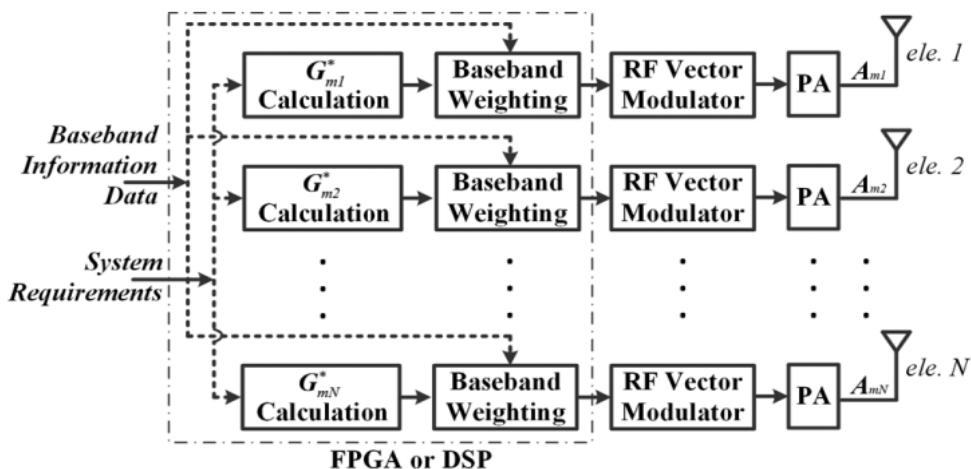


Figura 61. Arquitectura d'un transmissor de modulació direccional. [3]

Primer veurem una explicació i demostració matemàtica de la modulació direccional.

## 5.2. Fonaments teòrics per generar DM

En el nostre cas, el sistema SDR utilitzat (ad-fmcomms5-ebz) està format per 4 transmissors de forma que s'utilitzaran aquests 4 transmissors per aconseguir la modulació direccional. La figura mostra un esquema de la radiació dels diferents transmissors:

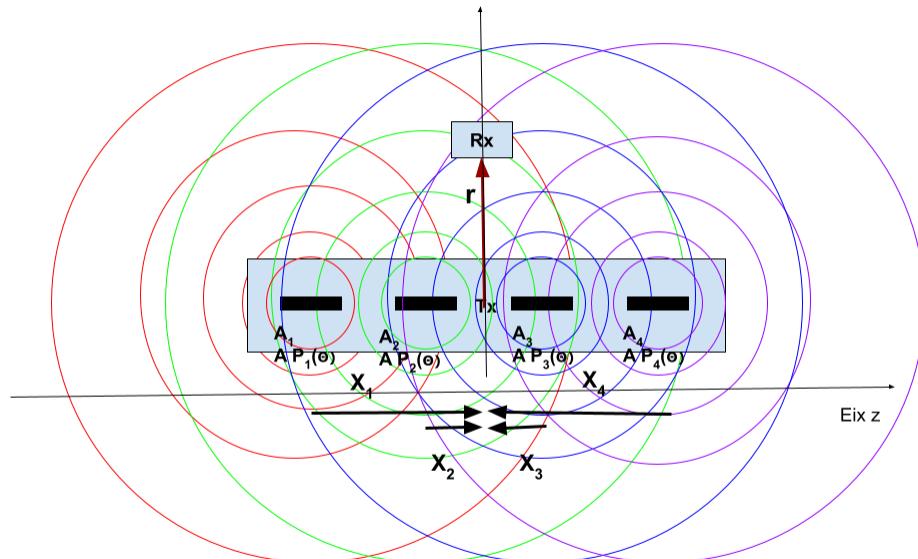


Figura 62. Esquema fórmula modulació direccional.

El camp elèctric quan hi ha 4 antenes radiant en una direcció es pot definir com:

$$E(\theta) = \frac{e^{-j\vec{k} \cdot \vec{r}}}{|\vec{r}|} \cdot |AP_1(\theta) \quad AP_2(\theta) \quad AP_3(\theta) \quad AP_4(\theta)| \begin{vmatrix} A_1 \cdot e^{j\vec{k} \cdot \vec{x}_1} \\ A_2 \cdot e^{j\vec{k} \cdot \vec{x}_2} \\ A_3 \cdot e^{j\vec{k} \cdot \vec{x}_3} \\ A_4 \cdot e^{j\vec{k} \cdot \vec{x}_4} \end{vmatrix} \quad (3)$$

- $\vec{k}$  és el vector de propagació de l'ona.
- $\vec{x}_n$  representa el vector des del transmissor n fins al centre dels transmissors des del qual es traça el vector r
- $\vec{r}$  representa el vector des del centre dels transmissors fins el receptor.
- $A_n$  és un nombre complex que representa el pes que té cada antena en el camp generat.

- $AP_n(\theta)$  Representa el diagrama de radiació per cada antena, que mostra la potència en la que radia en una direcció l'antena.

Una forma més visual d'entendre la fórmula, és amb la figura 62, es pot apreciar els diferents transmissors radiant de forma omnidiireccional (aleshores  $AP_n(\theta) = 1$ ), és a dir en  $360^\circ$  de l'espai, i el receptor rep les diferents ones radiades. En el gràfic es pot observar la majoria d'elements de la fórmula, bàsicament es pot observar que  $r$  és un vector que mostra la direcció i distància del transmissor fins al receptor,  $x_n$  mostra la direcció des de cada un dels transmissors fins al receptor, aquest vector té més importància per la diferència de fase que pot aportar entre cada un dels transmissors que amb el mòdul que té un efecte menor.  $A_n$  indica l'alimentació amb la que s'exciten els transmissors i finalment  $AP_n(\theta)$  és un paràmetre del diagrama de radiació, si és 1 vol dir que radia de forma uniforme les mostres.

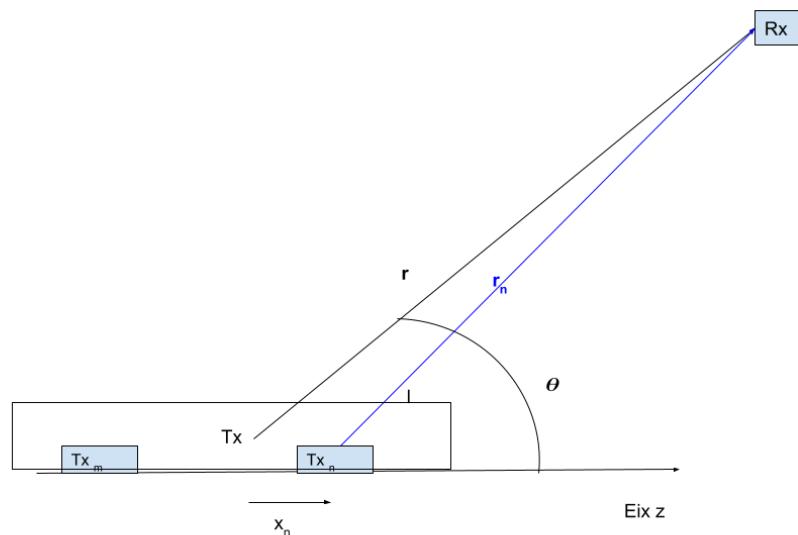


Figura 63. Esquema vectors  $r$  i  $x_n$  modulació direccional.

Per entendre millor la fórmula i els vectors  $\vec{r}$  i  $\vec{x}_n$  i d'on surt el cosinus de  $\theta$ . Si suposem que el receptor es troba molt lluny de l'emissor aleshores es pot suposar que els vectors representats a la imatge  $\vec{r}$  i  $\vec{r}_n$  són paral·lels. Es pot suposar que el vector  $\vec{x}_n$ , que relaciona el centre dels transmissors amb la posició dels diferents transmissors, és de la forma  $(0, 0, d)$ , de forma que al multiplicar amb el vector  $k$  que serà de la forma  $\vec{k}=k(\sin(\theta)+\cos(a), \sin(\theta)+\cos(\theta), \cos(\theta))$  només quedarà el component en  $z$  amb el  $\cos(\theta)$ . Finalment es podrà suposar degut a que s'interpreta que són paral·lels  $\vec{r}$  i  $\vec{r}_n$ , que es desplaçarà en fase  $\vec{x}_n + \vec{r}$  quan s'emet des del transmissor  $n$  al receptor.

De la mateixa manera, també es pot entendre com el sumatori de les contribucions de camp elèctric emès per cada una de les antenes:

$$E_m(\theta) = \sum_{n=1}^4 (A_{mn} \cdot e^{j\pi(n - ((4+1)/2))\cos\theta}) \quad (4)$$

On  $A_{mn} = D_m \cdot G_{mn}$  (on  $D_m$  representa la modulació de les dades per l'emenéssim símbol i  $G_{mn}$  el guany complex de l'antena n per aquell símbol, el guany podria arribar a ser variable per a cada símbol enviat). Una altre forma que es pot considerar per aconseguir més seguretat és utilitzar aquest guany com un paràmetre variable de la modulació direccional, la qual cosa ens permetrà afegir un altre grau de llibertat i de seguretat.

Per aconseguir la modulació direccional hi ha dos conceptes claus que cal garantir:

- En la direcció que es decideix, és a l'angle seleccionat, en que es podran rebre les dades correctament s'ha de preservar format del senyal transmès. Aquest angle el definirem com  $\Theta_0$ .
- Distorsió des patrons de constel·lació en la resta de direccions.

Es pot obtenir el punt de la constel·lació en l'espai IQ, que anomenarem  $C_m$  com:

$$C_m(\theta) = \underbrace{\sum_{n=1}^4 (A_{mn} \cdot e^{j\pi(n - ((4+1)/2))\cos\theta})}_{B_{mn}(\theta)} \quad (5)$$

De forma que per cada símbol que es transmet s'ha de poder distingir el punt  $C_m$  únicament en la direcció  $\Theta_0$ , la direcció definida on el senyal es podrà recuperar.

Si definim l'espai IQ, cada suma dels diferents vectors pot seguir diferents camins fins a crear la modulació definida originalment, que en el nostre cas la definirem com la modulació QPSK. Per altra banda també si ens fixem en la variació de la QPSK segons l'angle d'emissió del senyal.

Algunes propietats que es poden identificar de les constel·lacions son [3]:

- Quan el nombre d'elements radials es impar, els camins de les constel·lacions a través dels angles són tancats. En canvi quan el nombre d'elements radials es parell és obert complint  $C_m(0^\circ) = -C_m(180^\circ)$ .
- Canviar la direcció  $\Theta_0$  només canvia l'inici i final del camí de constel·lacions.

- Diferents camins de vectors provoquen diferents camins de les constel·lacions.
- Pel qéssim símbol emés, si  $B_{qn}(\theta_0)$  és el factor escalat de  $B_{mn}(\theta_0)$  amb el mateix factor d'escalat per cada element radiant aleshores  $C_q(\theta_0)$  també escalarà a  $C_m(\theta_0)$  en un factor K.

Els darrers dos últims es basen en que si es poden garantir que els vectors de  $Bq(\theta_0)$  i  $B_m(\theta_0)$  són linealment independents, aleshores es podrà assegurar que la constel·lació es veurà distorsionada en altres direccions de l'espai.

D'aquesta forma, es pot aconseguir modulació direccional quan es compleixen les següents condicions:

$$\begin{cases} \sum_{n=1}^4 B_{mn}(\theta) = C_{m_{st}}(\theta_0) \\ [B_{q1}(\theta_0) \quad B_{q3}(\theta_0) \quad B_{q3}(\theta_0) \quad B_{q4}(\theta_0)] \neq \\ K \cdot [B_{m1}(\theta_0) \quad B_{m2}(\theta_0) \quad B_{m3}(\theta_0) \quad B_{m4}(\theta_0)] \end{cases} \quad (6)$$

A continuació, veurem com implementar la modulació direccional garantint que es compleixen les propietats anteriors, els passos que s'han de seguir i les transformacions a seguir i també com aquests passos garanteixen que s'assoleix l'objectiu principal de la DM, transmetre en una direcció de forma segura.

Podem definir el vector de canal  $\vec{H}$ , que tindrà tants elements com el nombre de transmissors que tenim, per cada un d'elles caracteritzarà el comportament del senyal (variació en fase) des de que es transmès fins que es rep. El vector de canal es de la forma:

$$\vec{H}(\theta) = \left[ \underbrace{(e^{jk_1\pi\cos\theta})}_{H_1} \quad \underbrace{(e^{jk_2\pi\cos\theta})}_{H_2} \quad \dots \quad \underbrace{(e^{jk_{n-1}\pi\cos\theta})}_{H_{n-1}} \quad \underbrace{(e^{jk_n\pi\cos\theta})}_{H_n} \right]^T$$

Es pot observar que els diferents elements que formen el vector de canal s'estreuen de  $C_m$ , es tracta de la variació en fase de cada un dels components.

Podem definir el vector de canal  $\vec{H}$ , que tindrà tants elements com el nombre de transmissors que tenim, per cada un d'elles caracteritzarà el comportament del senyal (variació en fase) des de que es transmès fins que es rep.

Un concepte molt important per poder aconseguir la modulació direccional és la propietat de l'ortogonalitat entre vectors. Podem definir el senyal a emetre com:

$$\vec{S} = \vec{P}X + \vec{W} \quad (7) \quad \text{On:} \quad \vec{P} = \frac{\vec{H}}{\|\vec{H}\|} \quad (8)$$

On:

- $\vec{W}$  serà un soroll artificial que crearem de forma que només en la direcció desitjada el soroll s'anul·larà i el receptor podrà recuperar correctament el senyal emès. Aleshores quan enviem el senyal pel canal de transmissió el que estarem fent essencialment serà multiplicar-lo per la matriu que modela el canal ( $\vec{H}$ ).
- $X$  es correspon als símbols a enviar pel canal
- $\vec{P}$  es pot calcular amb l'equació 8.
- $\vec{S}$  representa el senyal modulat amb DM que s'enviarà pel canal.

Quan el receptor es trobi en la direcció desitjada aleshores aquest rebrà:

$$Y_h = \vec{H}^\dagger \vec{S} = \vec{H}^\dagger \vec{P}X + \vec{H}^\dagger \vec{W} = \|\vec{H}\| X \quad (9)$$

Per contra quan el senyal s'enviï en una direcció diferent, es distorsionaren els patrons de constel·lacions i ens trobarem en el següent cas:

$$Y_Z = Z^\dagger \vec{S} = \vec{Z}^\dagger \vec{P}X + \vec{Z}^\dagger \vec{W} \quad (10)$$

On  $Z$  correspon al vector de canal en una direcció diferent a  $\theta_0$ .

Es pot observar que addicionalment a la component del senyal transmès, també s'afegeix una component addicional ( $\vec{Z}^\dagger \vec{W}$ ) que serà el soroll que s'ha afegit artificialment i que ens garantitzarà la seguretat de la modulació direccional. Aquest soroll es podrà calcular a partir la propietat d'ortogonalitat, per fer-ho caldrà calcular els vectors ortogonals a  $H$ , d'aquesta forma en la direcció desitjada el soroll s'anul·larà i en les altres no. A partir d'aquesta idea apareixen dues idees per la modulació direccional [3]:

- Modulació direccional estàtica: En aquest tipus de modulació per cada símbol enviat sempre s'afegeix un mateix soroll, és a dir per cada símbol se li associa un vector ortogonal i per un mateix símbol enviat en diferents instants de temps s'afegirà el mateix soroll.
- Modulació direccional dinàmica: En aquesta vessant per cada símbol enviat es triarà de forma aleatòria un vector ortogonal. De forma que per un mateix símbol enviat en instants de temps diferents, el soroll artificial que s'afegirà serà diferent.

## 5.3. Implementació teòrica de la modulació direccional

En aquesta secció analitzarem com s'ha implementat la modulació direccional a través de Matlab, s'ha creat un esquema de modulació direccional a través de Matlab i s'ha simulat el seu enviament pel canal per veure el comportament teòric de la DM. En la següent secció s'implementarà en el SDR i s'enviarà pel canal per analitzar un exemple de comportament pràctic en unes condicions ideals.

Per provar la implementació s'ha suposat que hi ha 5 elements radiant al receptor per un canal amb el següent vector de canal:

$$\vec{H}(\theta) = \left[ \underbrace{(e^{j2\pi\cos\theta})}_{H_1} \quad \underbrace{(e^{j\pi\cos\theta})}_{H_2} \quad \underbrace{(e^{j0})}_{H_3} \quad \underbrace{(e^{-j\pi\cos\theta})}_{H_4} \quad \underbrace{(e^{-j2\pi\cos\theta})}_{H_5} \right]^T \quad (11)$$

S'ha calculat inicialment el vector de canal quan aquest radia en totes les direccions, de  $0^\circ$  a  $180^\circ$  i s'ha definit la direcció desitjada,  $\Theta_0 = 45^\circ$ . Primerament s'ha definit una funció que donades la matriu H per totes les direccions i la direcció desitjada calcula P i una matriu base dels vectors ortogonals a  $\vec{H}$ .

Seguidament s'ha simulat el comportament quan s'envia la cadena de símbols [1, 2, 3, 4] modulats amb una QPSK seguint els passos especificats anteriorment, primerament calcular el senyal a transmetre amb l'equació (7).

On X correspon a un vector dels símbols que s'envien, P la matriu calculada anteriorment amb l'equació 8 i s'hi afegeix un soroll W, aquest soroll W es calcula com una combinació lineal de la base ortogonal de H, els pesos associats a cada un dels vectors es fa de forma aleatòria per cada un dels símbols que s'envien, de forma que s'ha generat modulació direccional dinàmica. Els pesos es generen seguint una distribució Gaussiana de variància 1, de forma que la diferència de potències que s'aplica a cada transmissor no serà molt elevat.

Finalment el senyal S es simula que s'envia pel canal, primerament s'ha comprovat que quan s'envia en la direcció desitjada  $\Theta_0 = 45^\circ$ , el senyal rebut es correspon al senyal enviat formant una QPSK perfecte com es pot apreciar en la figura 64.

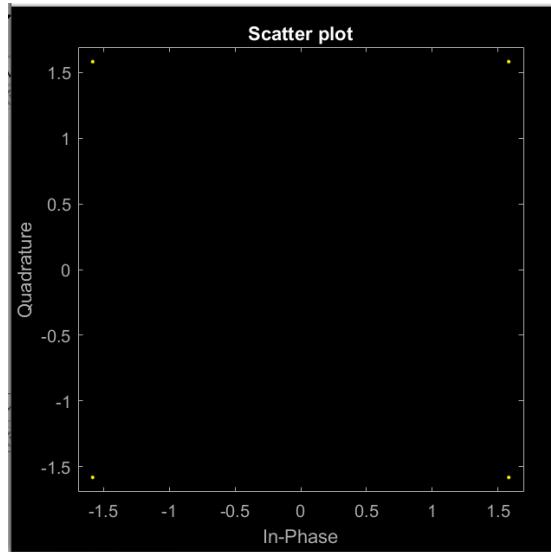


Figura 64. Constel·lació rebuda en la direcció desitjada.

Per altra banda, quan s'envia en una direcció diferent a la desitjada no es rep la constel·lació de la QPSK, a continuació es poden veure el la constel·lació del senyal rebut en dos exemples a  $80^\circ$  i  $120^\circ$  respectivament:

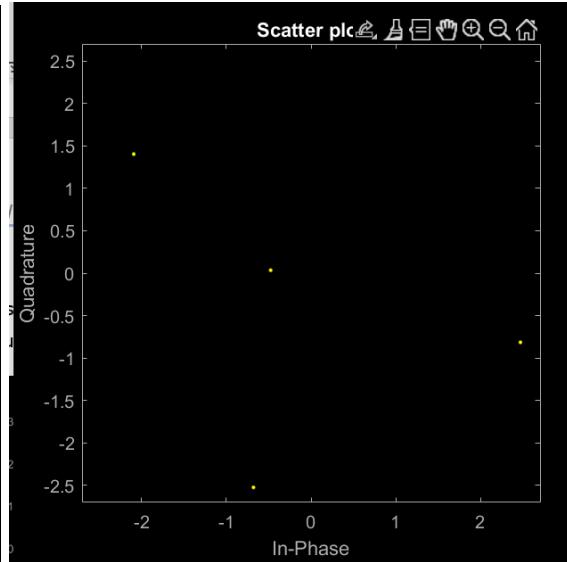
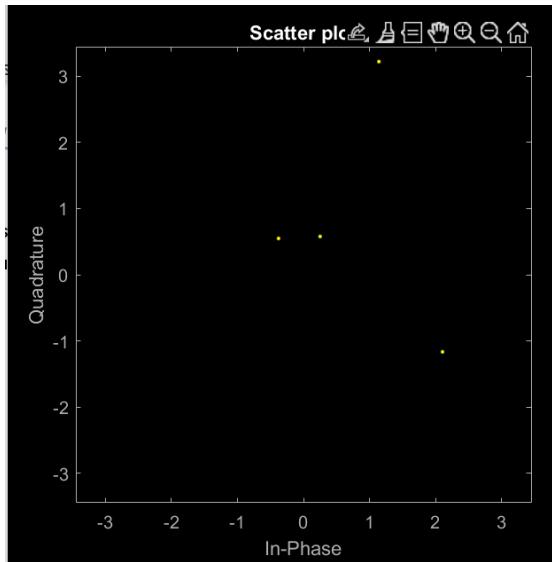


Figura 65. Constel·lació rebuda  $80^\circ$  amb DM. Figura 66. Constel·lació rebuda  $120^\circ$  amb DM.

Per fer un anàlisi més detallat també es pot veure a continuació el resultat del mòdul obtingut quan s'envia en tots els angles, en un quadrat es pot veure marcat el mòdul en la direcció desitjada  $\Theta_0$ , on es pot veure que el mòdul del senyal rebut per qualsevol bit enviat coincideix com es pot veure en la figura 67.

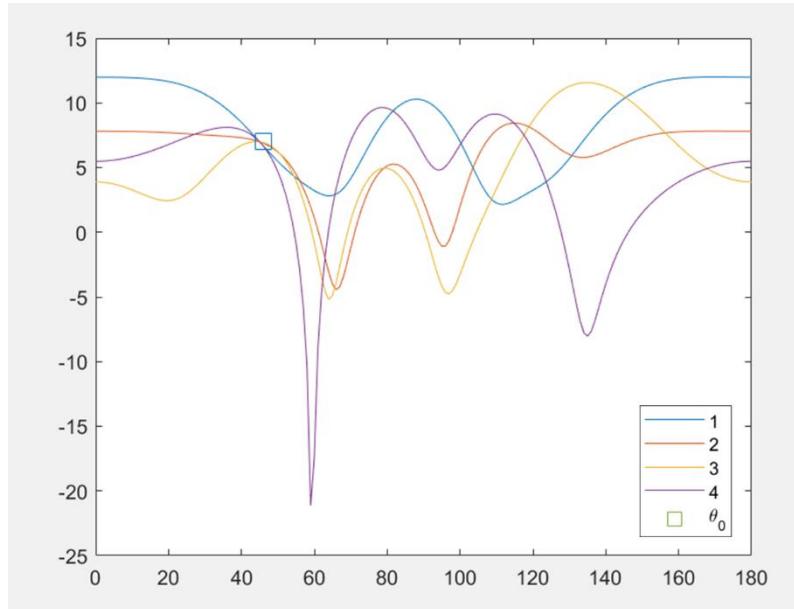


Figura 67. Mòdul del senyal rebut amb DM.

També ens podem fixar amb la fase del senyal rebut per cada un dels símbols depenen de l'angle en que s'envia en la figura 68.

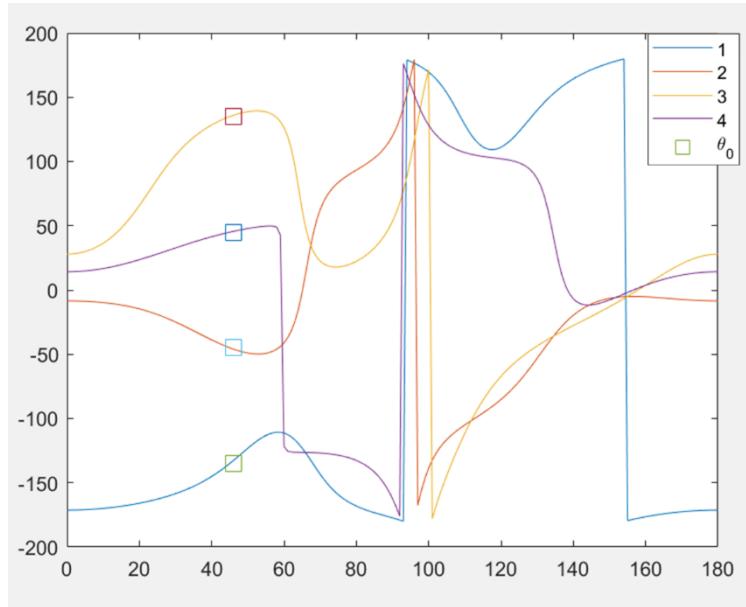


Figura 68. Fase del senyal rebut amb DM.

Es pot identificar 4 fases diferents en la direcció desitjada depenen del bit enviat pel canal.

Finalment observarem la constel·lació rebuda en funció de l'angle en que s'envia la informació, en aquest cas s'ha marcat amb un quadrat els símbol rebut quan s'envia per la direcció desitjada, quan s'envia en una direcció de 90º es pot veure marcat amb

circumferències i quan s'envia en una direcció de  $135^\circ$  amb un asterisc, es pot veure fàcilment que només en la direcció desitjada la constel·lació de la QPSK es pot identificar:

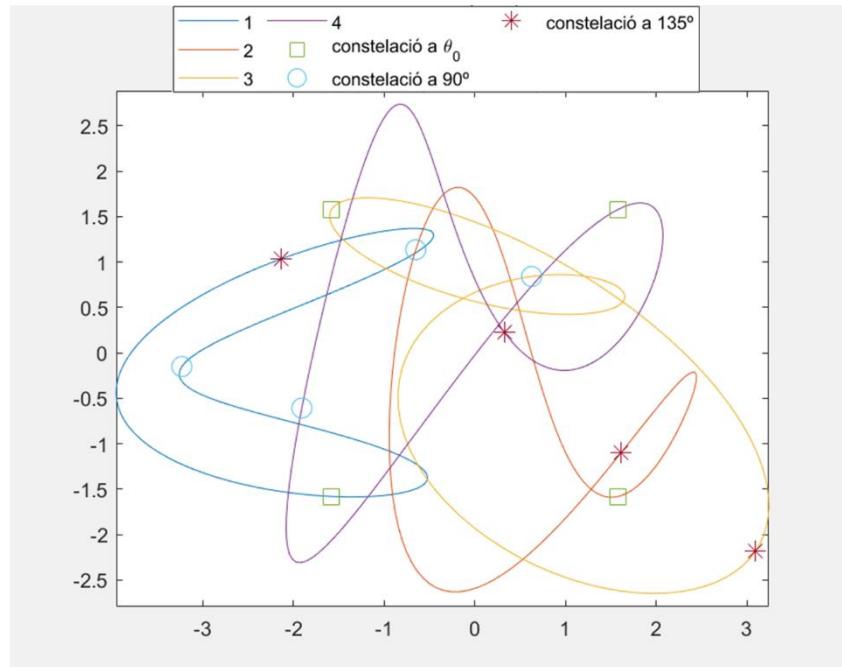


Figura 69. Variació símbols rebuts respecte direcció d'emissió amb DM.

D'aquesta forma només podrà recuperar el senyal aquell usuari situat en la direcció desitjada.

A continuació s'ha enviat la següent seqüència de símbols pel canal: [1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4], d'aquesta forma es podrà observar com respon la modulació quan s'envia varades vegades el mateix símbol. Primer veurem la constel·lació rebuda en aquesta situació quan s'envia en la direcció  $\Theta_0$ :

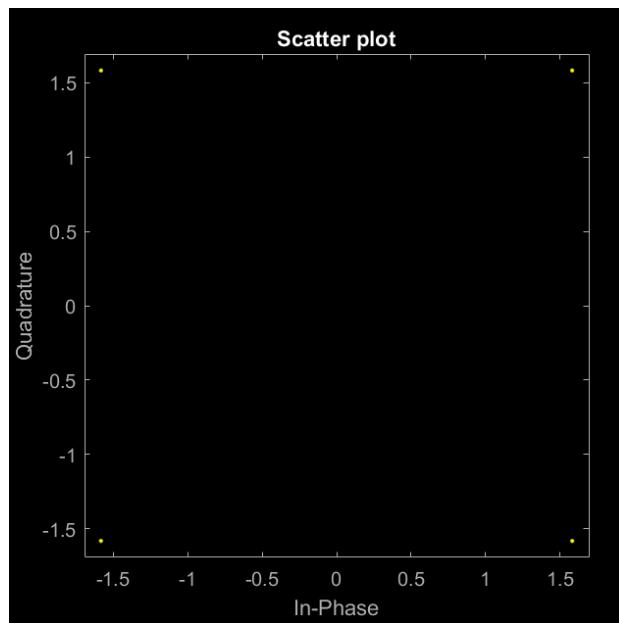


Figura 70. Constel·lació rebuda en la direcció  $\Theta_0$  amb DM.

Es pot observar en la figura 70 que la constel·lació QPSK rebuda es pot distingir clarament i té una forma sense distorsió.

També observarem la constel·lació quan s'envia la mateixa seqüència en diferents direccions que la desitjada, concretament a  $80^\circ$  i  $120^\circ$  respectivament (Figures 71 i 72).

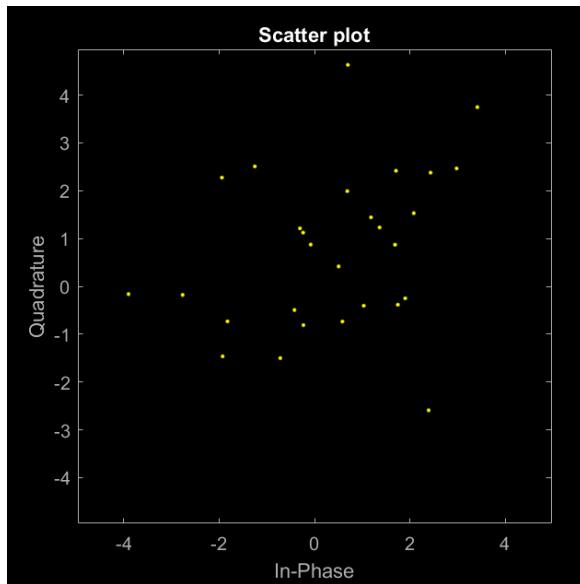


Figura 71. Constel·lació rebuda en la direcció  $80^\circ$  amb DM.

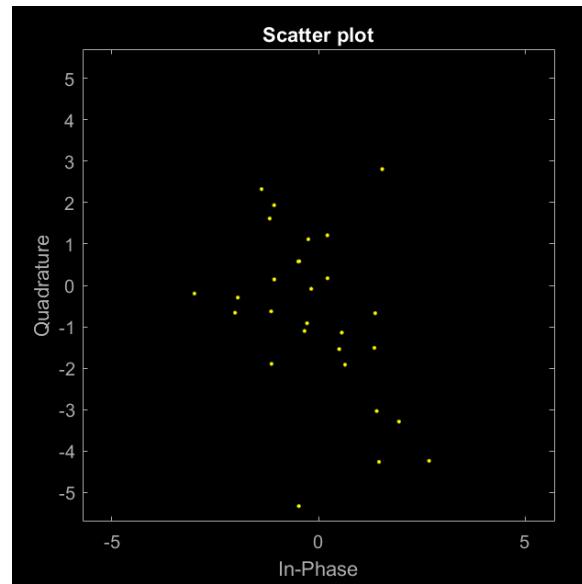


Figura 72. Constel·lació rebuda en la direcció  $120^\circ$  amb DM.

Es pot observar que en ambdós casos no es pot distingir la constel·lació QPSK ni tampoc es pot distingir entre els diferents símbols enviats.

També ens fixarem amb els mòduls obtingut quan s'envia en tots els angles, en un quadrat es pot veure marcat el mòdul en la direcció desitjada  $\Theta_0$ , on es pot veure que el mòdul del senyal rebut per qualsevol símbol enviat coincideix en aquest punt, mentre que per la resta canvia. També es pot apreciar que tots els símbols enviats, encara que es tracti del mateix símbol enviat en un instant de temps diferents, el mòdul es different per la resta d'angles (figura 73).

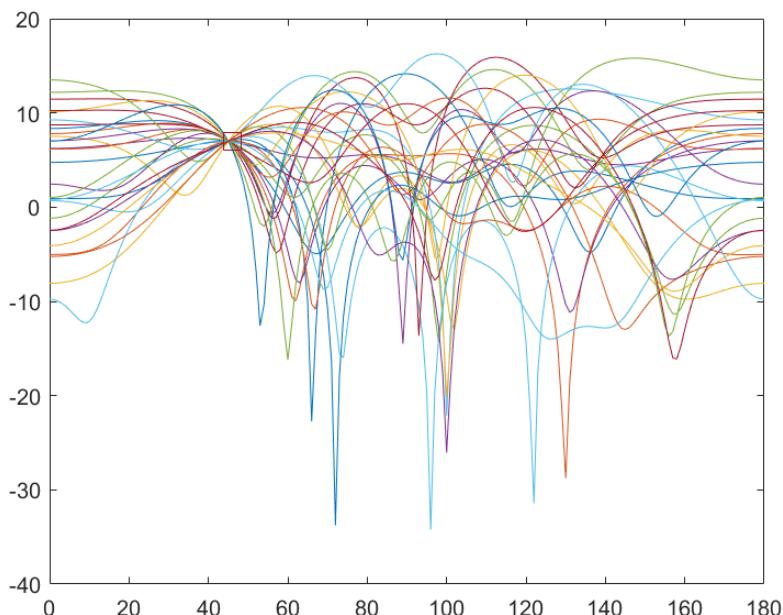


Figura 73. Mòdul del senyal rebut segons la direcció amb DM per seqüència de símbols.

També ens podem fixar amb la fase del senyal rebut per cada un dels símbols depenen de l'angle en que s'envia (figura 74).

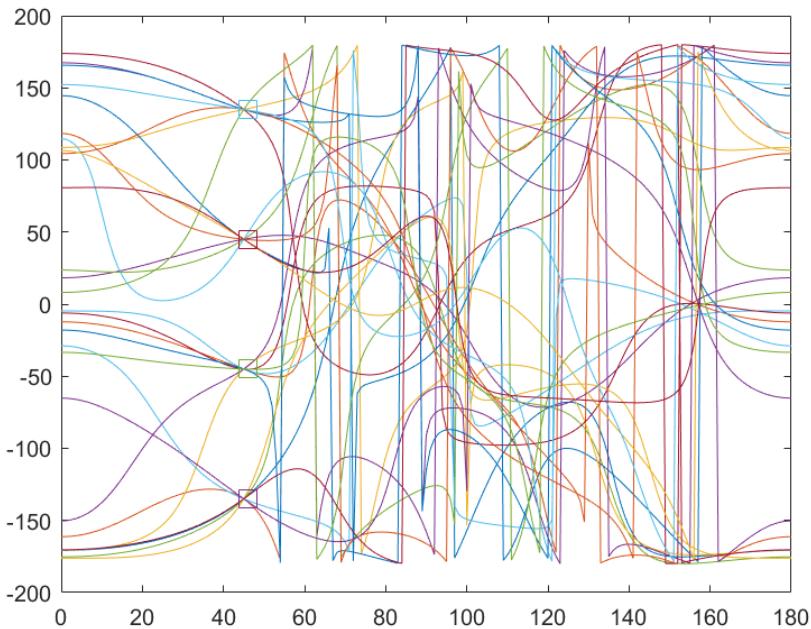


Figura 74. Fase del senyal rebut segons la direcció amb DM per seqüència de símbols.

Es pot apreciar que per cada símbol enviat, els mateixos símbols en l'angle desitjat produeixen una mateixa fase, generant 4 fases diferents una per cada un dels diferents símbols enviats, mentre que en la resta d'angles de transmissió no es pot distingir que per un mateix símbol enviat en diferents instants de temps es pugui identificar la mateixa fase.

### 5.3.1. Anàlisi de rendiment

Per tal de donar un sentit més pràctic a tot el procés d'enviament de dades, s'ha fet de forma que en la transmissió s'emeti una cadena de text, i posteriorment aquesta es recupera.

Per fer-ho, en la transmissió, primer es defineix la cadena de text que es vol emetre, aquesta converteix cada un dels caràcters a enviar en un codi binari corresponent al codi ASCII del caràcter i aquests es converteixen en els símbols a enviar seleccionant de dos en dos bits.

En el procés de recepció, primer cal definir un decisor que permeti per cada símbol rebut l'associï a un símbol dels existents (1, 2, 3 o 4). Hem definit dos decisors diferents un més complex que l'altre, aquests decisors estaven entrenats de forma ideal suposant que inicialment que les primeres dades enviades pel canal són conegeudes de forma que es pot entrenar el decisor:

- L'altre decisor es basarà en maquines de vector de suport (SVM), que es una eina utilitzada en aprenentatge computacional que intenta separar els símbols d'entrenament amb 4 vectors intentant maximitzar l'espai entre els diferents vectors de decisió. Posteriorment a partir d'aquests vectors classificarà les dades que es volen classificar.
- Un decisor es basarà amb el conjunt d'entrenament recollir tots els símbols amb un mateix valor i fer la mitjana per cada un d'ells creant un centre per cada un d'ells. Posteriorment, per cada símbol rebut buscarem la distància euclidiana a cada un dels centres i aquella a la que més s'aproxima serà el símbol que es predirà que s'ha emès.

Posteriorment a passar pel decisor, es farà el procés invers a la transmissió es convertiran els símbols ajuntant-los de 4 en 4 i a partir del codi ASCII en la cadena de text rebuda.

Quan s'incorpora per exemple el decisor basat amb centroides amb la modulació direccional, si seleccionem l'angle d'enviament desitjat a  $45^\circ$  i comprovem per la cadena enviada "cadena de text" obtenim els següents resultats en els diferents angles d'enviament:

Angle de transmissió	Cadena rebuda	Símbols rebuts correctament
$200^\circ$	"waeazq0\$d`1e=u"	38
$250^\circ$	"ru\$\$~` t%%pdxt"	41
$300^\circ$	"b de*a0`u u`8t"	44
$350^\circ$	"cadeo` `e` utxd"	48
$400^\circ$	"cadena de text"	56
$450^\circ$	"cadena de text"	56
$500^\circ$	"cadena de text"	56
$550^\circ$	"b`d\$na(`e äe d"	46
$600^\circ$	"Q ef-I hu v)x¥"	37
$650^\circ$	"aTu{w7ug0y¥``&"	29
$700^\circ$	"^• þ BçpNk°5(S"	16

Taula 2. Cadena de text rebuda i símbols rebuts correctament quan s'emet "cadena de text".

Es pot veure que en la direcció desitjada i en aquelles més properes si es pot distingir la cadena de text, però a mesura que ens allunyem d'aquesta direcció la cadena de text rebuda és més diferents i el nombre de símbols rebuts correctament també disminueix.

Si provem a emetre els mateixos símbols en totes les direccions i entrenant en cada direcció amb un model de centroides, quan s'emeten 100.000 símbols i mirem el percentatge de símbols rebuts correctament de forma teòrica en cada direcció obtindrem un gràfic similar al de la figura 75.

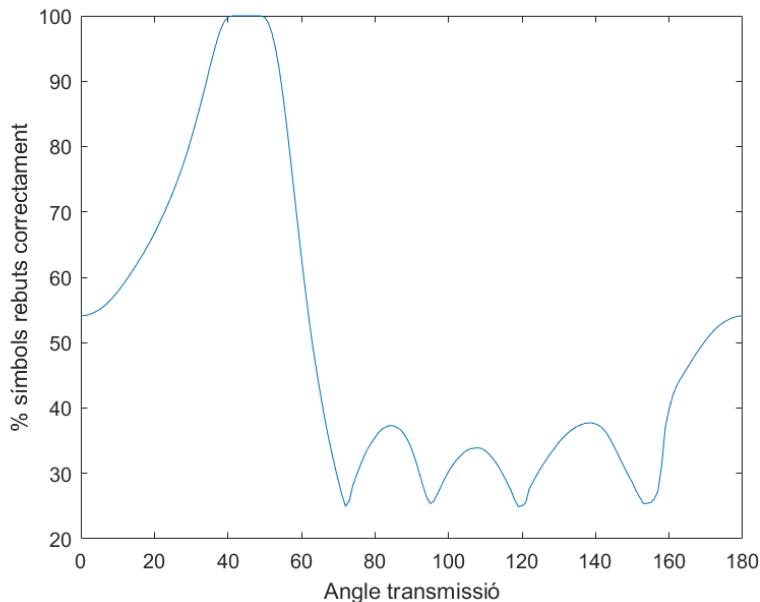


Figura 75. Percentatge símbols rebuts correctament segons direcció d'emissió (Pes potència=1).

També un aspecte interessant és comparar depenen de la potència de soroll que afegim amb la modulació direccional, això es pot variar multiplicant per un valor escalar  $W$ , per exemple si multipliquem per 2, obtindrem una resposta com la figura 76 i si multipliquem per 0.25 obtindrem un gràfic com la figura 77.

També es poden pensar en les gràfiques de la forma inversa per tenir una noció molt més pròxima en l'àmbit de les telecomunicacions com és la SER, que es pot relacionar fàcilment amb la BER dividint-la pel nombre de bits per codificar cada un dels símbols.

En aquest estudi hem treballat amb el percentatge de correctes perquè per entendre-ho considero que és molt més visual i senzill, però per comparar amb altres sistemes serà més útil convertir-ho en la BER o la SER.

Quan analitzem la transmissió amb la DM en el sistema real, per facilitar la comparació amb altres esquemes, s'explica amb més detall la BER i SER i es fa l'anàlisi també amb la SER, per poder comparar amb més facilitat amb altres sistemes existents (Apartat 5.4).

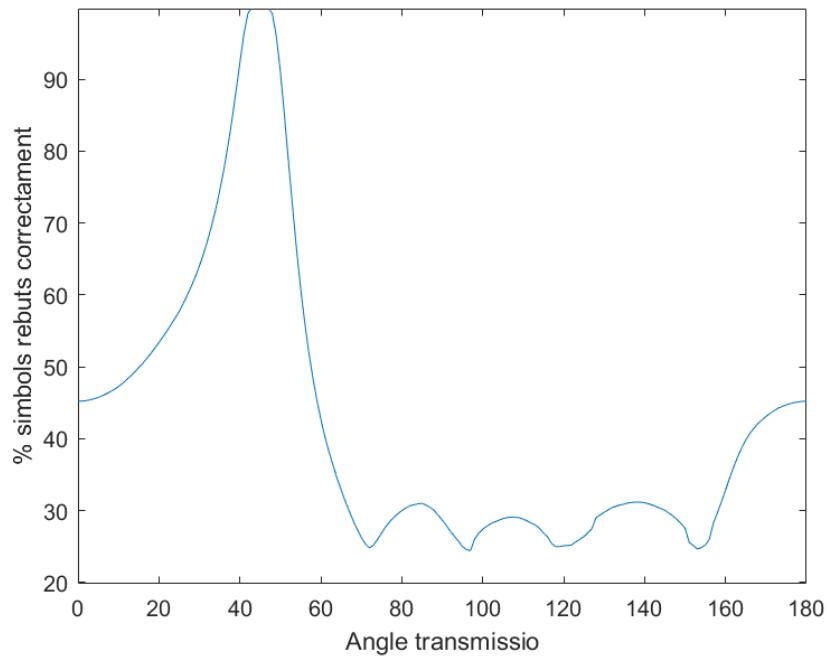


Figura 76. Percentatge símbols rebuts correctament segons direcció d'emissió (Pes potència=2).

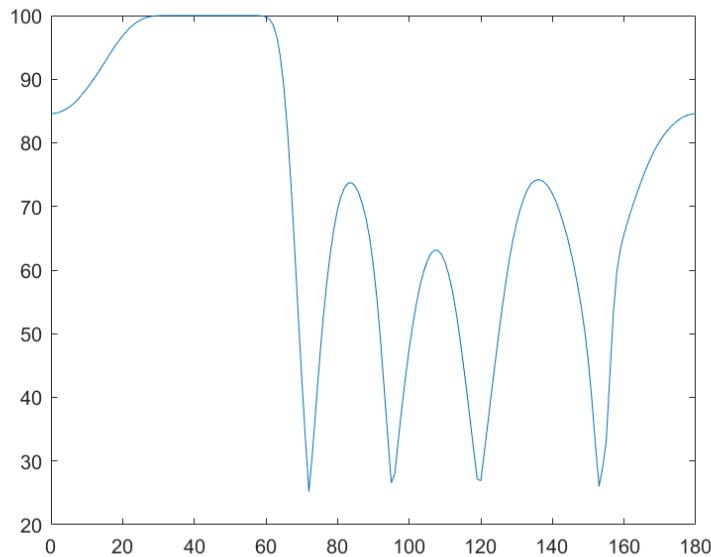


Figura 77. Percentatge símbols rebuts correctament segons direcció d'emissió (Pes potència=0.25).

Es pot observar que quan major és la potència del soroll, hi ha més errors en els símbols rebuts en direccions diferents a la desitjades. Es veu també que el gràfic té forma amb lòbuls i el lòbul principal on s'arriba a 100% és en la direcció desitjada, en aquest lòbul es pot observar que quan la potència és major també és més estret. Cal tenir en compte

que aplicar una potència tindrà un cost major, és per això que caldrà establir un balanç entre el cost i la potència del soroll.

## 5.4. Verificació experimental

Un cop comprovat el seu funcionament teòric a través de Matlab, passarem a avaluar el seu comportament en el SDR dins la sala anecoica per avaluar com difereix el seu comportament respecte el teòric. Primer definirem l'angle  $\Theta_0$  com  $0^\circ$  i comprovarem que en aquesta direcció la modulació direccional tingui comportament esperat. Podrem comprovar el seu comportament de forma que si transmetem un senyal modulat amb una QPSK, a la sortida també es pugui apreciar la constel·lació pròpia de la QPSK.

Primer caldrà calcular el vector de canal per la direcció desitjada ( $\vec{H}$ ), la qual cosa es pot fer amb l'algoritme de calibració que ens donarà per cada transmissor la variació en fase mitja entre el receptor i el transmissor. Amb el vector de canal ( $\vec{H}$ ), amb l'equació ( $\vec{P}$ ) i calculant la seva base ortogonal podrem obtenir els vectors  $\vec{P}$  i  $\vec{W}$  respectivament,  $\vec{W}$  serà qualsevol combinació lineal de la base ortogonal calculada.

Generarem una trama amb 256 símbols (on els dos primers seran 0s) generats de forma aleatòria modulats amb una QPSK i la modularem amb la DM seguint l'equació 7 a partir dels paràmetres calculats prèviament, per aquest pas els pesos generats de forma aleatòria estaran multiplicats per 0.25, això es pot entendre com la potència de soroll que afegirem al sistema. Quan s'emet la trama pel canal en la recepció es pot observar la constel·lació de la figura 78, de forma que es pot comprovar que en la direcció  $\Theta_0$  les dades s'estan rebent els mateixos transmesos gràcies a l'equació 9.

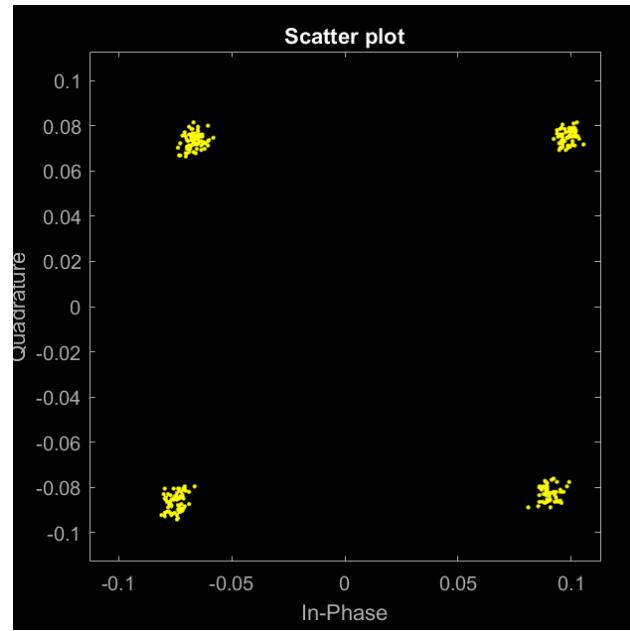


Figura 78. Constel·lació rebuda amb DM amb el SDR direcció  $\Theta_0$ .

El següent pas es comprovar que un usuari situat en una altre direcció rebrà les dades amb distorsió afegida degut a l'equació 7. Per fer-ho seguirem amb el mateix procediment a l'anterior generant els símbols a emetre i s'utilitzarà el mateix vector  $\vec{P}$  i la mateixa base ortogonal. En la recepció si mirem per un angle  $30^\circ$  es pot observar que la constel·lació QPSK no es pot identificar.

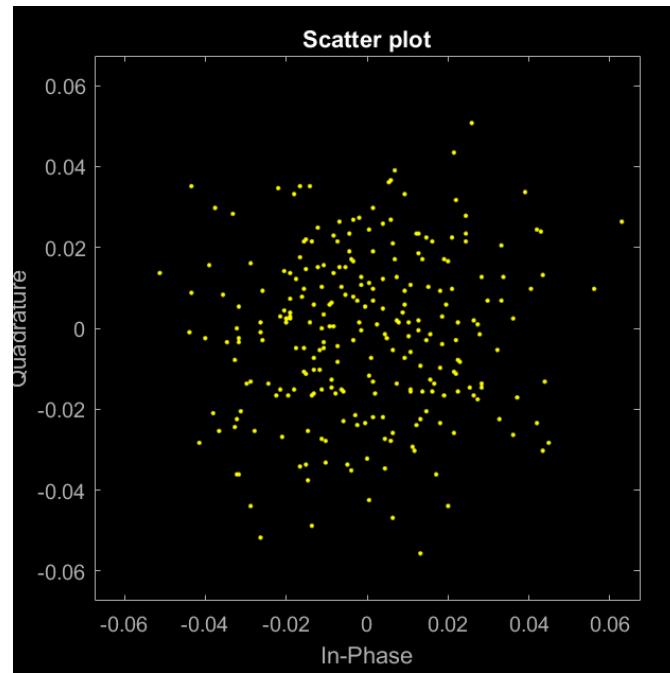


Figura 79. Constel·lació rebuda amb DM amb el SDR direcció  $30^\circ$ .

Amb aquests exemples comprovats podem veure que la modulació direccional està distorsionant el senyal rebut en totes les direccions excepte en  $\Theta_0$ , on la constel·lació es manté clarament definida, la qual cosa verifica el correcta funcionament de la implementació. A continuació passarem a analitzar amb més detall el comportament de la DM pels diferents angles d'emissió i variant alguns paràmetres.

En la figura 80 es pot veure el diagrama de radiació del transmissor a través dels diferents angles de l'espai en que s'ha radiat separats entre ells en  $5^\circ$ . Es pot observar que en l'angle 0 es troba el lòbul principal i en els angles  $70^\circ$  i  $-70^\circ$  hi ha lòbuls secundaris.

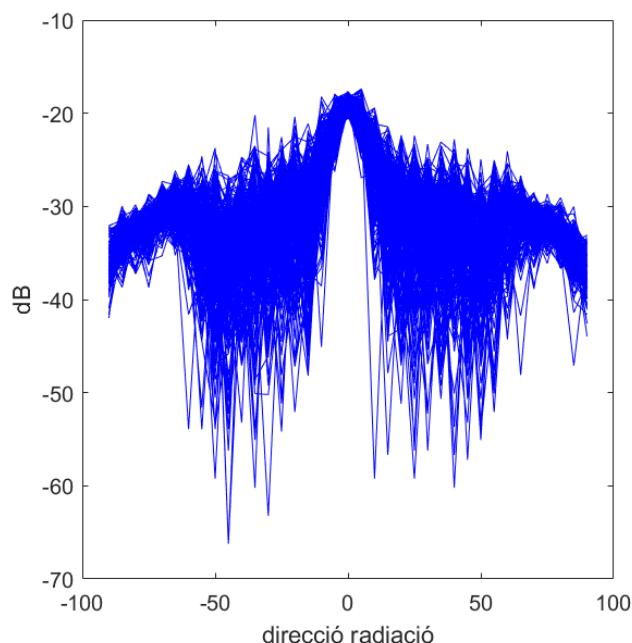


Figura 80. Potència senyal rebut amb DM (pes potència soroll=0.25).

A la figura 81 es pot veure la fase dels símbols rebuts, es pot veure que en la direcció d'emissió de  $0^\circ$  es pot veure les 4 fases clarament distingides corresponent els símbols possibles de la QPSK. En angles propers també es pot apreciar les fases dels símbols i en angles que corresponen als lòbuls secundaris també es poden distingir 4 fases, tot i que en aquests alguns símbols són erronis:

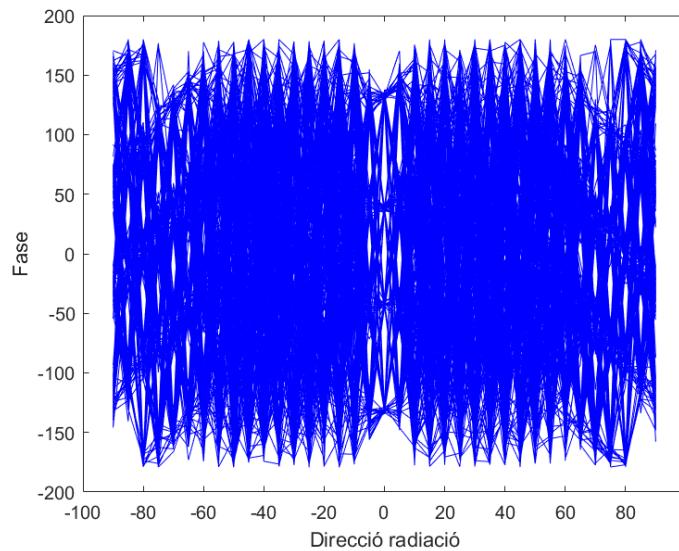


Figura 81. Fase senyal rebut amb DM (pes potència soroll=0.25).

Un dels paràmetres que es pot variar és la potència de soroll que s'introduceix, això es pot implementar afegint un multiplicador als pesos aleatoris que generen la combinació, en el codi que hem creat això es pot modificar en la línia 95 del codi que es troba en l'annex "Codi transmetre modulació direccional" (Annex 8.1.10). En l'anterior figura el pes que s'havia especificat és de 0.25.

Quan s'introduceix més potència de soroll, per exemple si enlloc de 0.25 especifiquem un pes de 1, quan observem el diagrama de radiació, que es pot observar a la figura 82, és pot observar un increment del soroll en totes les direccions. A més també es veu en el diagrama de fases que no és tan fàcilment distingibles les 4 fases.

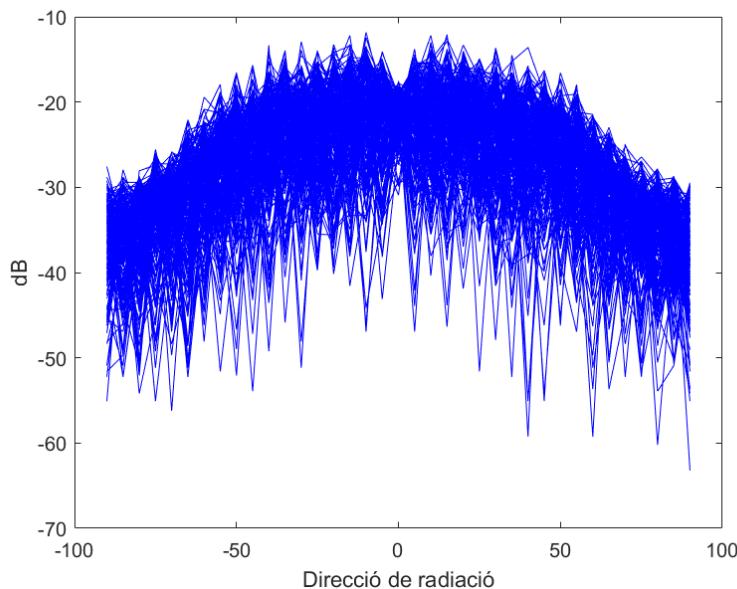


Figura 82. Potència senyal rebut amb DM (pes potència soroll=1).

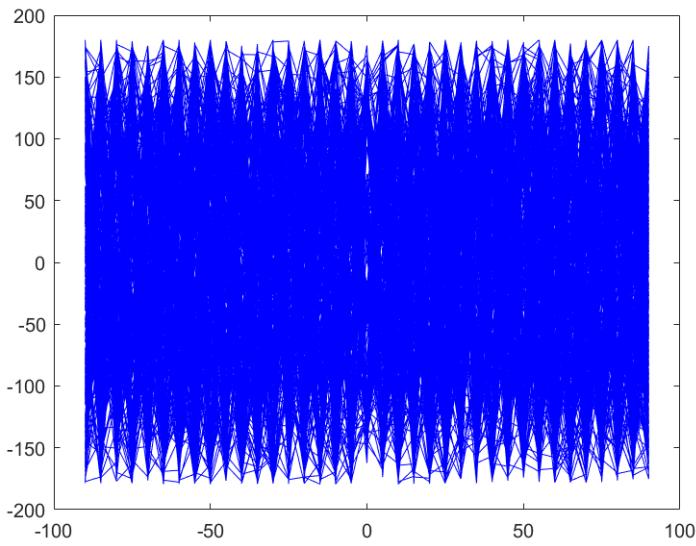


Figura 83. Fase senyal rebut amb DM (pes potència soroll=1).

Per altra banda si reduïm el pes a 0, aleshores el soroll en el diagrama de radiació és molt baix i es pot distingir clarament el diagrama de radiació de la mateixa manera amb les 4 diferents fases de la modulació de la QPSK, en aquest cas és com si estiguéssim transmeten dades sense la modulació direccional, per això es pot distingir clarament les fases dels símbols emesos. Es pot observar el diagrama de radiació i la fase del senyal rebuda en les figures 84 i 85 respectivament.

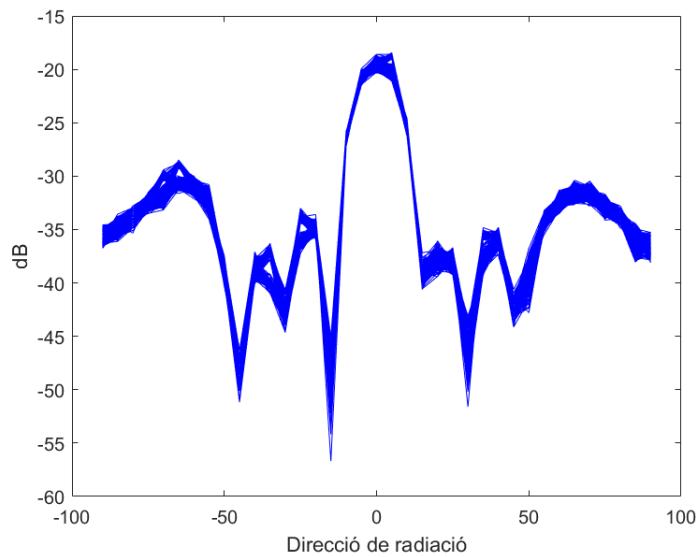


Figura 84. Potència senyal rebut amb DM (pes potència soroll=0).

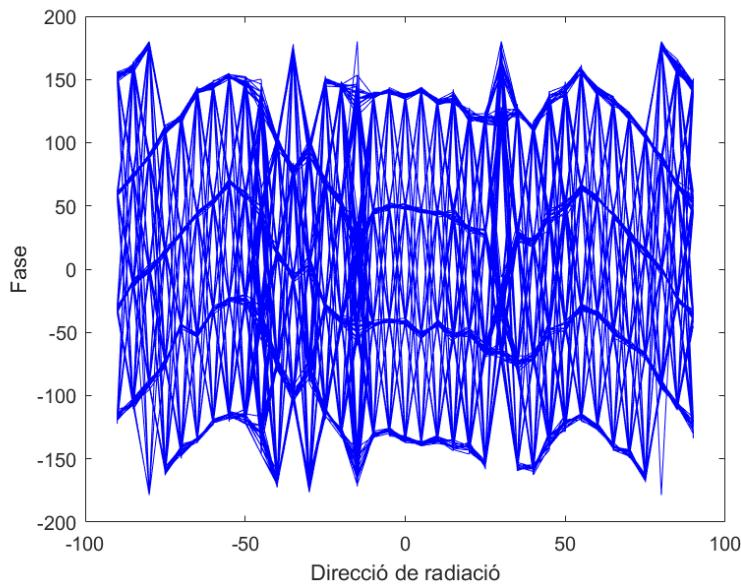


Figura 85. Fase senyal rebut amb DM (pes potència soroll=0).

Es pot notar que el soroll té un efecte molt major quan els dB rebuts són menors, és a dir en les direccions on el diagrama de radiació mostra una potència de recepció molt menor, el soroll té un major impacte i pel receptor serà molt més difícil distingir el símbol QPSK emès.

Es pot observar que quan més soroll estem afegint aleshores les fases són menys identificables en els gràfics i el soroll té una influència major en totes les direccions, però en la direcció desitjada la influència d'aquest es molt menor que en la resta. De forma que els símbols rebut seran molt més difícils de distingir en totes les direccions menys la desitjada, on no afectarà tan, a mesura que augmentem el soroll, de forma que ens oferirà més seguretat en la transmissió dificultant la identificació de símbols emesos a les persones que interceptin el senyal.

En els annexos es pot veure es pot trobar el codi per la transmissió de la DM amb el SDR i el seu ànalisi, per fer-ho s'han utilitzat els annexos 8.1.10 i 8.1.3 per fer la transmissió en diferents angles amb el controlador.

Per altra banda si mirem el percentatge de símbols que es recuperen correctament quan fem una transmissió amb el SDR treballant amb un pes per la potència del soroll afegit de 1 i considerem l'angle desitjat per l'emissió en  $0^\circ$ , obtenim gràfiques com les figures 86 i 87, que són respectivament pels decisors amb l'algoritme de centroides i l'SVM.

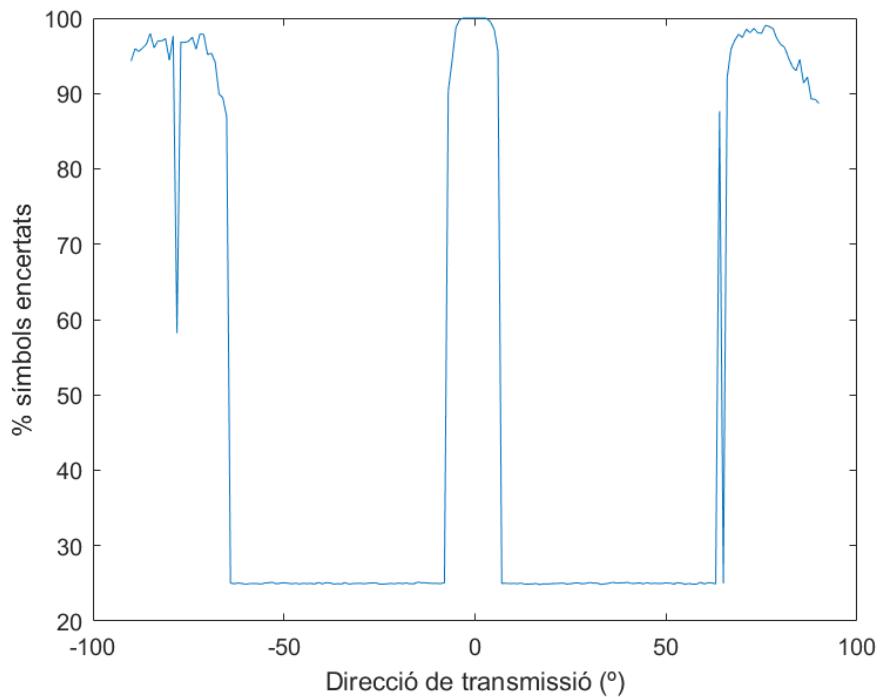


Figura 86. Percentatge símbols rebuts correctament amb DM  
i decisor centroides (pes potència soroll=1).

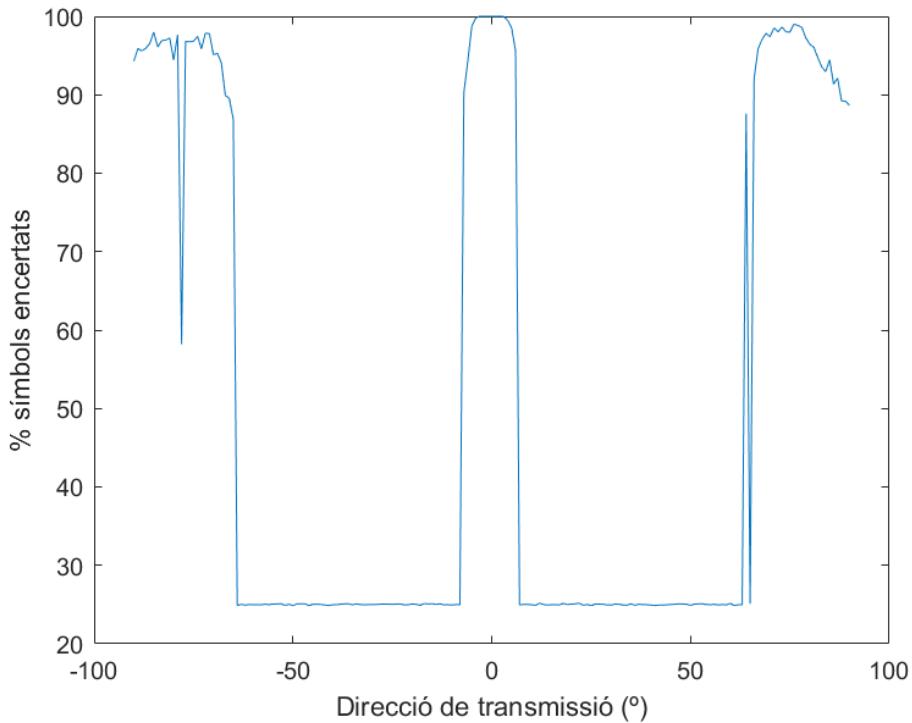


Figura 87. Percentatge símbols rebuts correctament amb DM  
i decisor SVM (pes potència soroll=1).

Els dos transmissors donen un rendiment bastant similar. Es pot observar que en la direcció desitjada totes els bits s'estan recuperant tots els bits transmesos, però també en direccions similars també s'estan recuperant, a més en direccions properes als -70º i 70º corresponents als lòbuls secundaris també s'estan recuperant la majoria de símbols correctament, arriba a recuperar-se el símbol emès correctament en algunes direccions en un 99% dels casos. En la resta de direccions es recupera el símbol emès aproximadament en un 25% dels casos, que realment com estem enviant 4 símbols de forma aleatòria es pot entendre com que s'està recuperant un símbol de forma aleatòria.

Això és degut a que el soroll afegit en aquestes direccions, on es troben els lòbuls secundaris, el soroll afegit per la modulació direccional no és suficient per tal d'alterar suficientment el símbol rebut. De forma que per aconseguir una major seguretat en aquestes direccions caldria augmentar el soroll per tal de reduir el percentatge.

També podem fer el mateix anàlisi des del punt de vista de la SER (Symbol Error Rate), que es pot calcular com:

$$SER = \frac{1}{n} \sum_1^n 1_{\{x_i \neq x_{0,i}\}} \quad (11)$$

És a dir, en el nostre cas compte el nombre d'elements que es reben correctament i els divideix pel nombre d'elements totals. La BER es pot relacionar amb la SER com:

$$BER = SER/M \quad (12)$$

On M representa el nombre de bits per codificar cada un dels símbols.

Les gràfiques de la SER depenen de la direcció de transmissió són com la següent, pels decisors basats en centroides i el SVM respectivament en les figures 88 i 89:

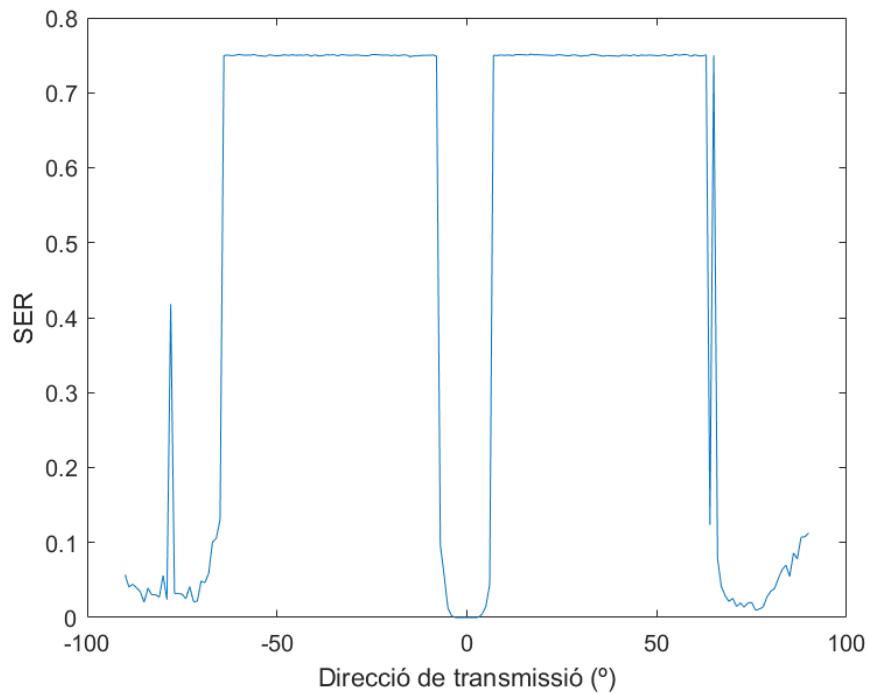


Figura 88. SER pel senyal amb DM i decisor centroides (pes potència soroll=1).

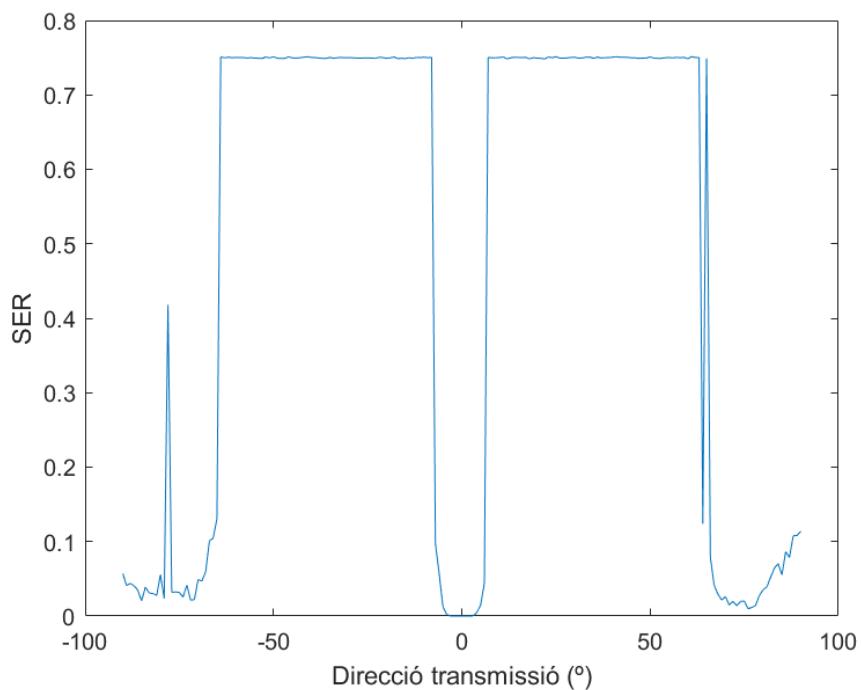


Figura 89. SER pel senyal amb DM i decisor SVM (pes potència soroll=1).

Es pot veure com bàsicament són com les gràfiques 86 i 87 però en la inversa. No es cometen errors en la transmissió en la direcció desitjada ( $0^\circ$ ), aproximadament entre els  $-70^\circ$  fins als  $-10^\circ$  i entre els  $10^\circ$  i els  $70^\circ$  la tassa d'error es troba en un 0.25 i en direccions superiors, degut a que és on es troben els lòbuls secundaris del diagrama de radiació hi ha una gran quantitat d'encerts.

Per calcular en termes de la BER

En els annexos 8.1.11 i 8.1.12 el codi corresponent per recuperar els símbols, amb els decisors de centroides i SVM respectivament, de la transmissió feta anteriorment en el codi 8.1.10 pels diferents angles amb el codi del controlador 8.1.3 i comparar-los amb els símbols emesos per calcular el percentatge de símbols rebuts correctament. Per calcular la SER les úniques variacions en el codi necessàries són no passar-ho en un percentatge i enllloc de comptar el nombre de mostres que es reben correctament comptar el nombre que són incorrectes.

L'anàlisi s'ha realitzat amb 400.000 símbols dels quals els 10.000 primers s'han utilitzat per l'entrenament dels decisors.

## 6. Conclusions i línies futures

En aquest projecte he recopilat informació general sobre els SDR i concretament he mirat amb més detall l'AD9361 amb el que posteriorment hem treballat. Primer he fet una etapa on m'he familiaritzat amb la forma d'interactuar amb aquest SDR i posteriorment he adaptat la interacció amb l'AD9361 amb una Toolbox de Matlab més actual. Per fer aquesta familiarització ens hem basat en un codi preexistent que permetia calcular el vector de canal, a partir del qual posteriorment es podia calibrar el sistema.

Per altra banda, he estudiat el funcionament de la modulació direccional i he fet uns primers estudis sobre ella a nivell teòric, estudiant el seu comportament variant la potència iavaluant els símbols que es rebien correctament en les diferents direccions. Finalment, he combinat els progressos en ambdues per avaluar sota unes condicions pròximes a les ideals com funcionava la DM.

D'aquesta forma si comparem el que s'ha assolit respecte als objectius marcats, s'han aconseguit tots. Respecte a la metodologia planificada per dur a terme el projecte no s'ha seguit, el que ha causat algunes complicacions en el treball.

Al llarg del treball hem pogut beneficiar-nos de la facilitat de reconfigurar el SDR i modificar-lo per tal de corregir amb errors, com per exemple amb el guany de 30 dB que s'ha modificat a 32 dB, per tant, hem vist de primera mà els avantatges dels sistemes SDR.

Hem vist com crear un esquema de transmissió de dades més segur amb la DM i com implementar-lo amb el SDR, alguns dels aspectes més importants que ens hem adonat durant la implementació d'aquest model són:

- El balanç entre la potència del soroll, i per tan el cost de la modulació direccional, i la seguretat que el sistema ens està aportant. És necessari fer un estudi per definir un compromís entre aquests aspectes.
- La importància de tindre en compte el desfase entre els diferents transmissors de cara a funcionalitats com la DM.
- La modulació direccional dinàmica aporta major seguretat que l'estàtica i el cost per aconseguir-la és relativament baix.
- La modulació direccional aporta una dificultat afegida per tal de vulnerar la seguretat d'una transmissió, però és un esquema que es pot arribar a vulnerar, en especial si es defineix una potència de soroll massa baixa, i en lòbuls secundaris es pot identificar sense gran dificultat, ja que el soroll té un impacte menor.

- És important si es vol crear un esquema segur tenir en compte el diagrama de radiació i identificar el comportament de la DM en totes les direccions de l'espai, no només basar-se en aspectes teòrics ni en unes direccions definides a l'atzar.

Aquest treball m'ha permès aprendre molt respecte a la modulació direccional i els sistemes SDR, els quals són molt interessants per la seva facilitat per reconfigurar-los i la facilitat avui en dia que ofereixen eines com les Toolboxes de Matlab per interactuar amb elles, són especialment molt útils per propòsits educatius perquè es poden utilitzar i configurar-los per moltes funcionalitats diferents. A més, he vist com esquemes de transmissió com la modulació direccional afegeixen seguretat en la transmissió de dades, la qual cosa és una característica de gran interès avui en dia, ja que els atacants busquen moltes maneres de vulnerar-la, d'aquesta forma tot i que esquemes d'encriptació puguin fallar, degut per exemple a errors humans, tenim una transmissió de dades amb uns mínims de seguretat. M'ha servit per tenir una millor visió dels camps de treball dels enginyers de telecomunicació.

De cara a possibles futurs treballs derivats d'aquest, es podrien fer estudis més profunds en termes de BER o EVM respecte a la potència del soroll que s'afegeix, provant també amb diferents decisors, alguns més complexos i altres més simples. També fer estudis en situacions reals, per exemple simulant el soroll de l'espai exterior o fent-lo en unes situacions menys ideals per veure com la modulació direccional es pot veure afectada. Així mateix es poden fer estudis per millorar la modulació direccional afegint el paràmetre de guany variable en el símbol transmès i veure com millora la seguretat aquest en termes de BER o EVM. Finalment, combinar aquest sistema de seguretat amb esquemes d'encriptació, com per exemple esquemes de clau pública, per implementar un sistema que aporti una gran seguretat en la transmissió de dades.

## 7. Bibliografia

- [1] T. Ulversoy, "Software Defined Radio: Challenges and Opportunities," in *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 531-550, Fourth Quarter 2010, doi: 10.1109/SURV.2010.032910.00019.
- [2] Jondral, Friedrich K. "Software-Defined Radio—Basics and Evolution to Cognitive Radio", EURASIP Journal on Wireless Communications and Networking, 2005, VL - 2005, IS - 3, SN - 1687-1499. Available: <https://doi.org/10.1155/WCN.2005.275>
- [3] Yuan Ding and Vincent F. Fusco, "A Vector Approach for the Analysis and Synthesis of Directional Modulation Transmitters", IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, VOL. 62, NO. 1, Gener 2014
- [4] G. Foreman Campins, "Control and evaluation of the ad-fmcomms5-ebz software-defined radio", Ddd.uab.cat, 2020. [Online]. Available: [https://ddd.uab.cat/pub/tfg/2020/234191/Guillem\\_Foreman\\_Campins.pdf](https://ddd.uab.cat/pub/tfg/2020/234191/Guillem_Foreman_Campins.pdf). [Accessed: 16- Mar- 2022].
- [5] "Software-defined radio - Wikipedia", En.wikipedia.org, 2022. [Online]. Available: [https://en.wikipedia.org/wiki/Software-defined\\_radio](https://en.wikipedia.org/wiki/Software-defined_radio). [Accessed: 04- May- 2022]
- [6] P. Johnson, "New research lab leads to unique radio receiver," E-Systems Team, vol. 5, no. 4, pp. 6-7, May 1985.
- [7] P. Vices, "Important Parameters for SDR Systems", Everythingrf.com, 2021. [Online]. Available: <https://www.everythingrf.com/community/important-parameters-for-sdr-systems>. [Accessed: 04- May- 2022].
- [8] "IoT Security – Part 8 (Introduction to software defined radio)", Payatu, 2020. [Online]. Available: <https://payatu.com/blog/appar/iot-security-%E2%80%93-part-8-introduction-to-software-defined-radio-sdr>. [Accessed: 04- May- 2022].
- [9] P. Vices, "What is a Software Defined Radio? - everything RF", Everythingrf.com, 2022. [Online]. Available: <https://www.everythingrf.com/community/what-is-a-software-defined-radio>. [Accessed: 04- May- 2022].
- [10] D. Jaramillo Quintanar, "Respuesta en frecuencia ideal de cuatro tipos de filtros a filtro pasabajas", Research Gate, 2017. [Online]. Available: [https://www.researchgate.net/figure/Figura-3-Respuesta-en-frecuencia-ideal-de-cuatro-tipos-de-filtros-a-filtro-pasabajas\\_fig2\\_337447960](https://www.researchgate.net/figure/Figura-3-Respuesta-en-frecuencia-ideal-de-cuatro-tipos-de-filtros-a-filtro-pasabajas_fig2_337447960). [Accessed: 10- Apr- 2022].

- [11] C. Laufer, "ABOUT RTL-SDR", Rtl-sdr.com, 2021. [Online]. Available: <https://www rtl-sdr com/tag/applications-2/>. [Accessed: 04- May- 2022].
- [12]"Advantages of SDR | Disadvantages of SDR,Software Defined Radio", Rfwireless-world.com, 2016. [Online]. Available: <https://www rfwireless-world com/Terminology/Advantages-and-Disadvantages-of-SDR-Software-Defined-Radio.html>. [Accessed: 04- May- 2022].
- [13]"What are the advantages of Software Defined Radio?", Rfcec.com, 2022. [Online]. Available: [http://rfcec.com/RFCEC/Section-3%20-%20Fundamentals%20of%20RF%20Communication-Electronics/33%20-%20TRANSCEIVER/Transceiver%20-%20A%20Software%20Defined%20Radio%20What%20are%20the%20advantages%20\(By%20ARRL%20SDR%20Working%20Group\).pdf](http://rfcec.com/RFCEC/Section-3%20-%20Fundamentals%20of%20RF%20Communication-Electronics/33%20-%20TRANSCEIVER/Transceiver%20-%20A%20Software%20Defined%20Radio%20What%20are%20the%20advantages%20(By%20ARRL%20SDR%20Working%20Group).pdf). [Accessed: 04- May- 2022].
- [14] T. Boncompte Vilaró, "A NEW VISION OF SOFTWARE DEFINED RADIO: FROM ACADEMIC EXPERIMENTATION TO INDUSTRIAL EXPLOITATION", Upcommons.upc.edu, 2012. [Online]. Available: [https://upcommons.upc.edu/bitstream/handle/2099.1/18021/teresaboncomptevilaro\\_anewvisionofsoftwaredefinedradio,fromacademicexperimentationtoindustrialexploration..pdf?sequence=1](https://upcommons.upc.edu/bitstream/handle/2099.1/18021/teresaboncomptevilaro_anewvisionofsoftwaredefinedradio,fromacademicexperimentationtoindustrialexploration..pdf?sequence=1). [Accessed: 19- May- 2022].
- [15] V. K. Garg, "Fourth Generation Systems and New Wireless Technologies", sciencedirect.com, 2007. [Online]. Available: <https://www sciencedirect com/topics/engineering/software-defined-radio>. [Accessed: 04- May- 2022].
- [16] L. Teschler, "Software-defined radio powers the IoT", Microcontrollertips.com, 2016. [Online]. Available: <https://www microcontrollertips com/software-defined-radio-powers-the-iot-tag31/>. [Accessed: 15- May- 2022].
- [17] Wiki "AD-FMCOMMS5-EBZ User Guide [Analog Devices Wiki]", Wiki.analog.com, 2021. [Online]. Available: <https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms5-ebz>. [Accessed: 16- Jul- 2022].
- [18] "Data Sheet AD9361", Analog.com. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad9361.pdf>. [Accessed: 17- Jul- 2022]
- [19] "AD9361 Reference Manual", Farnell.com. [Online]. Available: <https://www.farnell.com/datasheets/2007082.pdf>. [Accessed: 23- Jul- 2022].

- [20] "Modulación por desplazamiento de fase - Wikipedia, la enciclopedia libre", Es.wikipedia.org. [Online]. Available: [https://es.wikipedia.org/wiki/Modulaci%C3%B3n\\_por\\_desplazamiento\\_de\\_fase](https://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_desplazamiento_de_fase). [Accessed: 25- Jul- 2022].
- [21] "IIO Oscilloscope [Analog Devices Wiki]", Wiki.analog.com, 2022. [Online]. Available: [https://wiki.analog.com/resources/tools-software/linux-software/iio\\_oscilloscope](https://wiki.analog.com/resources/tools-software/linux-software/iio_oscilloscope). [Accessed: 04- May- 2022].

## 8. Annexes

### 8.1. Codi

#### 8.1.1. Codi anterior

```
1 %% Transmitre el senyal
2 %Generar els simbols a enviar a través del canal
3
4 fs=30.72e6;%Sampling frequency
5 Ts=1/fs;%Sampling period
6 n_simb=1024;% # symbols per frame
7 l_s=10; % # samples in a symbol
8 nframe=15;% # frames
9
10 M=4; %QPSK = 4, 16-QAM = 16
11 %Generem els axis dels simbols a enviar
12 X=linspace(-sqrt(2)/2,sqrt(2)/2,sqrt(M));% Generem 2 (^4) punts
13 entre -^2/2 i ^2/2
14 Y=X;
15 [YY,XX]=meshgrid(Y,X); %creem els simbols de la QPSK (constelació
16 4)
17 simb_TX=XX+1i*YY;simb_TX=simb_TX(:).';%Available symbols (symbols
18 disponibles)
19
20 lframe=n_simb*l_s;% length frame in samples = (n simb/frame)*
21 (mostres/simbol)
22 t_series=[0:(nframe-1)]*lframe*Ts;%time frames (marca el temps
23 inici de cada serie enviada)
24 inputchannel=lframe; % porque??? (dades del simulink ad9361: Input
25 data channel size [samples])
26 outputchannel=lframe; % porque??? (dades del simulink ad9361:
27 Output data channel size [samples])
28 stop_time=nframe*lframe*Ts;%Simulink simulation time (n_frames*
29 longitud (samples)/frame* long(valors)/sample)
30
31 %Initialize channels (inicialitzem transmissors TXA I TXB
32 (Simulink))
33 ch_1i_data=zeros(lframe,1,nframe); % TXA
34 ch_1q_data=zeros(lframe,1,nframe); % TXA
35 ch_2i_data=zeros(lframe,1,nframe); % TXA
36 ch_2q_data=zeros(lframe,1,nframe); % TXA
37 ch_3i_data=zeros(lframe,1,nframe); % TXb
38 ch_3q_data=zeros(lframe,1,nframe); % TXb
39 ch_4i_data=zeros(lframe,1,nframe); % TXb
40 ch_4q_data=zeros(lframe,1,nframe); % TXb
41
42 %s_ini=zeros(nframe,n_simb-2);%Initialize record of transmitted
43 symbols (Guarda el que es va enviant?? deixem 2 zeros perquè
44 indicaran comença un nou simbol)
45
46
47 %% Transmissió pels canals
48
49
50 %%%%%% Channel 1
```

```

51
52 %%inicializació seqüència a enviar
53
54
55 s_ini=simb_TX(randi(M,1,n_simb-2));%random symbols to transmit (-2
56 que els dos primers son per detectar), tria aleatoriament entre
57 simbols a detectar
58 simb=repmat([0 0 s_ini],l_s,1);simb=simb(:);%Enlarge pulses to
59 avoid distortion
60 % Cada Frame està conformat per simbols que s'escullen de forma
61 aleatoria
62 %Utilitzarem únicament una seqüència per comparar les fases per
63 una
64 %mateixa seqüència enviada
65
66
67 for cont=1:nframe      %transmeto 15 frames on primer només envio pel
68 receptor 1
69
70     ch_1i_data(:,1,cont)=real(simb*1);
71     ch_1q_data(:,1,cont)=imag(simb*1);
72
73     ch_2i_data(:,1,cont)=real(simb*0); %0 no transmeto res pels
74 altres transmissors excepte pel qu'em'interessa
75     ch_2q_data(:,1,cont)=imag(simb*0);
76
77     ch_3i_data(:,1,cont)=real(simb*0);
78     ch_3q_data(:,1,cont)=imag(simb*0);
79
80     ch_4i_data(:,1,cont)=real(simb*0);
81     ch_4q_data(:,1,cont)=imag(simb*0);
82
83 end
84
85
86 %Build time series that feed the simulink model (la placa demana
87 aquest
88 %tipus d'entrada, associa el temps de la trama amb les dades a
89 enviar)
90 ch_1i=timeseries(ch_1i_data,t_series);
91 ch_1q=timeseries(ch_1q_data,t_series);
92
93 ch_2i=timeseries(ch_2i_data,t_series);
94 ch_2q=timeseries(ch_2q_data,t_series);
95
96 ch_3i=timeseries(ch_3i_data,t_series);
97 ch_3q=timeseries(ch_3q_data,t_series);
98
99 ch_4i=timeseries(ch_4i_data,t_series);
100 ch_4q=timeseries(ch_4q_data,t_series);
101
102
103 adi.AD9361.Tx
104 sim('qpsk_4ch.slx',stop_time)% amb les dades creadas simulo amb el
105 circuit simulink
106
107 %Recullo dades després de la transmissió
108 ch_cali=timeseries(ch_1i_data,t_series);
109 ch_calq=timeseries(ch_1q_data,t_series);
110 %% Hola
111

```

```

112 %load RX1_1.mat
113
114 cal_ccconv_frame
115 ch1_corr=S_corr;
116
117
118 %%% Channel 2
119
120 for cont=1:nframe %transmeto 15 frames només envio pel receptor 2
121 % s_ini(cont,:)=simb_TX(randi(M,1,n_simb-2));%random symbols to
122 transmit
123 % simb=repmat([0 0 s_ini(cont,:)],l_s,1);simb=simb(:);%Enlarge
124 pulses to avoid distortion
125
126 ch_1i_data(:,1,cont)=real(simb*0);
127 ch_1q_data(:,1,cont)=imag(simb*0);
128
129 ch_2i_data(:,1,cont)=real(simb*1);
130 ch_2q_data(:,1,cont)=imag(simb*1);
131
132 ch_3i_data(:,1,cont)=real(simb*0);
133 ch_3q_data(:,1,cont)=imag(simb*0);
134
135 ch_4i_data(:,1,cont)=real(simb*0);
136 ch_4q_data(:,1,cont)=imag(simb*0);
137
138 end
139
140 %Build time series that feed the simulink model
141 ch_1i=timeseries(ch_1i_data,t_series);
142 ch_1q=timeseries(ch_1q_data,t_series);
143
144 ch_2i=timeseries(ch_2i_data,t_series);
145 ch_2q=timeseries(ch_2q_data,t_series);
146
147 ch_3i=timeseries(ch_3i_data,t_series);
148 ch_3q=timeseries(ch_3q_data,t_series);
149
150 ch_4i=timeseries(ch_4i_data,t_series);
151 ch_4q=timeseries(ch_4q_data,t_series);
152
153
154
155 sim('qpsk_4ch.slx',stop_time)
156 ch_cali=timeseries(ch_2i_data,t_series);
157 ch_calq=timeseries(ch_2q_data,t_series);
158
159
160 %load RX1_2.mat
161 cal_ccconv_frame
162 ch2_corr=S_corr;
163
164
165 %%% Channel 3
166
167 for cont=1:nframe %transmeto 15 frames només envio pel receptor 2
168 % s_ini(cont,:)=simb_TX(randi(M,1,n_simb-2));%random symbols to
169 transmit
170 % simb=repmat([0 0 s_ini(cont,:)],l_s,1);simb=simb(:);%Enlarge
171 pulses to avoid distortion
172

```

```

173     ch_1i_data(:,1,cont)=real(simb*0);
174     ch_1q_data(:,1,cont)=imag(simb*0);
175
176     ch_2i_data(:,1,cont)=real(simb*0);
177     ch_2q_data(:,1,cont)=imag(simb*0);
178
179     ch_3i_data(:,1,cont)=real(simb*1);
180     ch_3q_data(:,1,cont)=imag(simb*1);
181
182     ch_4i_data(:,1,cont)=real(simb*0);
183     ch_4q_data(:,1,cont)=imag(simb*0);
184
185 end
186
187 %Build time series that feed the simulink model
188 ch_1i=timeseries(ch_1i_data,t_series);
189 ch_1q=timeseries(ch_1q_data,t_series);
190
191 ch_2i=timeseries(ch_2i_data,t_series);
192 ch_2q=timeseries(ch_2q_data,t_series);
193
194 ch_3i=timeseries(ch_3i_data,t_series);
195 ch_3q=timeseries(ch_3q_data,t_series);
196
197 ch_4i=timeseries(ch_4i_data,t_series);
198 ch_4q=timeseries(ch_4q_data,t_series);
199
200
201
202
203 sim('qpsk_4ch.slx',stop_time)
204 ch_cali=timeseries(ch_3i_data,t_series);
205 ch_calq=timeseries(ch_3q_data,t_series);
206
207 %load RX1_3.mat
208
209 cal_ccconv_frame
210 ch3_corr=S_corr;
211
212
213
214 %%% Channel 4
215
216 for cont=1:nframe    %transmeto 15 frames només envio pel receptor 2
217 %    s_ini(cont,:)=simb_TX(randi(M,1,n_simb-2));%random symbols to
218 %    transmit
219 %    simb=repmat([0 0 s_ini(cont,:)],1,s,1);simb=simb(:);%Enlarge
220 %    pulses to avoid distortion
221
222     ch_1i_data(:,1,cont)=real(simb*0);
223     ch_1q_data(:,1,cont)=imag(simb*0);
224
225     ch_2i_data(:,1,cont)=real(simb*0);
226     ch_2q_data(:,1,cont)=imag(simb*0);
227
228     ch_3i_data(:,1,cont)=real(simb*0);
229     ch_3q_data(:,1,cont)=imag(simb*0);
230
231     ch_4i_data(:,1,cont)=real(simb*1);
232     ch_4q_data(:,1,cont)=imag(simb*1);
233

```

```

234 end
235
236 %Build time series that feed the simulink model
237 ch_1i=timeseries(ch_1i_data,t_series);
238 ch_1q=timeseries(ch_1q_data,t_series);
239
240 ch_2i=timeseries(ch_2i_data,t_series);
241 ch_2q=timeseries(ch_2q_data,t_series);
242
243 ch_3i=timeseries(ch_3i_data,t_series);
244 ch_3q=timeseries(ch_3q_data,t_series);
245
246 ch_4i=timeseries(ch_4i_data,t_series);
247 ch_4q=timeseries(ch_4q_data,t_series);
248
249
250
251 sim('qpsk_4ch.slx',stop_time)
252 ch_cali=timeseries(ch_4i_data,t_series);
253 ch_calq=timeseries(ch_4q_data,t_series);
254
255 %load RX1_4.mat
256 cal_cconv_frame
257 ch4_corr=S_corr;
258
259
260
261 H_3_1=[ch1_corr ch2_corr ch3_corr ch4_corr] %vector de canal
262
263 save H_3_1.mat H_3_1 ;

```

### cal\_cconv\_frame.mat

```

1 %% Inici
2 contframe=10;
3 refconv=zeros(lframe,1);refconv(1:(2*l_s))=1; %poso un 1 als dos
4 primers simbols
5
6
7 simbinicial=zeros((lframe-2*l_s)/l_s,contframe); %simbol inicial=0
8 simbfinal=zeros((lframe-2*l_s)/l_s,contframe); %simbol final=0
9
10 %% For
11 for cont=1:contframe,
12     %cont
13     yyy=RX1.data(:,1,cont+4);yyy=yyy(:); % El primer correspon amb el
14     4rt, guardo els valors de sortida del receptor 1
15
16     ref=ch_cali.data((2*l_s+1):end,1,cont+1)+j*ch_calq.data((2*l_s+1):end
17     ,1,cont+1);ref=ref(:);
18
19     xxx=cconv(abs(yyy),refconv,lframe);
20     plot(abs(xxx));

```

```

21      plot(ref)
22      plot(yyy)
23
24      pos_min=find(xxx==min(xxx));pos_min=mod(pos_min+1-2*l_s,lframe);
25 % busco inici
26
27      zzz=circshift(yyy,-pos_min); % shift circularment per tenir
28 l'inici al principi
29      vvv=zzz((2*l_s+1):end);
30
31      simbinicial(:,cont)=ref((l_s/2):l_s:end); %Llegeixo el punt mig
32      simbfinal(:,cont) =vvv((l_s/2):l_s:end);
33 end;
34 %Per cada frame rebut, l'he concatenat 10 vegades per eviat problemes
35 de
36 %distorsió
37 %Per trobar l'inici de cada frame busco el minim d'una convolució amb
38 només
39 %2, que es convolucionaran amb els 2 primers 0 que marquen inici d'un
40 frame
41 % i donaran el mínim, d'això guardaré l'inici i respecto el qual tinc
42 la
43 % referencia
44
45 %% FI
46 S_corr=conj(mean(simbfinal(:)./simbinicial(:))) % calculo per tenir
47 la fase del canal
48 S_L=mean(abs(simbfinal(:)))
49
50 %save inicial_30_44 simbinicial
51 %save final_30_44 simbfinal
52 %save corr_30_44 S_corr
53 % s_f=simbfinal(:)/mean(abs(simbfinal(:)));
54 % s_i=simbinicial(:)/mean(abs(simbinicial(:)));
55 % SNR=20*log10(mean(abs(s_f))/mean(abs(s_i-s_f)))

```

## 8.1.2. Codi Enviar 4 tons

```

1 %% Receive Tone Signal Using Analog Devices AD9361/AD9364
2 %
3 % This example shows how to use the Xilinx(R) Zynq-Based Radio
4 Support
5 % Package and Communications Toolbox(TM) software to perform a
6 % simple loopback of a complex sinusoid signal at RF Using Analog
7 % Devices(TM) AD9361/AD9364. A Direct Digital
8 % Synthesizer (DDS) in the FPGA generates a complex sinusoid and
9 transmits
10 % it using the RF card. The transmitted signal is then received by
11 the RF
12 % card and the downsampled baseband signal is visualized in
13 MATLAB(R). This
14 % simple example confirms that the SDR hardware is setup correctly
15 and
16 % shows how to capture RF data from SDR hardware using MATLAB.
17

```

```
18 % Copyright 2014-2019 The MathWorks, Inc.
19
20 %% Configure SDR Hardware
21 % If you are running this example after performing the setup of
22 % hardware using Support Package Installer then you can skip this
23 section.
24 % Otherwise, follow <docid:xilinxzynqbasedradio_ug#bulkohy-14 Guided
25 Host-Radio
26 % Hardware Setup> to configure your host computer to work with the
27 SDR hardware.
28 % Connect an SMA loopback cable with attenuation between TX1A and
29 RX1A (for
30 % FMCOMMS2 or FMCOMMS3), between TXA and RXA (for FMCOMMS4), between
31 % TX1A_A and RX1A_A (for FMCOMMS5), or attach appropriate antenna
32 suitable
33 % for 2.4 GHz band.
34
35 %% Running the Example
36 % This example can be run by executing
37 % <matlab:edit('zyngRadioToneReceiverAD9361AD9364ML')>
38 % zyngRadioToneReceiverAD9361AD9364ML.m>.
39 if ~exist('prmToneRx','var')
40     prmToneRx.SDRDeviceName = 'FMCOMMS5';
41     prmToneRx.IPAddress = '158.109.64.241';
42 end
43 % To update the example for FMCOMMS5, set |prmToneRx.SDRDeviceName| to
44 % '|FMCOMMS5'|.
45 %% Transmit a Tone Signal from the FPGA
46 % Set the Direct Digital Synthesizer (DDS) in the FPGA fabric to
47 transmit a
48 % complex sinusoid to the RF card. This is provided in the FPGA for
49 testing
50 % and debugging purposes.
51
52 %%
53 % Create a transmitter System object(TM) to configure the RF card
54 settings. Set
55 % the RF card to transmit data at a center frequency of 2.4 GHz.
56
57 RadioBasebandRate = 1e6;
58 CenterFrequency = 3.6e9;
59 ToneFrequency = 25e3;
60
61 sdrTransmitter = sdrtx(prmToneRx.SDRDeviceName, ...
62     'IPAddress', prmToneRx.IPAddress, ...
63     'CenterFrequency', CenterFrequency, ...
64     'ChannelMapping', [1 2 3 4]);
65
66
67
68
69 %%
70 % Turn on the properties related to DDS by setting
71 % |ShowAdvancedProperties| to true. Set the |DataSourceSelect|
72 property of
73 % sdrTransmitter System object to 'DDS'. Set the tone frequency and
74 scale for
75 % DDS.
76
77 sdrTransmitter.ShowAdvancedProperties = true;
78 sdrTransmitter.BasebandSampleRate = RadioBasebandRate;
```

```

79 sdrTransmitter.DataSourceSelect = 'DDS';
80 %sdrTransmitter.DDSTone1Scale = [0.1];
81 %sdrTransmitter.DDSTone2Scale = [0.1];
82 %sdrTransmitter.DDSTone1Freq = [ToneFrequency]
83 %sdrTransmitter.DDSTone2Freq = [ToneFrequency]
84 sdrTransmitter.DDSTone1Scale = [0.1 0.1 0.1 0.1];
85 sdrTransmitter.DDSTone2Scale = [0.1 0.1 0.1 0.1];
86 sdrTransmitter.DDSTone1Freq =
87 [ToneFrequency,2*ToneFrequency,3*ToneFrequency,4*ToneFrequency]
88 sdrTransmitter.DDSTone2Freq =
89 [ToneFrequency,2*ToneFrequency,3*ToneFrequency,4*ToneFrequency]
90 sdrTransmitter.Gain = 0;
91
92
93
94
95
96 %%
97 % Next, initiate the transmission of data from DDS to RF card.
98
99 sdrTransmitter();
100
101 %%
102 % Create our data to send
103
104
105 %%
106 % Note that the simultaneous transmission and reception of data
107 % from MATLAB to RF card (duplex) is currently not supported.
108 Therefore, for
109 % this example the data is generated in FPGA using DDS and
110 transmitted directly to the
111 % RF card. MATLAB is only used for signal reception
112
113 %% Capture RF Signal
114 % To capture the RF tone signal into MATLAB create an SDR receiver
115 System
116 % object and configure it to receive the samples at the baseband
117 rate.
118
119 % Radio parameters
120 RadioFrameLength = 4000;
121
122 % Create a receiver System object with desired radio parameters
123 sdrReceiver = sdrrx(prmToneRx.SDRDeviceName, ...
124     'IPAddress', prmToneRx.IPAddress, ...
125     'CenterFrequency', CenterFrequency, ...
126     'BasebandSampleRate', RadioBasebandRate, ...
127     'GainSource', 'Manual', ...
128     'Gain', 30, ...
129     'SamplesPerFrame', RadioFrameLength, ...
130     'ChannelMapping', 1, ...
131     'OutputDataType', 'double');
132 %%
133 % To visualize the received signal in frequency and time domain use
134 % Spectrum Analyzer and Time Scope System objects. In addition, set
135 up a
136 % Constellation Diagram System object for plotting signal as two
137 % dimensional scatter diagram in the complex plane.
138 spectrumScope = dsp.SpectrumAnalyzer('SampleRate',
139 RadioBasebandRate);

```

```

140 timeScope =
141 dsp.TimeScope('TimeSpan',5/ToneFrequency,'SampleRate',RadioBasebandRa
142 te);
143 constellation =
144 comm.ConstellationDiagram('ShowReferenceConstellation', false);
145
146 % Create a container for the three scopes
147 scopesContainer = HelperCreateScopesContainer(... ...
148     {spectrumScope,timeScope,constellation},...
149     'Name','Zynq Radio Tone Receiver',...
150     'Layout',[2 2],...
151     'ExpandToolbar',false);
152 scopesContainer.setColumnSpan(1,1,2);
153
154 %%
155 % Set the simulation to capture 100 milliseconds of data.
156
157 StopTime      = 100e-3;                                % seconds
158 RadioFrameTime = (RadioFrameLength / RadioBasebandRate); % seconds
159 %%
160 %
161 % If the processing of received data in MATLAB is slower than the
162 speed at
163 % which the data is captured, you will encounter loss of samples.
164 This will
165 % be reflected by non-zero value of |overflow| variable. To ensure
166 % reception of contiguous data in MATLAB you can capture signals by
167 % enabling <docid:xilinxzynqbaseradio_ug#bulkof4-43 burst mode> and
168 by specifying
169 % the number of frames as the size of burst. In this mode, the
170 specified
171 % amount of data is captured in a buffer first and later it is
172 available
173 % for processing in MATLAB. Due to a limitation in the burst mode,
174 overflows in
175 % the first step when using burst mode are ignored.
176
177 numFramesinBurst = ceil(RadioBasebandRate*StopTime/RadioFrameLength);
178 sdrReceiver.EnableBurstMode = true;
179 sdrReceiver.NumFramesInBurst = numFramesinBurst;
180 %%
181 % Capture and visualize data by calling the corresponding System
182 % objects in a loop.
183
184 try
185     % Loop until the example reaches the target stop time.
186     timeCounter = 0;
187
188     while timeCounter < StopTime
189
190         [data, valid, overflow] = sdrReceiver();
191         if (overflow > 0) && (timeCounter > 0)
192
193             warning(message('sdrpluginbase:zynqradiotransmitter:DroppedSamples'));
194         end
195
196         if valid
197             % Visualize frequency spectrum
198             spectrumScope(data);
199             % Visualize in time domain
200             timeScope([real(data), imag(data)]);

```

```

201           % Visualize the scatter plot
202           constellation(data);
203
204           % Set the limits in scopes
205           dataMaxLimit = max(max(abs([real(data); imag(data)])));
206           constellation.XLimits = [-dataMaxLimit*1.5,
207 dataMaxLimit*1.5];
208           constellation.YLimits = [-dataMaxLimit*1.5,
209 dataMaxLimit*1.5];
210           timeScope.YLimits = [-dataMaxLimit*2, dataMaxLimit*2];
211           timeCounter = timeCounter + RadioFrameTime;
212       end
213   end
214 catch ME
215     rethrow(ME);
216 end
217
218 %% Visualize Signal
219 % You will notice a peak at 25 kHz corresponding to the received tone
220 signal
221 % in frequency spectrum in Signal Analyzer. Depending on the quality
222 of signal
223 % received, you may see a peak around DC and negative 25 kHz in
224 frequency
225 % spectrum indicating existence of a DC offset and IQ imbalance
226 % respectively. You should see sinusoidal signals (real and
227 imaginary) in
228 % the time domain, shown in the Time Scope display. In Constellation
229 % Diagram display, you should see a ring like plot visualizing the
230 complex
231 % sinusoidal vector signal in the complex plane. The ring should be a
232 % perfect circle. Any warping of the circle is an indication of IQ
233 % imbalance.
234 %
235 % <<zynqRadioToneReceiverFreqSpectrumAD9361AD9364ML.png>>
236 %
237 % <<zynqRadioToneReceiverTimeScopePlotAD9361AD9364ML.png>>
238 %
239 % <<zynqRadioToneReceiverConstDiagmPlotAD9361AD9364ML.png>>
240
241 %%
242 % Release the SDR Transmitter/Receiver and visualization scopes.
243
244 release(sdrReceiver);
245 release(sdrTransmitter);
246
247 release(spectrumScope);
248 release(timeScope);
249 release(constellation);
250
251 %% Conclusion
252 % In this example, you used SDR Transmitter and Receiver System
253 objects to
254 % transmit a complex sinusoidal signal from the FPGA and receive it
255 in MATLAB. You
256 % visualized the received signal in time, frequency and on the
257 complex plane. By
258 % performing this loopback test, you can confirm that the SDR system
259 is
260 % setup correctly. You can now proceed to use it in conjunction with
261 % Communications Toolbox to develop your baseband algorithms and

```

```
262 % verify using real world RF data.
263 displayEndOfDemoMessage(mfilename)
```

### 8.1.3. Codi de Matlab pel controlador

```

1 clear all
2
3 n_vueltas = 1;%30
4 step = 1;
5 motor = 1;%axis 1 = azimuth, axis 2 = roll, axis 3 = polar
6 speed = 70; % percentatge velocitat es mou
7 dir = 'FWD'; %direcció en moure's
8
9 tic
10 t=tcpip('158.109.74.12',4444);
11 fopen(t)
12 fprintf(t,'@REM\n');%control remoto
13 fscanf(t)
14 fprintf(t,'MOVE:POSI:AXIS %s\n',num2str(motor)); %indico paràmetres control
15 remot
16 fprintf(t,'MOVE:POSI:SPEE %s\n',num2str(speed)); %velocitat a la que rota
17 fprintf(t,'MOVE:POSI:DIRE %s\n',dir); %direcció de rotació
18
19
20 ph_med=[-90:step:90]; %angles entre els quals es mou
21 tabla_pos=zeros(length(ph_med),n_vueltas);
22 tabla_med=zeros(length(ph_med),n_vueltas);
23 n_frame=3;
24 n_simb=500000-2;
25 dades_rebudes=zeros(n_simb*n_frame,length(ph_med));
26 dades_enviades=zeros(n_simb*n_frame,length(ph_med));
27 s=zeros(n_simb,length(ph_med));
28
29 for cont=1:n_vueltas,
30     cont
31         for cont2=1:length(ph_med)
32             tabla_pos(cont2,cont)=ph_med(cont2);%posicion teorica
33             fprintf(t,'MOVE:POSI:TO %d\n',ph_med(cont2));%nueva posicion
34             fprintf(t,'MOVE:POSI:STAR\n');%empieza movimiento
35             pause(1); %espero abans de transmetre per evitar transmetre mentre
36             es mou
37
38
39
40         enviar_modulacio_direccional %transmeto amb DM
41         dades_rebudes(:, cont2)=simbfinal(:);
42         dades_enviades(:, cont2)=simbemes(:);
43         s(:, cont2)=s_ini1(:);
44
45     end
46 end;
47
48 fprintf(t,'@LOC\n');%control remoto
49
```

```

50
51 fclose(t)
52 toc
53
54
55 % plot(ph_med,20*log10(abs(dades_rebudes)),'b')
56
57 % plot(ph_med,(angle(dades_rebudes))*180/pi,'b')

```

### 8.1.4. Calcular vector del canal

```

1 if ~exist('prmToneRx','var')
2     prmToneRx.SDRDeviceName = 'FMCOMMS5';
3     prmToneRx.IPAddress = '158.109.64.241';
4 end
5 % To update the example for FMCOMMS5, set |prmToneRx.SDRDeviceName|
6 to
7 % |'FMCOMMS5'|.
8
9
10%%
11 % Create a transmitter System object(TM) to configure the RF card
12 settings. Set
13 % the RF card to transmit data at a center frequency of 2.4 GHz.
14
15 RadioBasebandRate = 15.36e6;
16 CenterFrequency = 3.6e9;
17 %ToneFrequency = 25e3;
18
19 sdrTransmitter = sdrtx(prmToneRx.SDRDeviceName, ...
20     'IPAddress',      prmToneRx.IPAddress, ...
21     'CenterFrequency', CenterFrequency, ...
22     'BasebandSampleRate',    RadioBasebandRate, ...
23     'CenterFrequency',       CenterFrequency, ...
24     'ChannelMapping',       [1 2 3 4], ...
25     'ShowAdvancedProperties', true);
26
27
28
29
30 fs=15.36e6;%Sampling frequency
31 Ts=1/fs;%Sampling period
32 n_simb=1024;% # symbols per frame
33 l_s=10; % # samples in a symbol
34 nframe=15;% # frames
35
36 M=4; %QPSK = 4, 16-QAM = 16
37 %Generem els axis dels simbols a enviar
38 X=linspace(-sqrt(2)/2,sqrt(2)/2,sqrt(M));% Generem 2 (^4) punts entre
39 -^2/2 i ^2/2
40 Y=X;
41 [YY,XX]=meshgrid(Y,X); %creem els simbols de la QPSK (constelació 4)
42 simb_TX=XX+1i*YY;
43 simb_TX=simb_TX(:);%Available symbols (simbols disponibles)
44
45
46 lframe=n_simb*l_s*nframe;% length frame in samples = (n simb/frame)*

```

```

47 (mostres/simbol)
48
49
50
51 envi1=double.empty;
52 envi2=double.empty;
53 envi3=double.empty;
54 envi4=double.empty;
55
56 s_ini=simb_TX(randi(M,1,n_simb-2));%random symbols to transmit (-2
57 que els dos primers son per detectar), tria aleatoriament entre
58 simbols a detectar
59 %s_ini=simb_TX([1 2 3 4 1 1 2 2 3 3 4 4]);%random symbols to transmit
60 (-2 que els dos primers son per detectar), tria aleatoriament entre
61 simbols a detectar
62 simb=repmat([0 0 s_ini],l_s,1);
63 simb=simb(:);%Enlarge pulses to avoid distortion
64 %% Channel 1
65
66 for cont=1:nframe
67
68
69     envi1=simb*1;
70     envi2=simb*0;
71     envi3= simb*0;
72     envi4= simb*0;
73 end
74
75
76
77 envi=[envi1 envi2 envi3 envi4];
78
79 transmitRepeat(sdrTransmitter, envi );
80 %n_simb, RadioBasebandRate, RadioFrameLength, lframe, Ts, nframe,
81 simb
82
83 %% Capture RF Signal
84 % To capture the RF tone signal into MATLAB create an SDR receiver
85 System
86 % object and configure it to receive the samples at the baseband
87 rate.
88
89 % Radio parameters
90 RadioFrameLength = n_simb*l_s*1;
91
92 % Create a receiver System object with desired radio parameters
93 sdrReceiver = sdrRx(prmToneRx.SDRDeviceName, ...
94     'IPAddress', prmToneRx.IPAddress, ...
95     'CenterFrequency', CenterFrequency, ...
96     'BasebandSampleRate', RadioBasebandRate, ...
97     'GainSource', 'Manual', ...
98     'Gain', 32, ...
99     'SamplesPerFrame', RadioFrameLength, ...
100    'ChannelMapping', 1, ...
101    'OutputDataType', 'double');
102 %% Sent data and calculate phase
103
104 corr_canall=Calcular_corr(n_simb, l_s, RadioBasebandRate,
105 RadioFrameLength, lframe, Ts, nframe, simb, sdrReceiver);
106
107 release(sdrTransmitter);

```

```
108 %displayEndOfDemoMessage(mfilename);
109 %% Channel 2
110
111 envi1=double.empty;
112 envi2=double.empty;
113 envi3=double.empty;
114 envi4=double.empty;
115
116 for cont=1:nframe
117
118     envi1=simb*0;
119     envi2=simb*1;
120     envi3=simb*0;
121     envi4= simb*0;
122
123 end
124
125
126
127 envi=[envi1 envi2 envi3 envi4];
128
129 transmitRepeat(sdrTransmitter, envi );
130 corr_canal2=Calcular_corr(n_simb, l_s, RadioBasebandRate,
131 RadioFrameLength, lframe, Ts, nframe, simb, sdrReceiver);
132
133 release(sdrTransmitter);
134 %% Channel 3
135
136 envi1=double.empty;
137 envi2=double.empty;
138 envi3=double.empty;
139 envi4=double.empty;
140
141 for cont=1:nframe
142
143
144     envi1=simb*0;
145     envi2=simb*0;
146     envi3=simb*1;
147     envi4= simb*0;
148
149 end
150
151
152 envi=[envi1 envi2 envi3 envi4];
153
154 transmitRepeat(sdrTransmitter, envi );
155 corr_canal3=Calcular_corr(n_simb, l_s, RadioBasebandRate,
156 RadioFrameLength, lframe, Ts, nframe, simb, sdrReceiver);
157
158 release(sdrTransmitter);
159
160 %% Channel 4
161
162 envi1=double.empty;
163 envi2=double.empty;
164 envi3=double.empty;
165 envi4=double.empty;
166
167 for cont=1:nframe
168
```

```

169
170     envi1=simb*0;
171     envi2=simb*0;
172     envi3=simb*0;
173     envi4= simb*1;
174 end
175
176
177
178 envi=[envi1 envi2 envi3 envi4];
179
180 transmitRepeat(sdrTransmitter, envi );
181 corr_canal4=Calcular_corr(n_simb, l_s, RadioBasebandRate,
182 RadioFrameLength, lframe, Ts, nframe, simb, sdrReceiver);
183
184 release(sdrTransmitter);
185
186 [corr_canal1, corr_canal2, corr_canal3, corr_canal4]
187
188 function symbols_reordenats=reordenar(dades_rebudes, length_frame,
189 n_frames, repeticions, refconv )
190
191
192 xxx=cconv(abs(dades_rebudes),refconv(1:(length_frame/n_frames)),length_
193 h_frame/n_frames);
194 plot(dades_rebudes)
195 plot(xxx)
196
197 pos_min=find(xxx==min(xxx));
198 pos_min=mod(pos_min+1-2*repeticions,length_frame);
199 zzz=circshift(dades_rebudes,-pos_min); % shift circularment per
200 tenir l'inici al principi
201 vvv=zzz((2*repeticions):end);
202 symbols_reordenats =vvv((repeticions/2):repeticions:end);
203
204 end
205 function S_corr=Calcular_corr(n_simb_enviats,l_s,
206 RadioBasebandRate_f,RadioFrameLength_f, long_frame, T_sample,
207 num_frames, simb_trans, receptor)
208
209 spectrumScope = dsp.SpectrumAnalyzer('SampleRate',
210 RadioBasebandRate_f);
211 timeScope =
212 dsp.TimeScope('TimeSpan',long_frame,'SampleRate',RadioBasebandRate_f)
213 ;
214 constellation =
215 comm.ConstellationDiagram('ShowReferenceConstellation', false);
216
217 % Create a container for the three scopes
218 scopesContainer = HelperCreateScopesContainer(...%
219 {spectrumScope,timeScope,constellation},...
220 'Name','Zynq Radio QPSK Receiver',...
221 'Layout',[2 2],...
222 'ExpandToolbar',false);
223 scopesContainer.setColumnSpan(1,1,2);
224
225
226 StopTime      = long_frame*T_sample; %4 antenes
227 RadioFrameTime = (RadioFrameLength_f / RadioBasebandRate_f); %
228 seconds
229

```

```

230     numFramesinBurst =
231     ceil(RadioBasebandRate_f*StopTime/RadioFrameLength_f);
232     receptor.EnableBurstMode = true;
233     receptor.NumFramesInBurst = numFramesinBurst;
234
235     simbinicial=zeros(n_simb_enviats-2,num_frames); %simbol inicial=0
236     simbfinal=zeros(n_simb_enviats-2,num_frames);
237     refconv=zeros(long_frame/num_frames,1);
238
239     refconv(1:(2*ls))=1;
240     try
241         % Loop until the example reaches the target stop time.
242         timeCounter = 0;
243         num_frame=1;
244         while timeCounter < StopTime
245
246             [data, valid, overflow] = receptor();
247             if (overflow > 0) && (timeCounter > 0)
248
249             warning(message('sdrpluginbase:zynqradioExamples:DroppedSamples'));
250             end
251
252             if valid
253                 % Visualize frequency spectrum
254                 spectrumScope(data);
255                 % Visualize in time domain
256                 timeScope([real(data), imag(data)]);
257                 % Visualize the scatter plot
258                 constellation(data);
259
260
261             simbinicial(1:end,num_frame)=simb_trans((2*ls+(ls/2)):ls:end);
262             %Llegeixo el punt mig
263
264             simbfinal(:,num_frame) = reordenar(data, long_frame,
265             num_frames, ls, refconv);
266
267             % Set the limits in scopes
268             dataMaxLimit = max(max(abs([real(data);
269             imag(data)])));
270             constellation.XLimits = [-dataMaxLimit*1.5,
271             dataMaxLimit*1.5];
272             constellation.YLimits = [-dataMaxLimit*1.5,
273             dataMaxLimit*1.5];
274             timeScope.YLimits = [-dataMaxLimit*2,
275             dataMaxLimit*2];
276             timeCounter = timeCounter + RadioFrameTime;
277             num_frame = num_frame + 1;
278         end
279
280     end
281
282     catch ME
283         rethrow(ME);
284     end
285
286     enviat_channel1=simbinicial(:,1:num_frames);
287
288     rebut_channel1=simbfinal(:,1:num_frames);
289     S_corr=conj(mean(rebut_channel1(:)./enviat_channel1(:)));
290

```

```

291     release(receptor);
292     release(spectrumScope);
293     release(timeScope);
294     release(constellation);
295
296 end

```

### 8.1.5. Transmissió amb el canal calibrat

```

1 %% SDR Configuration
2 if ~exist('prmToneRx','var')
3     prmToneRx.SDRDeviceName = 'FMCOMMS5';
4     prmToneRx.IPAddress = '158.109.64.241';
5 end
6 % To update the example for FMCOMMS5, set |prmToneRx.SDRDeviceName|
7 to
8 % |'FMCOMMS5'|.
9
10
11 %% Transmitter Configuration
12 % Create a transmitter System object(TM) to configure the RF card
13 settings. Set
14 % the RF card to transmit data at a center frequency of 2.4 GHz.
15
16 RadioBasebandRate = 15.36e6;
17 CenterFrequency = 3.6e9;
18 %ToneFrequency = 25e3;
19
20 sdrTransmitter = sdrtx(prmToneRx.SDRDeviceName, ...
21     'IPAddress', prmToneRx.IPAddress, ...
22     'CenterFrequency', CenterFrequency, ...
23     'BasebandSampleRate', RadioBasebandRate, ...
24     'CenterFrequency', CenterFrequency, ...
25     'ChannelMapping', [1 2 3 4], ...
26     'ShowAdvancedProperties', true);
27 %% Set Information to send with Directional Modulation
28
29 H_adj=[ -0.0517 - 0.0024i -0.0505 + 0.0317i 0.0212 - 0.0571i
30 0.0399 - 0.0345i]; %definim vector de canal calculat previament
31
32
33
34 [base_W, P]=calcular_ortogonals(H_adj);
35
36
37 fs=15.36e6;%Sampling frequency
38 Ts=1/fs;%Sampling period
39 n_simb=1024;% # symbols per frame
40 l_s=10; % # samples in a symbol
41 nframe=1;% # frames
42
43 M=4; %QPSK = 4, 16-QAM = 16
44 %Generem els axis dels simbols a enviar
45 X=linspace(-sqrt(2)/2,sqrt(2)/2,sqrt(M));% Generem 2 (^4) punts entre
46 -^2/2 i ^2/2
47 Y=X;
48 [YY,XX]=meshgrid(Y,X); %creem els simbols de la QPSK (constelaciÃ³ 4)

```

```

49 simb_TX=XX+1i*YY;
50 simb_TX=simb_TX(:);%Available symbols (symbols disponibles)
51
52 n_radians=4;
53 lframe=n_simb*l_s*nframe;% length frame in samples = (n simb/frame)*
54 (mostres/simbol)
55
56
57
58
59 s_ini=simb_TX(randi(M,1,n_simb-2));%random symbols to transmit (-2
60 que els dos primers son per detectar), tria aleatoriament entre
61 simbols a detectar
62 %s_ini=simb_TX([1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3
63 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2
64 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1
65 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1
66 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
67 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
68 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2
69 transmit (-2 que els dos primers son per detectar), tria
70 aleatoriament entre simbols a detectar
71
72
73
74
75
76 simb=repmat([0 0 s_ini],l_s,1);
77 simb=simb(:);%Enlarge pulses to avoid distortion
78 %w=zeros(length(simb),n_radians);
79 %for k=1:length(simb)
80 %    [numRows_w, numCol_w]=size(base_W);
81 %    pesos=1*(randn(numCol_w,1)+1i*randn(numCol_w,1));
82 %
83 %    w(k,:)=base_W*pesos;
84
85 %end
86
87 envi1=double.empty;
88 envi2=double.empty;
89 envi3=double.empty;
90 envi4=double.empty;
91
92 for cont=1:nframe
93
94     envi1=simb*1;
95     envi2=simb*1;
96     envi3=simb*1;
97     envi4= simb*1;
98
99 end
100 envi=[envi1 envi2 envi3 envi4];
101 X1=simb*P;
102 %X=X1+w;
103
104 %% Send Information
105 transmitRepeat(sdrTransmitter, envi);
106 %transmitRepeat(sdrTransmitter, X1);
107
108 %% Receiver configuration
109 % To capture the RF tone signal into MATLAB create an SDR receiver

```

```

110 System
111 % object and configure it to receive the samples at the baseband
112 rate.
113
114 % Radio parameters
115 RadioFrameLength = n_simb*l_s*1;
116
117 % Create a receiver System object with desired radio parameters
118 sdrReceiver = sdrrx(prmToneRx.SDRDeviceName, ...
119     'IPAddress', prmToneRx.IPAddress, ...
120     'CenterFrequency', CenterFrequency, ...
121     'BasebandSampleRate', RadioBasebandRate, ...
122     'GainSource', 'Manual', ...
123     'Gain', 32, ...
124     'SamplesPerFrame', RadioFrameLength, ...
125     'ChannelMapping', 1, ...
126     'OutputDataType', 'double');
127
128
129 %% Configure to analize receive data
130
131 spectrumScope = dsp.SpectrumAnalyzer('SampleRate',
132 RadioBasebandRate);
133 timeScope =
134 dsp.TimeScope('TimeSpan',lframe,'SampleRate',RadioBasebandRate);
135 constellation =
136 comm.ConstellationDiagram('ShowReferenceConstellation', false);
137
138 % Create a container for the three scopes
139 scopesContainer = HelperCreateScopesContainer(...,
140     {spectrumScope,timeScope,constellation},...
141     'Name','Zynq Radio QPSK Receiver',...
142     'Layout',[2 2],...
143     'ExpandToolbar',false);
144 scopesContainer.setColumnSpan(1,1,2);
145
146 %%
147
148
149 %stop_time=nframe*lframe*Ts;
150 StopTime      = lframe*Ts; %4 antenes
151 % seconds
152 RadioFrameTime = StopTime;% (RadioFrameLength / RadioBasebandRate); %
153 seconds
154
155
156 numFramesinBurst = ceil(RadioBasebandRate*StopTime/RadioFrameLength);
157 sdrReceiver.EnableBurstMode = true;
158 sdrReceiver.NumFramesInBurst = numFramesinBurst;
159 %% Receive Data a process it
160 % Capture and visualize data by calling the corresponding System
161 % objects in a loop.
162 simbinicial=zeros(n_simb-2,nframe); %simbol inicial=0
163 simbfinal=zeros(n_simb-2,nframe);
164 refconv=zeros(lframe/nframe,1);
165
166 refconv(1:(2*l_s))=1;
167 try
168     % Loop until the example reaches the target stop time.
169     timeCounter = 0;
170     num_frame=1;

```

```

171     while timeCounter < StopTime
172
173         [data, valid, overflow] = sdrReceiver();
174         if (overflow > 0) && (timeCounter > 0)
175
176     warning(message('sdrpluginbase:zynqradioExamples:DroppedSamples'));
177     end
178
179     if valid
180         % Visualize frequency spectrum
181         spectrumScope(data);
182         % Visualize in time domain
183         timeScope([real(data), imag(data)]);
184         % Visualize the scatter plot
185         constellation(data);
186
187
188 %%%
189 %
190
191 simbinicial(1:end,num_frame)=simb((2*l_s+(l_s/2)):l_s:end); %Llegeixo
192 el punt mig
193
194
195
196 xxxx=cconv(abs(data),refconv(1:(lframe/nframe)),lframe/nframe);
197 %plot(simb)
198 %plot(data)
199 %plot(xxx)
200
201 pos_min=find(xxx==min(xxx));
202 pos_min=mod(pos_min+1-2*l_s,lframe);
203 zzz=circshift(data,-pos_min); % shift circularment per
204 tenir l'inici al principi
205 vvv=zzz((2*l_s):end);
206 simbfinal(:,num_frame)=vvv((l_s/2):l_s:end);
207 %
208 %%%%%%
209 %%%
210 %
211
212
213
214 % Set the limits in scopes
215 dataMaxLimit = max(max(abs([real(data); imag(data)])));
216 constellation.XLimits = [-dataMaxLimit*1.5,
217 dataMaxLimit*1.5];
218 constellation.YLimits = [-dataMaxLimit*1.5,
219 dataMaxLimit*1.5];
220 timeScope.YLimits = [-dataMaxLimit*2, dataMaxLimit*2];
221 timeCounter = timeCounter + RadioFrameTime;
222 num_frame = num_frame + 1;
223
224
225 end
226
227 catch ME
228     rethrow(ME);
229 end
230 %% Close SDR Transmitter/Receiver and visualization scopes.
231

```

```

232 release(sdrReceiver);
233 release(sdrTransmitter);
234
235 release(spectrumScope);
236 release(timeScope);
237 release(constellation);
238
239 %% Funció per calcular base ortogonal pesos, i vector P
240
241 function [Ws,P] = calcular_ortogonals(H)
242     Ws=null(conj(H.')');
243     P = H/norm(H);
244     for k=1:length(P)
245         P(k)=P(k)/abs(P(k));
246     end
247 end

```

## 8.1.6. Codi modulació direccional teòrica

```

1 n_radians=5;
2 th=0:180;
3 for k=1:n_radians
4     H1(k,:)=exp(1i*(3-k)*pi*cosd(th));
5
6 end
7
8
9 theta_0=45;
10
11 [W,P] = calcular_ortogonals(H1, theta_0);
12
13
14 QPSK_x=linspace(-sqrt(2)/2, sqrt(2)/2, sqrt(4));
15 QPSK_y=QPSK_x;
16 [QPSK_YY,QPSK_XX]=meshgrid(QPSK_y,QPSK_x); %creem els simbols de la
17 QPSK (constelació 4)
18 simb_TX=QPSK_XX+1*i*QPSK_YY;
19 simb_TX=simb_TX(:);%Available symbols (simbols disponibles)
20 Y_entrenament=randi(4,1,500000);
21 s_ini=simb_TX(Y_entrenament);%random symbols to transmit (-2 que els
22 dos primers son per detectar), tria aleatoriament entre simbols a
23 detectar
24 simb=[s_ini];
25 simb=simb(:);%
26
27
28 %W=[W;W];
29
30
31
32 %w=[W(:,1).'; W(:,1).'; W(:,1).'; W(:,1).'];
33 w=zeros(length(simb),n_radians);
34
35
36 for k=1:length(simb)
37     [numRows_w, numCol_w]=size(W);

```

```
38     pesos=randn(numCol_w,1);
39
40     w(:, :)=W*pesos;
41
42 end
43
44 X1=simb*P;
45 X=X1+w;
46
47 %Y=zeros(180,length(simb));
48 %for k=(th+1)
49 %    H_canal=conj(H1(:,k).');
50 %    Y(k,:)=H_canal*(X.');
51
52
53 %end
54
55 H_canal=conj(H1(:,theta_0+30).');
56 X_entrenament=H_canal*(X.');
57
58 %scatterplot(YY)
59
60
61
62 Y_test=randi(4,1,1000000);
63
64 s_ini=simb_TX(Y_test);
65
66 simb=[s_ini];
67 simb=simb(:);%
68
69
70 w=zeros(length(simb),n_radians);
71
72
73
74 for k=1:length(simb)
75     [numRows_w, numCol_w]=size(W);
76     pesos=randn(numCol_w,1);
77
78     w(:, :)=W*pesos;
79
80 end
81
82 X1=simb*P;
83 X=X1+w;
84
85
86 H_canal=conj(H1(:,theta_0+30).');
87 X_test=H_canal*(X.');
88
89 %scatterplot(YY)
90
91 %H_canal=conj(H1(:,theta_0+36).');
92 %YY=H_canal*(X.');
93
94 %scatterplot(YY)
95
96 %H_canal=conj(H1(:,theta_0+76).');
97 %YY=H_canal*(X.');
98
```

```

99 %scatterplot(YY)
100
101 %figure()
102
103 %p=plot(Y);
104 %hold on
105 %plot(Y(46,:),'s', 'MarkerSize',10);
106 %plot(Y(91,:),'o', 'MarkerSize',10);
107 %plot(Y(136,:),'*', 'MarkerSize',10);
108 %axis('equal');
109 %lgd=legend('1','2','3','4','constelació a \theta_0', 'constelació a
110 '90°', 'constelació a 135°', 'Orientation','vertical');
111
112 %lgd.NumColumns = 3;
113 %figure
114 %plot(th,20*log10(abs(Y)));
115 %hold on
116 %plot([46],20*log10(abs(Y(46,:))), 's', 'MarkerSize',10);
117 %lgd=legend('1','2','3','4', '\theta_0');
118
119 %figure
120 %plot(th,angle(Y)*180/pi)
121 %hold on
122 %plot([46],angle(Y(46,:))*180/pi,'s', 'MarkerSize',10);
123 %lgd=legend('1','2','3','4', '\theta_0');
124
125
126
127
128 function [Ws,P] = calcular_ortogonals(H, TH_0)
129     A = H(:,TH_0+1)';
130     %Ha = A * H;
131     th=0:180;
132
133
134
135     base_ortho=null(A);
136     ori = (A)*H;
137     W1 = (A+base_ortho(:,1).')*H;
138     Ws = base_ortho;
139     W3 =(A+base_ortho(:,3).')*H;
140     P = H(:,TH_0+1).'/norm(H(:,TH_0+1));
141     %figure()
142     %plot(th,20*log10(abs(ori)/5));
143     %hold on
144     %plot(th,20*log10(abs(W1)/5));
145     %plot(th,20*log10(abs((A+base_ortho(:,2).')*H)/5));
146     %plot(th,20*log10(abs(W3)/5));
147     %plot(th,20*log10(abs((A+base_ortho(:,4).')*H)/5));
148
149 %legend('original','original+baseortogonal1','original+baseortogonal2
150 ','original+baseortogonal3','original+baseortogonal4')
151
152
153 end

```

## 8.1.7. Decisors

### 8.1.7.1. Decisor basat en centroides

```

1 centroids=entrenament(X_entrenament, Y_entrenament);
2 predict_val=predict(X_test, centroids);
3 correctes=0;
4 for k=1:length(Y_test)
5     if (predict_val(k)==Y_test(k))
6         correctes=correctes+1;
7     end
8 end
9
10 function res=entrenament(X_train, Y_train)
11     codificacions=4;
12     res=[];
13     for k=1:codificacions
14         pos=find(Y_train==k);
15         res(k)=sum(X_train(pos))/length(pos);
16     end
17 end
18
19 function predicted=predict(X_test, centres)
20     N=length(X_test);
21     D=0;
22     for k=1:N
23         d=distance(centres, X_test(k));
24         predicted(k)=find(d==min(d));
25     end
26 end
27 function d=distance(centres, X)
28     codificacions=4;
29     for k2=1:codificacions
30         %d(k2)=norm(X-centres(k2));
31         d(k2)=norm(X/abs(X)-centres(k2)/abs(centres(k2)));
32     end
33 end

```

### 8.1.7.2. Decisor SVM

```

1 t = templateSVM('Standardize',true,'SaveSupportVectors',true,
2 'KernelFunction','rbf', 'IterationLimit', 1e4);
3 predictorNames = {'Real','Imaginary'};
4 responseName = 'Simbol';
5 classNames = {'1','2','3','4'}; % Specify class order
6 Mdl = fitcecoc([real(X_entrenament); imag(X_entrenament) ].',
7 Y_entrenament.', 'Learners',t,'ResponseName',responseName, ...
8 'PredictorNames',predictorNames,'ClassNames',classNames);
9
10 [scores1,scores2] = predict(Mdl,[real(X_test); imag(X_test) ].');
11 predicted= str2num(cell2mat(scores1));
12
13
14 correctes=0;
15 for k=1:length(Y_test)
16     if (predicted(k)==Y_test(k))
17         correctes=correctes+1;
18     end

```

19 end

## 8.1.8. Codi de la modulació direccional teòrica enviant cadena de text amb decisor centroides

```
1 text='cadena de text';
2
3 text_to_ascii=unicode2native(text); % convertir cadena de text en
4 codi_ascii
5 text_to_binari=int2bit(text_to_ascii,10).' ; %passem a un binari de 8
6 bits
7
8 [numRow, numCol]=size(text_to_binari);
9 % convertim el codi binari en símbols per enviar amb una QPSK.<
10 for k=1:numRow
11     for l=1:numCol/2
12
13         text_to_simb((k-1)*4+l)=2*(text_to_binari(k, (l-
14 1)*2+1))+(text_to_binari(k, (l-1)*2+2))+1;
15
16     end
17 end
18
19 n_radians=5;
20 th=0:180;
21 for k=1:n_radians
22     H1(k,:)=exp(1i*(3-k)*pi*cosd(th));
23
24 end
25
26
27 theta_0=45;
28
29 [W,P] = calcular_ortogonals(H1, theta_0);
30
31
32 QPSK_x=linspace(-sqrt(2)/2, sqrt(2)/2, sqrt(4));
33 QPSK_y=QPSK_x;
34 [QPSK_YY,QPSK_XX]=meshgrid(QPSK_y,QPSK_x); %creem els símbols de la
35 QPSK (constelació 4)
36 simb_TX=QPSK_XX+1*i*QPSK_YY;
37 simb_TX=simb_TX(:);%Available symbols (símbols disponibles)
38 Y_entrenament=randi(4,1,1000);
39
40
41 s_ini=simb_TX(Y_entrenament);%random symbols to transmit (-2 que els
42 dos primers son per detectar), tria aleatoriament entre símbols a
43 detectar
44 simb=[s_ini];
45 simb=simb(:);%
46
47
48 %W=[W;W];
49
50
51 %w=[W(:,1).'; W(:,1).'; W(:,1).'; W(:,1).'];
```

```
52 w=zeros(length(simb),n_radians);
53
54
55 for k=1:length(simb)
56     [numRows_w, numCol_w]=size(W);
57     pesos=randn(numCol_w,1);
58
59     w(k,:)=W*pesos;
60
61 end
62
63 X1=simb*P;
64 X=X1+w;
65
66 %Y=zeros(180,length(simb));
67 %for k=(th+1)
68 %    H_canal=conj(H1(:,k).');
69 %    Y(k,:)=H_canal*(X.');
70
71 %end
72
73
74 H_canal=conj(H1(:,theta_0+20).');
75 X_entrenament=H_canal*(X.');
76
77 %scatterplot(YY)
78
79 centroids=entrenament(X_entrenament, Y_entrenament);
80
81 %Y_test=randi(4,1,1000000);
82 Y_test=text_to_simb;
83
84 s_ini=simb_TX(Y_test);
85
86 simb=[s_ini];
87 simb=simb(:);%
88
89
90 w=zeros(length(simb),n_radians);
91
92
93 for k=1:length(simb)
94     [numRows_w, numCol_w]=size(W);
95     pesos=randn(numCol_w,1);
96
97     w(k,:)=W*pesos;
98
99 end
100
101 X1=simb*P;
102 X=X1+w;
103
104
105 H_canal=conj(H1(:,theta_0+20).');
106 X_test=H_canal*(X.');
107
108
109 predict_val=predict(X_test, centroids);
110 correctes=0;
111
112
```

```

113  for k=1:length(Y_test)
114      if (predict_val(k) == Y_test(k))
115          correctes=correctes+1;
116      end
117  end
118
119
120
121 binari_predict = de2bi( predict_val-1,2 );
122
123
124
125 binari_predict_orde=binari_predict(:,end:-1:1);
126 binari_ascii=zeros(length(binari_predict_orde)/4,8);
127
128 for k=1:length(binari_predict_orde)/4
129     binari_ascii(k,:)=[binari_predict_orde(4*(k-1)+1,:);
130 binari_predict_orde(4*(k-1)+2,:); binari_predict_orde(4*(k-1)+3,:);
131 binari_predict_orde(4*(k-1)+4,:)];
132 end
133 decVal = binaryVectorToDecimal(binari_ascii);
134 cadena_rebuda = convertCharsToStrings(char(decVal));
135
136 function [Ws,P] = calcular_ortogonals(H, TH_0)
137     A = H(:,TH_0+1)';
138     %Ha = A * H;
139     th=0:180;
140
141     base_orto=null(A);
142     ori = (A)*H;
143     W1 = (A+base_orto(:,1).')*H;
144     Ws = base_orto;
145     W3 =(A+base_orto(:,3).')*H;
146     P = H(:,TH_0+1).'/norm(H(:,TH_0+1));
147
148 end
149
150 function res=entrenament(X_train, Y_train)
151     codificacions=4;
152     res=[];
153     for k=1:codificacions
154         pos=find(Y_train==k);
155         res(k)=sum(X_train(pos))/length(pos);
156     end
157 end
158
159 function predicted=predict(X_test, centres)
160     N=length(X_test);
161     D=0;
162     for k=1:N
163         d=distance(centres, X_test(k));
164         predicted(k)=find(d==min(d));
165     end
166 end
167
168 function d=distance(centres, X)
169     codificacions=4;
170     for k2=1:codificacions
171         %d(k2)=norm(X-centres(k2));
172         d(k2)=norm(X/abs(X)-centres(k2)/abs(centres(k2)));
173     end

```

174 end

## 8.1.9. Codi percentatge símbols correctes teòrics

```
1 text='cadena de text';
2
3 text_to_ascii=unicode2native(text); % convertir cadena de text en
4 codi ascii
5 text_to_binari=int2bit(text_to_ascii,8).'; %passem a un binari de 8
6 bits
7
8 [numRow, numCol]=size(text_to_binari);
9 % convertim el codi binari en símbols per enviar amb una QPSK.<
10 for k=1:numRow
11     for l=1:numCol/2
12
13         text_to_simb((k-1)*4+1)=2*(text_to_binari(k,(l-
14 1)*2+1))+(text_to_binari(k,(l-1)*2+2))+1;
15
16     end
17 end
18
19 n_radians=5;
20 th=0:180;
21 for k=1:n_radians
22     H1(k,:)=exp(1i*(3-k)*pi*cosd(th));
23
24 end
25
26
27 theta_0=45;
28 direccio=0;
29
30 [W,P] = calcular_ortogonals(H1, theta_0);
31
32
33 QPSK_x=linspace(-sqrt(2)/2, sqrt(2)/2, sqrt(4));
34 QPSK_y=QPSK_x;
35 [QPSK_YY,QPSK_XX]=meshgrid(QPSK_y,QPSK_x); %creem els símbols de la
36 QPSK (constelació 4)
37 simb_TX=QPSK_XX+1*i*QPSK_YY;
38 simb_TX=simb_TX(:).' ;%Available symbols (símbols disponibles)
39 Y_entrenament=randi(4,1,1000);
40
41
42 s_ini=simb_TX(Y_entrenament);%random symbols to transmit (-2 que els
43 dos primers son per detectar), tria aleatoriament entre símbols a
44 detectar
45 simb=[s_ini];
46 simb=simb(:);%
47
48
49 %W=[W;W];
50
51
52 %w=[W(:,1).'; W(:,1).'; W(:,1).'; W(:,1).'];
```

```

53 w_entrenament=zeros(length(simb),n_radians);
54
55
56 for k=1:length(simb)
57     [numRows_w, numCol_w]=size(W);
58     pesos=0.25*randn(numCol_w,1);
59
60     w_entrenament(k,:)=W*pesos;
61
62 end
63
64
65 X1=simb*P;
66 X=X1+w_entrenament;
67
68 %Y=zeros(180,length(simb));
69 %for k=(th+1)
70 %    H_canal=conj(H1(:,k).');
71 %    Y(k,:)=H_canal*(X.');
72
73
74 %end
75
76 Y_test=randi(4,1,100000);
77 %Y_test=text_to_simb;
78
79 s_ini_test=simb_TX(Y_test);
80
81 simb_test=[s_ini_test];
82 simb_test=simb_test(:);%
83
84 w_test=zeros(length(simb_test),n_radians);
85
86
87 for k=1:length(simb_test)
88     [numRows_w, numCol_w]=size(W);
89     pesos=0.25*randn(numCol_w,1);
90
91     w_test(k,:)=W*pesos;
92
93 end
94
95 X1_test=simb_test*P;
96 X2_test=X1_test+w_test;
97
98
99 H_canal=conj(H1(:,theta_0+direccio+1).');
100 vector_correctes=zeros(181,1);
101
102 for t=0:180
103     H_canal=conj(H1(:,t+1).');
104     X_entrenament=H_canal*(X.');
105     centroids=entrenament(X_entrenament, Y_entrenament);
106     X_test=H_canal*(X2_test.');
107     predict_val=predict(X_test, centroids);
108     correctes=0;
109
110
111     for k=1:length(Y_test)
112         if (predict_val(k) == Y_test(k))
113             correctes=correctes+1;

```

```

114      end
115    end
116    vector_correctes(t+1)=correctes/length(Y_test)*100;
117  end
118
119
120 plot(th, vector_correctes)
121
122 function [Ws,P] = calcular_ortogonals(H, TH_0)
123   A = H(:,TH_0+1)';
124   %Ha = A * H;
125   th=0:180;
126
127   base_orto=null(A);
128   ori = (A)*H;
129   W1 = (A+base_orto(:,1).')*H;
130   Ws = base_orto;
131   W3 =(A+base_orto(:,3).')*H;
132   P = H(:,TH_0+1).'/norm(H(:,TH_0+1));
133
134 end
135
136 function res=entrenament(X_train, Y_train)
137   codificacions=4;
138   res=[];
139   for k=1:codificacions
140     pos=find(Y_train==k);
141     res(k)=sum(X_train(pos))/length(pos);
142   end
143 end
144
145 function predicted=predict(X_test, centres)
146   N=length(X_test);
147   D=0;
148   for k=1:N
149     d=distance(centres, X_test(k));
150     predicted(k)=find(d==min(d));
151   end
152 end
153
154 function d=distance(centres, X)
155   codificacions=4;
156   for k2=1:codificacions
157     %d(k2)=norm(X-centres(k2));
158     d(k2)=norm(X/abs(X)-centres(k2)/abs(centres(k2)));
159   end
160 end

```

## 8.1.10. Codi transmetre modulació direccional

```

1 %% SDR Configuration
2 %We configure the IP and type of SDR we are using
3 if ~exist('prmToneRx','var')
4   prmToneRx.SDRDeviceName = 'FMCOMMS5';
5   prmToneRx.IPAddress = '158.109.64.241';
6 end

```

```

7 % To update the example for FMCOMMS5, set |prmToneRx.SDRDeviceName|
8 to
9 % |'FMCOMMS5'|.
10
11
12 %% Transmitter Configuration
13 % Create a transmitter System object(TM) to configure the RF card
14 settings. Set
15 % the RF card to transmit data at a center frequency of 2.4 GHz.
16
17 RadioBasebandRate = 15.36e6;
18 CenterFrequency = 3.6e9;
19 %ToneFrequency = 25e3;
20
21 sdrTransmitter = sdrtx(prmToneRx.SDRDeviceName, ...
22     'IPAddress', prmToneRx.IPAddress, ...
23     'CenterFrequency', CenterFrequency, ...
24     'BasebandSampleRate', RadioBasebandRate, ...
25     'ChannelMapping', [1 2 3 4], ...
26     'ShowAdvancedProperties', true);
27 %% Set Information to send with Directional Modulation
28 %H_adj=[-0.0411 - 0.0278i -0.0608 - 0.0010i -0.0207 + 0.0553i -
29 0.0405 + 0.0370i];
30 %H_adj=[ -0.0404 - 0.0234i -0.0567 + 0.0047i -0.0159 + 0.0523i -
31 0.0331 + 0.0350i];
32 %H_adj=[ -0.0521 + 0.0003i -0.0516 + 0.0371i -0.0223 + 0.0588i -
33 0.0376 + 0.0361i];
34
35 %Vector de canal, calculat amb "Enviardades.m"
36 H_adj=[ -0.0517 - 0.0024i -0.0505 + 0.0317i 0.0212 - 0.0571i
37 0.0399 - 0.0345i];
38
39
40 % Calculem paràmetres per la DM
41 [base_W, P]=calcular_ortogonals(H_adj);
42
43 %% Create the symbols (information) to send with the transceiver
44 fs=15.36e6;%Sampling frequency
45 Ts=1/fs;%Sampling period
46 n_simb=500000;% # symbols per frame (include 2 0s at the beginning)
47 l_s=10; % # samples in a symbol
48 nframe=3;% # frames
49
50 M=4; %QPSK = 4, 16-QAM = 16
51 %Generem els axis dels simbols a enviar
52 X=linspace(-sqrt(2)/2,sqrt(2)/2,sqrt(M));% Generem 2 (^4) punts entre
53 -^2/2 i ^2/2
54 Y=X;
55 [YY,XX]=meshgrid(Y,X); %creem els simbols de la QPSK (constelaciÃ³ 4)
56 simb_TX=XX+1i*YY;
57 simb_TX=simb_TX(:).';%Available symbols (symbols disponibles)
58
59 n_radians=4; %nombre transmissors
60 lframe=n_simb*l_s*nframe;% length frame in samples = (n simb/frame)*
61 (mostres/simbol)
62
63
64
65
66 s_ini=simb_TX(randi(M,1,n_simb-2));%random symbols to transmit (-2
67 que els dos primers son per detectar), tria aleatoriament entre

```

```

68  simbols a detectar
69  %s_ini=simb_TX([1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
70  4 1 2
71  %3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
72  1 2 3
73  %4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1
74  2 3 4
75  %1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2
76  3 4 1
77  %2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3
78  4 1 2
79  %3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
80  1 2 3
81  %4 1 2 3 4 1 2 3 4]);%enviem
82
83
84
85
86
87  simb=repmat([0 0 s_ini],l_s,1);
88  simb=simb(:);%Enlarge pulses to avoid distortion
89
90  %% Apply DM to the information to send
91
92  w=zeros(length(simb),n_radians);
93  for k=1:length(simb)
94      [numRows_w, numCol_w]=size(base_W);
95      pesos=1*(randn(numCol_w,1)+1i*randn(numCol_w,1));
96
97      w(k,:)=base_W*pesos;
98
99  end
100
101 X1=simb*P;
102 X=X1+w;
103
104 %% Send Information
105 transmitRepeat(sdrTransmitter, X );
106
107
108 %% Receiver configuration
109 % To capture the RF tone signal into MATLAB create an SDR receiver
110 System
111 % object and configure it to receive the samples at the baseband
112 rate.
113
114 % Radio parameters
115 RadioFrameLength = n_simb*l_s*1;
116
117 % Create a receiver System object with desired radio parameters
118 sdrReceiver = sdrrx(prmToneRx.SDRDeviceName, ...
119     'IPAddress', prmToneRx.IPAddress, ...
120     'CenterFrequency', CenterFrequency, ...
121     'BasebandSampleRate', RadioBasebandRate, ...
122     'GainSource', 'Manual', ...
123     'Gain', 32, ...
124     'SamplesPerFrame', RadioFrameLength, ...
125     'ChannelMapping', 1, ...
126     'OutputDataType', 'double');
127
128

```

```

129 %% Configurate to analize received data
130
131 %spectrumScope = dsp.SpectrumAnalyzer('SampleRate',
132 RadioBasebandRate);
133 %timeScope =
134 dsp.TimeScope('TimeSpan',lframe,'SampleRate',RadioBasebandRate);
135
136 %constellation =
137 comm.ConstellationDiagram('ShowReferenceConstellation', false);
138
139 % Create a container for the three scopes
140 %scopesContainer = HelperCreateScopesContainer(...%
141 % {spectrumScope,timeScope,constellation},...
142 % 'Name','Zynq Radio QPSK Receiver',...
143 % 'Layout',[2 2],...
144 % 'ExpandToolbar',false);
145 %scopesContainer.setColumnSpan(1,1,2);
146
147 %%
148
149
150 %stop_time=nframe*lframe*Ts;
151 StopTime = lframe*Ts; %4 antenes
152 % seconds
153 RadioFrameTime = StopTime;% (RadioFrameLength / RadioBasebandRate); %
154 seconds
155
156
157 numFramesinBurst = ceil(RadioBasebandRate*StopTime/RadioFrameLength);
158 sdrReceiver.EnableBurstMode = true;
159 sdrReceiver.NumFramesInBurst = numFramesinBurst;
160 %% Receive Data a process it
161 % Capture and visualize data by calling the corresponding System
162 % objects in a loop, every iteration of the loop reads a frame.
163 simbinicial=zeros(n_simb-2,nframe); %simbol inicial en le sposicions
164 1 i 2 és 0 (no el tenim en compte)
165 simbfinal=zeros((n_simb-2)*nframe,1);
166
167
168 %Convolució circular amb tot 0s, excepte dos*ls 1s seguits, aquests
169 multiplicats per 0 dels simbols ens indicarab la posició mínima,
170 refconv=zeros(lframe/nframe,1);
171 refconv(1:(2*l_s))=1;
172 try
173     % Loop until the example reaches the target stop time.
174     timeCounter = 0;
175     num_frame=1;
176     while timeCounter < StopTime
177
178         [data, valid, overflow] = sdrReceiver();
179         if (overflow > 0) && (timeCounter > 0)
180
181             warning(message('sdrpluginbase:zynqradiotransmitter:DroppedSamples'));
182         end
183
184         if valid
185             % Visualize frequency spectrum
186             %spectrumScope(data);
187             % Visualize in time domain
188             %timeScope([real(data), imag(data)]);
189             % Visualize the scatter plot

```

```

190             %constellation(data);
191
192
193
194
195 simbinicial(1:end,num_frame)=simb((2*l_s+(l_s/2)):l_s:end); %Llegeixo
196 el punt mig dels simbols inicials
197
198
199 xxxx=cconv(abs(data),refconv(1:(lframe/nframe)),lframe/nframe);
200 %convolució circular
201         %plot(simb)
202         %plot(data)
203         %plot(xxx)
204
205 pos_min=find(xxx==min(xxx)); %trobem inici
206 pos_min=mod(pos_min+1-2*l_s,lframe);
207 zzz=circshift(data,-pos_min); % rto circularment per
208 tenir l'inici al principi
209 vvv=zzz((2*l_s):end); %Llegeixo punt mig del senyal rebut
210         simbfinal((n_simb-2)*(num_frame-1)+1:(n_simb-
211 2)*num_frame) =vvv((l_s/2):l_s:end);
212
213
214
215
216 % Set the limits in scopes
217 %dataMaxLimit = max(max(abs([real(data); imag(data)])));
218 %constellation.XLimits = [-dataMaxLimit*1.5,
219 dataMaxLimit*1.5];
220 %constellation.YLimits = [-dataMaxLimit*1.5,
221 dataMaxLimit*1.5];
222 %timeScope.YLimits = [-dataMaxLimit*2, dataMaxLimit*2];
223 timeCounter = timeCounter + RadioFrameTime;
224 num_frame = num_frame + 1;
225 end
226
227 end
228
229 catch ME
230     rethrow(ME);
231 end
232 %% Close SDR Transmitter/Receiver and visualization scopes.
233
234 release(sdrReceiver);
235 release(sdrTransmitter);
236
237 %release(spectrumScope);
238 %release(timeScope);
239 %release(constellation);
240
241 %% Funció per calcular base ortogonal pesos, i vector P
242
243 function [Ws,P] = calcular_ortogonals(H)
244     Ws=null(conj(H.') .');
245     P = H/norm(H);
246 end

```

### 8.1.11. Codi percentatge símbols correctes amb algoritme centroides

```
1 vector_correctes=zeros(181,1);
2
3 for t=1:181
4
5     X_entrenament=dades_rebudes(1:10000,t);
6     Y_entrenament=s(1:10000,t);
7     centroids=entrenament(X_entrenament, Y_entrenament);
8     X_test=dades_rebudes(10001:400000,t);
9     predict_val=predict(X_test, centroids);
10    Y_test=s(10001:400000,t);
11    correctes=0;
12
13
14    for k=1:length(Y_test)
15        if (predict_val(k) == Y_test(k))
16            correctes=correctes+1;
17        end
18    end
19    vector_correctes(t)=correctes/length(Y_test)*100;
20 end
21
22
23 plot(-90:90, vector_correctes)
24 %scatterplot(YY)
25
26
27
28 function res=entrenament(X_train, Y_train)
29     codificacions=4;
30     res=[];
31     for k=1:codificacions
32         pos=find(Y_train==k);
33         res(k)=sum(X_train(pos))/length(pos);
34     end
35 end
36
37 function predicted=predict(X_test, centres)
38     N=length(X_test);
39     D=0;
40     for k=1:N
41
42         d=distance(centres, X_test(k));
43         selected=find(d==min(d));
44
45         predicted(k)=selected(1);
46     end
47 end
48
49
50 function d=distance(centres, X)
51     codificacions=4;
52     for k2=1:codificacions
53         %d(k2)=norm(X-centres(k2));
54         if abs(X)==0
55             X=0;
```

```

56         d(k2)=norm(X-centres(k2)/abs(centres(k2)));
57     else
58         d(k2)=norm(X/abs(X)-centres(k2)/abs(centres(k2)));
59     end
60 end
61 end

```

### 8.1.12. Codi percentatge símbols correctes amb algoritme SVM

```

1 vector_correctes=zeros(181,1);
2
3 for t=1:181
4
5 X_entrenament=dades_rebudes(1:10000,t);
6 Y_entrenament=s(1:10000,t);
7
8 X_test=dades_rebudes(10001:400000,t);
9 X_entrenament2=[real(X_entrenament), imag(X_entrenament) ];
10
11 Y_test=s(10001:400000,t);
12
13 svm = templateSVM('Standardize',true,'SaveSupportVectors',true,
14 'KernelFunction','rbf', 'IterationLimit', 1e4);
15 predictorNames = {'Real','Imaginary'};
16 responseName = 'Simbol';
17 classNames = {'1','2','3','4'}; % Specify class order
18 Mdl = fitcecoc(X_entrenament2,
19 Y_entrenament.', 'Learners', svm, 'ResponseName',responseName, ...
20 'PredictorNames',predictorNames,'ClassNames',classNames);
21
22 [scores1,scores2] = predict(Mdl,[real(X_test), imag(X_test) ]);
23 predicted= str2num(cell2mat(scores1));
24
25
26
27 correctes=0;
28
29
30 for k=1:length(Y_test)
31     if (predicted(k) == Y_test(k))
32         correctes=correctes+1;
33     end
34 end
35 vector_correctes(t)=correctes/length(Y_test)*100;
36 end
37
38
39 plot(-90:90, vector_correctes)
40 %scatterplot(YY)

```