

Treball Final de Màster

Disseny i simulació de sistemes 4D OFDM òptics amb constel·lacions multinivell

Autor: Josep Antoni Solis Viladegut

Director del projecte: Josep Maria Fàbrega Sánchez

Màster Universitari d'Enginyeria de Telecomunicació (UOC-URL)



Els continguts d'aquesta obra estan subjectes a una llicència de Reconeixement No comercial - Sense obres derivades 3.0 de Creative Commons. Se'n permet la reproducció, distribució i comunicació pública sempre que se'n citi l'autor i no se'n faci un ús comercial.

La llicència completa es pot consultar a:

<http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>

B) GNU Free Documentation License (GNU)

Copyright © 2016 José Antonio Solís Viladegut.

1 Fitxa del Treball Final de Màster

Títol del treball:	<i>Disseny i simulació de sistemes 4D OFDM òptics amb constel·lacions multinivell</i>
Nom de l'autor:	<i>Josep Antoni Solis Viladegut</i>
Nom del consultor:	<i>Josep Maria Fàbrega Sánchez</i>
Data de lliurament:	<i>06/2016</i>
Àrea del Treball Final:	<i>Sistemes de comunicació òptics</i>
Titulació:	<i>Màster Universitari d'Enginyeria de Telecomunicacions (UOC-URL)</i>

Resum del Treball:

Per a transmetre informació a molt alta velocitat (>100 Gb/s) i de forma eficient en termes d'aprofitament d'espectre i d'energia, recentment s'ha proposat estendre els formats de modulació tradicionals cap a constel·lacions i trames que es puguin mapejar en les 4 dimensions possibles dels senyals òptics (amplitud, fase i dos estats de polarització ortogonals).

En aquest treball fi de màster es pretén dissenyar i avaluar un sistema de modulació 4D proposat en l'article Power-Efficient Modulation Formats in Coherent Transmission Systems (Agrell, Karlsson – 2009) i que els autors anomenen 6P-QPSK, incloent també la possibilitat d'emprar OFDM òptic. En una primera fase es proposa un algorisme per realitzar una adaptació de l'espai de modulacions 4D a un sistema representable per coordenades complexes. Posteriorment s'avalua la seva implementació realitzant simulacions del model implementat en Matlab, comparant els resultats amb els obtinguts amb els d'una constel·lació simple (C1, DP-QPSK). També s'implementa la modulació PS-QPSK (C2) seguint la proposta de mapatge que fan els autors pel conjunt C2 i es compara els resultats obtinguts amb els anteriors. Finalment, s'exposen les conclusions de l'estudi i es proposen futures vies d'investigació.

Abstract:

To transmit information at very high speed (> 100 Gb / s) and in terms of efficient use of spectrum and energy, it has been recently proposed to extend the modulation formats towards patterns and constellations that can be mapped in four possible dimensions of optical signals (amplitude, phase and two orthogonal polarization states).

This work aims to design and evaluate a modulation system proposed in the article "4D Power-Efficient Modulation Formats in Coherent Transmission Systems" (Agrell, Karlsson - 2009), named by the authors as 6P-QPSK, and including its possible implementation towards optical OFDM. In the first phase, it is proposed an algorithm to perform the adaptation of the 4D space modulation system represented by complex coordinates. Then the implementation of simulations models implemented in

Matlab is evaluated, comparing the results with those obtained with a simple constellation (C1, DP-QPSK). Also it is implemented the PS-QPSK modulation (C2), following the proposal made by authors for mapping the set C2 and the results are compared with those obtained in the past years. Finally, the conclusions are drawn from the study and future lines of research are proposed.

Paraules clau:

OFDM, modulació 4D, detecció coherent, DP-QPSK, 6P-QPSK, PS-QPSK, eficiència espectral.

2 Agraïments

Al director del projecte Josep Maria Fàbrega Sanchez, pel seu ajut i pels seus erudits consells.

Als meus fills i a la meva esposa Neus, per haver sofert la meva afició per l'estudi.

Als meus companys de feina, particularment al meu cap Pere Josep, per haver patit les meves absències aquestes darreres setmanes.

A l'Anna, per la seva correcció al redactat de la memòria.

3 Contingut

1	Fitxa del Treball Final de Màster.....	3
2	Agraïments.....	5
3	Contingut	6
4	Llista de figures	8
5	Context i justificació del Treball	11
5.1	Objectius del Treball.....	12
5.2	Enfocament i mètode seguit.....	13
5.3	Planificació del Treball	14
5.4	Productes obtinguts.....	15
5.5	Estructura de la memòria	15
6	Conceptes bàsics: Sistemes de transmissió òptics	16
6.1	La fibra òptica com a mitjà de transmissió	16
6.2	Pèrdues en els sistemes de comunicacions òptiques	17
6.2.1	Causes intrínseques.....	17
6.2.2	Causes extrínseques i balanç de potència.....	18
6.3	Dispersió en els sistemes de comunicacions òptiques.....	19
6.3.1	Dispersió modal	19
6.3.2	Dispersió cromàtica	20
6.4	Dispersió de mode de polarització (PMD).....	21
6.5	Efectes no lineals	21
7	Modulació òptica i sistemes de transmissió OFDM òptics	22
7.1	Principis de funcionament.....	22
7.1.1	Paràmetres de Stokes i esfera de Poincaré	22
7.1.2	Tècniques de modulació avançada.....	24
7.2	Conceptes bàsics del OFDM.....	25
7.3	Processat de la senyal en sistemes OFDM	26
7.3.1	Símbols d'entrenament	27
7.3.2	Transformada Inversa de Fourier (IFFT)	27
7.3.3	Prefix cíclic	28
7.4	Modulació de la senyal	28
7.5	Detecció coherent.....	30
7.6	Recepció: Equalització i estimació de canal	30
8	Modulació 4D multinivell.....	31
8.1	Modulació DP-QPSK.....	31
8.2	Proposta de modulació multinivell: 6P-QPSK.....	33
8.2.1	Descripció	33

8.2.2	Mapping 6-PQPSK	34
8.2.3	Adaptació dels moduladors I/Q al 6P-QPSK	35
8.2.4	BER i SER del nou escenari	35
8.3	Modulació PS-QPSK	37
9	Implementació	38
9.1	Blocs del sistema.....	39
9.2	Paràmetres de la simulació OFDM	40
9.3	Paràmetres de la fibra òptica i del làser	41
9.4	Codificació.....	42
9.4.2	Mapatge PS-QPSK i implementació.....	45
9.5	Transmissió per la fibra i etapa de recepció.....	46
9.5.1	Algorisme de <i>demapping</i> 6P-QPSK i implementació.....	46
9.5.2	Algorisme de <i>demapping</i> PS-QPSK i implementació.....	49
9.5.3	Implementació de l'efecte PMD en la fibra	49
10	Simulacions	51
10.1	Proves <i>Back-to-Back</i> sense fibra.....	52
10.2	Prova <i>Back-to-Back</i> amb enllaç òptic.....	59
10.3	Abast del sistema	60
10.4	Impacte de la dispersió del mode de polarització en enllaços de llarga distància	68
11	Conclusions	71
12	Propostes d'investigació	72
13	Glossari.....	73
14	Bibliografia.....	75
15	Annexes	77

4 Llista de figures

Figura 1: comparació dels diferents esquemes de modulació òptics	11
Figura 2: fites a assolir durant la elaboració del projecte	14
Figura 3: diagrama de Gantt proposat	15
Figura 4: estructura d'una fibra òptica.....	17
Figura 5: corba d'atenuació de la fibra òptica.....	18
Figura 6: fenomen de dispersió multimode	20
Figura 7: endarreriment del mode en el fenomen de PMD.....	21
Figura 8: esfera de Poincaré	23
Figura 9: representació geomètrica de la senyal.....	24
Figura 10: codificació bit a símbol emprades en diferents modulacions	25
Figura 11: ones ortogonalment polaritzades.....	25
Figura 12: ocupació temps-freqüència de la modulació OFDM.....	26
Figura 13: bloc corresponent al processat digital de la senyal.....	26
Figura 14: procés de realització del IFFT.....	27
Figura 15: vector d'entrada de l'algoritme FFT	28
Figura 16: prefix cíclic	28
Figura 17: esquema de l'interferòmetre Mach-Zehnder estàndard	29
Figura 18: esquema del modulador IQ-MZM	29
Figura 19: Receptor coherent utilitzant fotodetecció balancejada	30
Figura 20: bloc del processament digital en recepció.....	31
Figura 21: valors de fase emprats per la modulació DP-QPSK. Els eixos diagonals mostren les fases absoluta i relativa.....	32
Figura 22: nou escenari de transformació proposat	33
Figura 23: mapatge 6P-QPSK proposat.....	34
Figura 24: esquemes de transmissors proposats basats en moduladors IQ.....	35
Figura 25: SER versus E_b/N_0 per les modulacions DP-QPSK, PS-QPSK i 6P-QPSK.....	36
Figura 26: BER versus E_b/N_0 per les modulacions DP-QPSK, PS-QPSK i 6P-QPSK	37
Figura 27: processos que intervenen en la transmissió ODFM 4D	38
Figura 28: funcions i relacions de l'escenari de simulació	40
Figura 29: paràmetres de transmissió de Simulacio1.m	52
Figura 30: espectre transmès i rebut en la Simulacio1.m	53
Figura 31: constel·lacions rebudes en la Simulacio1.m	53
Figura 32: valors de BER obtinguts en la Simulacio1.m.....	54
Figura 33: valors numèrics de BER obtinguts en la Simulacio1.m	54
Figura 34: paràmetres de transmissió de Simulacio2.m	55
Figura 35: BER vs OSNR i constel·lacions rebudes per Simulacio2.m i 64 subportadores.....	55

Figura 36: BER vs OSNR i constel·lacions rebudes per Simulacio2.m i 256 subportadores	55
Figura 37: BER vs OSNR i constel·lacions rebudes per Simulacio2.m i 512 subportadores	55
Figura 38: BER vs OSNR de Simulacio2.m per 64, 256 i 512 subportadores	56
Figura 39: valors numèrics de BER obtinguts en la Simulacio2.m	56
Figura 40: OSNR per a un BER de $2e-2$	56
Figura 41: paràmetres de transmissió de Simulacio3.m	57
Figura 42: valors de BER obtinguts per les modulacions 6P-QPSK, DP-QPSK i PS-QPSK amb 64 subportadores OFDM	57
Figura 43: valors numèrics de BER / SNR per 6P-QPSK, DP-QPSK i PS-QPSK.....	58
Figura 44: OSNR per a un BER de $2e-2$	58
Figura 45: constel·lacions 6P-QPSK, DP-QPSK i PS-QPSK rebudes	58
Figura 46: valors de BER (6P-QPSK, DP-QPSK i PS-QPSK) amb 64 subportadores OFDM normalitzat a bit/símbol.....	59
Figura 47: paràmetres de transmissió de Simulacio4.m,i Simulacio5.m i Simulacio11.m.....	61
Figura 48: resultat de la prova d'abast per modulacions 6P-QPSK, DP-QPSK, PS-QPSK i 512 subportadores OFDM	61
Figura 49: resultats de BER/Km de Simulacio4.m i Simulacio5.m per modulació 6P-QPSK i DP-QPSK i 512 subportadores OFDM	61
Figura 50: Km per a un BER de $2e-2$	61
Figura 51: paràmetres de transmissió de Simulacio6.m i Simulacio7.m	62
Figura 52: resultat de Simulacio6.m i Simulacio7.m per modulació 6P-QPSK i DP-QPSK i 256 subportadores OFDM	62
Figura 53: resultats de la prova d'abast per modulació 6P-QPSK, DP-QPSK, PS-QPSK i 256 subportadores OFDM	63
Figura 54: Km per a un BER de $2e-2$	63
Figura 55: paràmetres de transmissió de Simulacio8.m	64
Figura 56: constel·lacions rebudes en Simulacio8.m per 256 subportadores OFDM.....	64
Figura 57: constel·lacions rebudes en Simulacio8.m per 256 i 512 subportadores OFDM.....	64
Figura 58: valors de BER per modulació 6P-QPSK amb 256 i 512 subportadores OFDM i 50 trams de 100 Km.....	65
Figura 59: valors de BER per modulació 6P-QPSK amb 256 i 512 subportadores OFDM i 50 trams de 100 Km.....	65
Figura 60: OSNR per a un BER de $2e-2$	65
Figura 61: paràmetres de transmissió de Simulacio13.m	66
Figura 62: constel·lacions rebudes en Simulacio13.m per 256 subportadores OFDM (40 x 125 Km)	67
Figura 63: constel·lacions rebudes en Simulacio13.m per 512 subportadores OFDM (50 x 100 Km)	67
Figura 64: valors de BER per modulació PS-QPSK amb 256 i 512 subportadores OFDM i 40 trams de 125 Km.....	67
Figura 65: valors de BER per modulació PS-QPSK amb 256 sc. (40 x 125 km) i 512 sc (50 x 100 Km) 68	

Figura 66: OSNR per a un BER de $2e-2$	68
Figura 67: coeficients genèrics de PMD requerits per evitar una excessiva penalització.....	68
Figura 68: paràmetres de transmissió de Simulacio9.m	69
Figura 69: constel·lacions rebudes en Simulacio9.m i $PMD=0.02e-12/(10^{3/2})$	70
Figura 70: constel·lacions rebudes en Simulacio9.m i $PMD=0.1e-12/(10^{3/2})$	70
Figura 71: resultat de BER/OSNR per un enllaç de 200 Km, 256 sc OFDM i $PMD=0.02e-12/(10^{3/2})/0.1e-12/(10^{3/2})$	70
Figura 72: resultat numèric de BER/OSNR per un enllaç de 200 Km, 256 sc OFDM i $PMD=0.02e-12/(10^{3/2})/0.1e-12/(10^{3/2})$	71
Figura 73: OSNR per a un BER de $2e-2$	71

5 Context i justificació del Treball

Durant els darrers anys, el consum de dades en temps real produït per fenòmens com poden ser la televisió sota demanda o l'intercanvi de continguts multimèdia, així com l'augment d'usuaris i l'increment de les xarxes i la seva interconnexió, ha implicat un augment exponencial de les necessitats de velocitat dels sistemes de transmissió de dades.

Conseqüentment, els sistemes de transmissió de dades per fibra òptica han evolucionat i han passat d'oferir velocitats de 10 Gb/s en xarxes jeràrquiques a implementar-se en malles fotòniques amb sistemes DWDM i a velocitats de fins a 400 Gb/s. Quan la longitud de l'enllaç òptic es excessiva, es fa necessari la utilització d'amplificadors òptics que impliquen l'aparició d'efectes distorsionadors com el soroll ASE, font de soroll gaussià. La disminució de la OSNR que això implica, limita la taxa màxima de transmissió de dades per lo que es fa palesa la necessitat de millorar l'estratègia de modulació de dades per millorar la OSNR.

En aquest sentit, la modulació en doble polarització aporta una millor taxa de transmissió de símbols i una millor relació OSNR respecte a d'altres modulacions en polarització simple. Per exemple, en una modulació DP-QPSK podem modular quatre nivells de fase de dos senyals òptics amb polaritzacions ortogonals, per lo que podem codificar quatre bits per símbol. Tot plegat, ens permet reduir l'ample de banda necessari per transmetre les mateixes dades amb la conseqüent simplificació i abaratiment dels components òptics i electrònics i un millor resposta dels sistemes a nivell de BER i de OSNR.

En aquest punt, se'ns planteja la següent pregunta: podem augmentar el nombre de símbols o nivells de la constel·lació (representació en el pla complex dels estats del símbol en termes d'amplitud i fase) sense tenir que augmentar la potència de la senyal o reduir la distància entre els punts de la constel·lació? La resposta es que sí, augmentant els estats de polarització.

Veiem en la següent figura com les noves constel·lacions de fins a sis estats de polarització aconsegueixen una millor eficiència espectral.

Name	Nbr. of pts. M	Nbr. of dims. N	Pow. Eff. γ [dB]	Spectral Eff. [bits/symb/pol]	Sens. at $BER = 10^{-3}$ E_b/N_0 [dB]	Sens. at $BER = 10^{-9}$ E_b/N_0 [dB]
BPSK	2	1	0	2	6.8	12.5
QPSK	4	2	0	2	6.8	12.5
8-PSK	8	2	-3.57	3	10.0	16.2
8-QAM	8	2	-3.01	3	9.0	14.6
16-QAM	16	2	-3.98	4	10.5	16.6
DP-QPSK = C_1	16	4	0	2	6.8	12.5
PS-QPSK = C_2	8	4	1.76	1.5	5.8	11.0
6P-QPSK	$2^{9/2} = 22.6$	4	0.51	2.25	6.9	12.2
C_3	24	4	0.59	2.29	N/A	N/A
C_4	25	4	0.83	2.32	N/A	N/A

Figura 1: comparació dels diferents esquemes de modulació òptics

Si apliquem una modulació 4D multinivell en cadascun dels canals d'un sistema OFDM aconseguim un sistema robust, amb una alta capacitat de transmissió de dades, un ús eficient del espectre, una baixa taxa d'interferència entre símbols (ISI) i una reducció de la complexitat computacional associada a l'alta taxa de transmissió gràcies al ús de la transformada discreta de Fourier (FFT, IFFT).

Altrament, per a la recepció de la senyal modulada en una constel·lació multinivell es recomanable utilitzar un sistema de detecció coherent, ja que permet detectar de manera immediata i exacta l'estat de la fase i polarització de cada símbol, en contraposició als sistemes de detecció directa, on es requereix un processat més complex de la senyal.

Aquest treball pretén aprofundir en l'estudi de les modulacions 4D multinivell, prenen com a punt de partida els articles i estudis realitzats al respecte. En aquest sentit, un punt de partida inicial serà el Projecte Final de Màster en Enginyeria en Telecomunicacions realitzats per en David Sacie Alcazar anomenat *Simulación de sistemas CO-OFDM 4D ópticos*. En aquest treball es dissenya i modelitza un sistema OFDM òptic coherent on cada símbol és codificat mitjançant una constel·lació 4D (DP-QPSK,C1). Altrament, l'article inicial de referència serà l'article de Erik Agrell i Magnus Karlsson anomenat *Power-Efficient Modulation Formats in Coherent Transmission Systems*. La proposta que fan aquests autors és la d'utilitzar la constel·lació 6P-QPSK per millorar encara més la eficiència espectral i la constel·lació PS-QPSK més eficient a nivell de E_b/N_0 . L'objectiu d'aquest treball serà doncs el d'implementar i quantificar les prestacions de les propostes que fan aquests autors.

5.1 Objectius del Treball

El TFM s'anomena Disseny i simulació de sistemes 4D OFDM òptics amb constel·lacions multinivell. Els objectius del treball són els següents:

- Realitzar un anàlisi genèric dels sistemes OFDM òptics existents i de les diferents modulacions, fent especial èmfasi en el DP-QPSK i els sistemes 4D.
- Fer un anàlisi en profunditat dels sistemes 4D multinivell, centrant-nos en la constel·lació 6P-QPSK i PS-QPSK.
- Proposar una adaptació genèrica de l'espai de modulacions 4D a un sistema representable per coordenades complexes centrant-nos en la proposta que fan els autors de l'article objecte d'estudi per representar les constel·lacions 6P-QPSK i PS-QPSK caldrà cercar i dissenyar un mòdul adaptador/desadaptador real/complex que sigui escalable a multinivell.

- Realitzar una implementació o simulació d'un sistema OFDM coherent, on cada símbol es codificarà mitjançant una constel·lació 4D multinivell. El programari que empraré serà bàsicament el Matlab.
- Realitzar el disseny i implementació d'un banc de proves complet, on es pugui comparar els sistemes implementats amb els basats amb altres modulacions com el DP-QPSK proposat en el treball d'en David Sacie.
- Analitzar els resultats de les proves realitzades.
- Extreure'n conclusions.
- Proposar futurs camps d'estudi.

5.2 Enfocament i mètode seguit

En una primera fase es procedirà a l'estudi dels articles de referència:

- **Power-Efficient Modulation Formats in Coherent Transmission Systems** (Erik Agrell and Magnus Karlsson).
- **OFDM for Optical Communications** (Sander Lars Jansen).
- **Bandwidth-Variable Transceivers based on Four-Dimensional Modulation Formats** (Johannes Karl Fischer, Member, IEEE, Saleem Alreesh, Student Member, IEEE, Robert Elschner, Member, IEEE).
- **Digital transmission with coherent four-dimensional modulation** (GEORGE R. WELT1 AND JHONG S. LEE, MEMBER, IEEE).
- **Polarization QAM Modulation (POL-QAM) for Coherent Detection Schemes** (Henning Bülow).
- **Bandwidth-Variable Transceivers based on Four-Dimensional Modulation Formats** (Johannes Karl Fischer, Member, IEEE, Saleem Alreesh, Student Member, IEEE, Robert Elschner, Member, IEEE, Felix Frey, Markus Nölle, Carsten Schmidt-Langhorst, and Colja Schubert).

En la segona fase es procedirà a analitzar les constel·lacions multinivell (sense OFDM) i plantejar quins són els requisits del OFDM que hem d'implementar.

En una tercera fase s'implementarà el model a avaluar així com el joc de proves mitjançant el programari Matlab.

En una quarta fase es procedirà a avaluar els resultat de les proves.

En una cinquena es redactarà la memòria del projecte.

Finalment, en una sisena fase es realitzarà la presentació del projecte.

5.3 Planificació del Treball

A continuació es mostres les principals fites a assolir i el pla temporal de treball:

Nombre de la tasca	Duratiu	Smart	Finish	Predecessors
TFM Comunicacions Òptiques	99 days	Wed 24/2/16	Sun 10/7/16	
Inici quadrimestre	0 days	Wed 24/2/16	Wed 24/2/16	
Pla de treball	14 days	Thu 25/2/16	Tue 15/3/16	
Selecció tema de treball	8 days	Thu 25/2/16	Mon 7/3/16	
Selecció software necessari	8 days	Thu 25/2/16	Mon 7/3/16	
Diagrama de Gantt inicial	1 day	Mon 7/3/16	Mon 7/3/16	
Lliurament del pla de treball	0 days	Tue 15/3/16	Tue 15/3/16	4;5;6
Anàlisi I disseny	53 days	Wed 16/3/16	Sat 28/5/16	3
Estudi i investigació	6 days	Wed 16/3/16	Wed 23/3/16	
Definició de les funcionalitats	4 days	Thu 24/3/16	Tue 29/3/16	9
Disseny dels diferents mòduls	26 days	Wed 30/3/16	Wed 4/5/16	10
Lliurament anàlisi I disseny	1 day	Thu 5/5/16	Thu 5/5/16	11
Construcció I proves	19 days	Fri 29/4/16	Wed 25/5/16	
Implementació del disseny	5 days	Fri 29/4/16	Thu 5/5/16	
Implementació de les proves	4 days	Fri 6/5/16	Wed 11/5/16	14
Informe d'errors	5 days	Thu 12/5/16	Wed 18/5/16	15
Lliurament de la implementació	5 days	Thu 19/5/16	Wed 25/5/16	16
Memòria	15 days	Wed 1/6/16	Tue 21/6/16	13;8
Elaboració de la memòria	6 days	Wed 1/6/16	Wed 8/6/16	
Correcció i depuració	2 days	Thu 9/6/16	Fri 10/6/16	19
Lliurament de la memòria	2 days	Mon 13/6/16	Tue 14/6/16	20
Presentació	5 days	Fri 17/6/16	Thu 23/6/16	
Elaboració de la presentació	3 days	Fri 17/6/16	Tue 21/6/16	21
Lliurament de la presentació	0 days	Tue 21/6/16	Tue 21/6/16	23
Lliurament final del TFM	0 days	Tue 21/6/16	Tue 21/6/16	24

Figura 2: fites a assolir durant l'elaboració del projecte

Finalment, el diagrama de Gantt proposat és el següent:

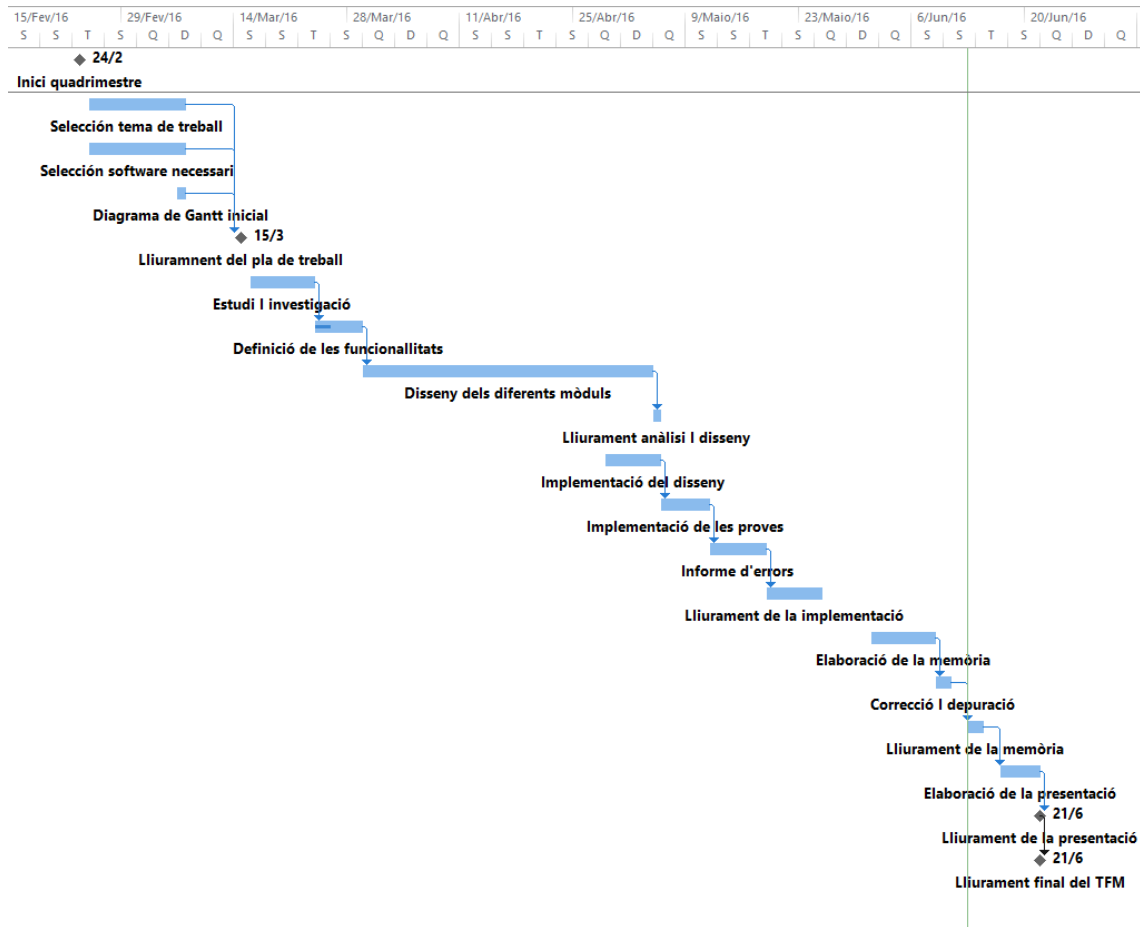


Figura 3: diagrama de Gantt proposat

5.4 Productes obtinguts

El projecte empara la implementació de *scripts* mitjançant el programari Matlab per poder modelitzar el funcionament de les modulacions 6P-QPSK / PS-QPSK i poder valorar el seu funcionament. Aquest procés de valoració es realitzarà analitzant els resultats obtinguts (a l'executar les funcions implementades) en forma de gràfiques de BER /OSNR, BER/Km, un llistat de valors numèrics obtinguts en les gràfiques i els espectres de la senyal rebuda en diferents escenaris. Part de les proves es realitzaran també emprant la modulació DP-QPSK per poder comparar els resultats obtinguts amb totes dues modulacions.

5.5 Estructura de la memòria

La memòria del projecte està estructurada en quinze capítols. Els quatre primers capítols contenen la fitxa del TFM, els agraïments, la descripció dels continguts i el llistat de figures. En el capítol

5 es descriu els objectius del TFM, la seva planificació i la metodologia emprada en el desenvolupament del projecte. Dins del capítol 6 s'introdueixen conceptes elementals referents als sistemes de comunicació òptics. En el capítol 7 s'inclouen aspectes iniciàtics referents a les modulacions emprades en els sistemes de comunicacions i dels sistemes ODFM. En el capítol 8 s'analitza la modulació 4D centrant l'anàlisi en la modulació 6P-QPSK i la modulació PS-QPSK. En el capítol 9 s'exposa la proposta d'implementació de les modulacions 6P-QPSK i PS-QPSK mitjançant el programari Matlab. En el capítol 10 s'exposa les diferents proves realitzades per avaluar la modulació 6P-QPSK i PS-QPSK. Dins del capítol 11 s'exposen les conclusions realitzades una vegada analitzat el resultat de les simulacions, de la tasca d'implementació de les modulacions 6P-QPSK / PS-QPSK i del seu estudi. En el capítol 12 es proposen noves vies d'investigació inspirades en les conclusions realitzades. Finalment, els tres darrers capítols corresponent al glossari, la bibliografia de referència i els annexos.

6 Conceptes bàsics: Sistemes de transmissió òptics

A continuació exposo els conceptes bàsics necessaris per comprendre els continguts i el desenvolupament dels treballs de recerca emparats pel treball final de màster.

6.1 La fibra òptica com a mitjà de transmissió

Els sistemes de comunicació òptics han millorat les prestacions de les xarxes de telecomunicacions gràcies a un medi físic capaç de transmetre dades a alta velocitat i a llargues distàncies. Els avantatges que han fet de la fibra òptica el medi de transmissió que permet velocitats de transmissió més grans són i), l'àmplia amplada de banda efectiva que té; ii) les baixes pèrdues de transmissió (a l'entorn de 0.2 dB/km); i, finalment, iii) que les transmissions guiades a través de fibra òptica no es veuen afectades per interferències electromagnètiques, com sí que succeeix en les transmissions realitzades en altres medis com poden ser el cable coaxial, el cable de parells trenats o les ones radioelèctriques.

La fibra òptica és una mena del fil de vidre compost per un nucli i un revestiment que tenen un índex de refracció diferent. L'índex de refracció del nucli és més alt que l'índex del revestiment, perquè la radiació pugui ser guiada per reflexió total en la interfície entre el nucli i el revestiment. D'aquesta manera, la radiació és guiada a través del nucli de la fibra òptica gràcies a les propietats físiques que regeixen la refracció i la reflexió d'una ona quan hi ha un canvi de medi (el nucli i el revestiment).

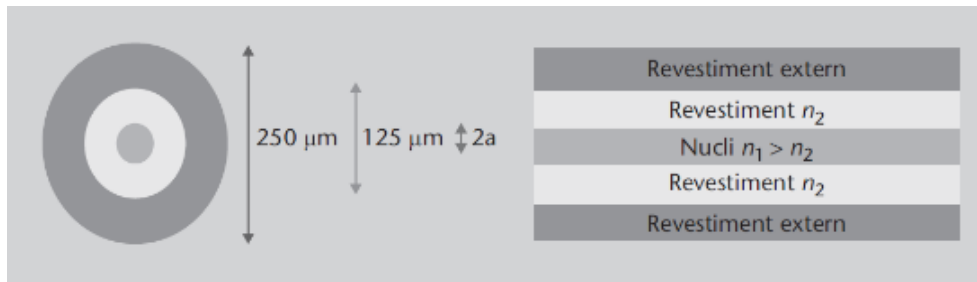


Figura 4: estructura d'una fibra òptica

Les fibres òptiques es poden distingir entre fibres monomode (*SMF*) i fibres multimode (*MMF*), en funció del nombre de modes que s'hi propaguen. En el cas que es propagui un sol mode, es tractarà d'una fibra monomode. En el cas que es propagui més d'un mode, llavors es denominarà fibra multimode. Els sistemes de transmissió òptics utilitzats en distàncies llargues empen fibres del tipus *SMF*, ja que respecte a les fibres *MMF* pateixen una menor dispersió temporal i l'amplada de banda de treball és molt major.

Per a poder determinar el nombre de modes i , per tant, poder discriminar entre fibres òptiques *SMF* i fibres *MMF*, el nombre de modes de propagació depèn d'un paràmetre que es denomina freqüència normalitzada (V), i que es defineix de la manera següent:

$$V = \frac{2\pi a}{\lambda} NA \quad (1)$$

La freqüència normalitzada depèn de les característiques físiques de la fibra ja que depèn de l'obertura numèrica i del diàmetre del nucli, per lo que podem afirmar que si $0 \leq V \leq 2,405$ llavors la fibra és *SMF*, mentre que si $V > 2,405$ llavors la fibra és *MMF*.

6.2 Pèrdues en els sistemes de comunicacions òptiques

Les pèrdues d'un enllaç de comunicació òptic són de tipus intrínsec i extrínsec. Les causes intrínseques són les pròpies del material amb el què es fabrica la fibra òptica (diòxid de silici), mentre que les causes extrínseques són les provocades per efectes externs, com curvatures de la fibra òptica o variacions geomètriques de la guia.

6.2.1 Causes intrínseques

Les principals causes intrínseques són la dispersió de Rayleigh, l'absorció d'infrarojos i la ressonància de l'ió OH. La dispersió de Rayleigh provoca un canvi de direcció d'alguns fotons del feix de llum que es propaga per la fibra òptica. Alguns d'aquests fotons poden prendre direccions que impossibilitaran el guiatge d'aquests per la fibra òptica. A efectes del senyal es provocarà una disminució de la quantitat de fotons que compleixen les condicions per a propagar-se i, per tant, hi haurà una pèrdua

o atenuació. Aquest efecte és inversament proporcional a la longitud d'ona. L'absorció d'infrarojos és deguda a les propietats òptiques del material (diòxid de Silici), ja que aquest absorbeix la llum a partir de 1,6 µm. Els ions OH són impureses fruit del procés de fabricació de la fibra òptica. Els sobretons de la freqüència de ressonància dels ions OH cauen dins de la banda espectral que s'utilitza per a comunicacions per fibra òptica. Les longituds d'ona de ressonància se situen a 720 nm, 950 nm i 1.380 nm. Per tot plegat es defineixen unes finestres o zones adients de treball, com podem veure en la corba que ens indica l'atenuació de la fibra òptica respecte de la longitud d'ona:

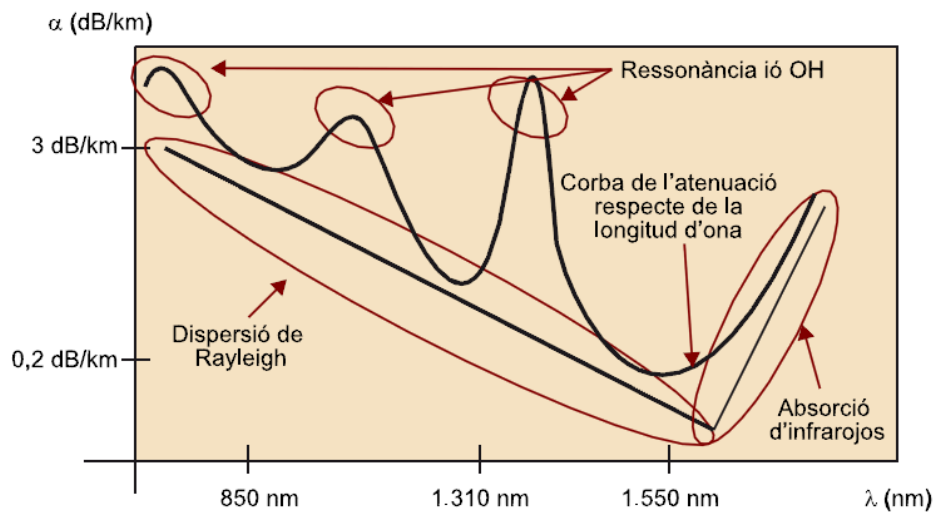


Figura 5: corba d'atenuació de la fibra òptica

Les simulacions realitzades en aquest projecte es realitzaran amb una longitud d'ona de treball de 1550 nm. (tercera finestra), que es el que s'utilitza normalment en els enllaços òptics de llarga distància. L'atenuació típica dels cables de fibra òptica de llarga distància (RTLTD) emprada habitualment és de 0,2 dB per Kilòmetre.

6.2.2 Causes extrínseques i balanç de potència

Com he exposat anteriorment, les causes extrínseques que provoquen pèrdues en els sistemes de comunicacions òptics són les provocades per efectes externs, com curvatures de la fibra òptica o variacions geomètriques de la guia. En dissenyar un enllaç de comunicacions òptiques, caldrà considerar les diferents fonts de pèrdues. En el procés de disseny això es tradueix en realitzar el balanç de potència de manera que:

$$P_{TX} > S_{RX} + L_{total} + M \quad (2)$$

On P_{TX} és la potència del transmissor, S_{RX} és la sensibilitat del receptor, L_{total} són les pèrdues totals i M és el marge de seguretat, que acostuma a ser de 3 dB.

Les pèrdues totals (L_{total}) venen determinades doncs per l'expressió:

$$L_{total} = L_{con} + L_{unió} + \alpha L \quad (3)$$

On L_{con} son les pèrdues dels connectors, $L_{unió}$ són les pèrdues provocades per les fusions de les fibres òptiques, α és l'atenuació per kilòmetre de la fibra òptica i L la distància del enllaç òptic.

6.3 Dispersió en els sistemes de comunicacions òptiques

La dispersió és el fenomen de separació de les ones de diferent freqüència en travessar un material. Tots els mitjans materials són més o menys dispersius, i la dispersió afecta totes les ones. La dispersió es deu al fet que la velocitat d'una ona depèn de la seva freqüència. Per exemple, les ones lluminoses de diferent longitud d'ona tenen velocitats de propagació diferents en el vidre, pel que són refractades en diferent mesura. Atès que les pèrdues en fibra són molt baixes i poden ser compensades mitjançant l'amplificació òptica, la causa principal de degradació del senyal deguda a la transmissió és la dispersió. La dispersió es pot dividir d'una manera simplificada en tres tipus: la intermodal o dispersió modal, la intramodal o dispersió cromàtica i la dispersió de mode de polarització (PMD).

6.3.1 Dispersió modal

La dispersió intermodal es deu a les diferents velocitats amb què es propaga cada mode. De fet, quan s'emet un pols de llum a l'entrada d'una fibra multimode, s'exciten diferents modes de propagació. Els polsos, associats a cada mode excitat, viatgen a través de la fibra amb velocitats diferents perquè tenen diferent velocitat de grup. Els temps de propagació en l'enllaç de fibra resulten diferents i a la sortida de la fibra hi ha un nombre de còpies del pols d'entrada igual al nombre de modes. Cada còpia té un retard diferent i, per tant, a la sortida de l'enllaç de fibra es genera un sol pols amb durada temporal més gran que la del pols a l'entrada. Quan es transmet un tren de polsos (en l'exemple de la figura 6 es mostren dos polsos) l'eixamplament pot produir que l'energia de dos polsos consecutius es superposi i ocorri l'anomenada interferència entre símbols (ISI), que depèn de la velocitat de transmissió (*bit rate*) i de la dispersió intermodal. Així, la dispersió és un fenomen clau que determina la velocitat de transmissió màxima a la qual es pot transmetre a través d'una fibra òptica.

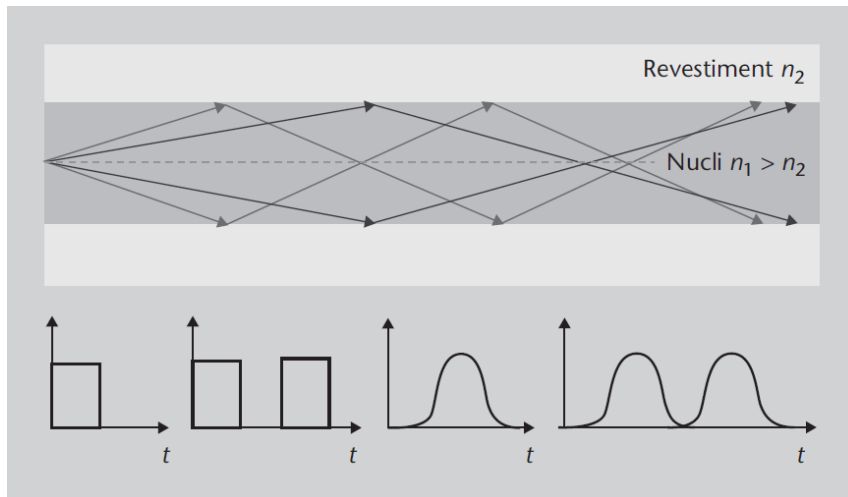


Figura 6: fenomen de dispersió multimode

Donat que aquest tipus de dispersió solament afecta a les fibres multimode, la dispersió intermodal no tindrà cap tipus d'impacte significatiu en les proves realitzades en aquest projecte.

6.3.2 Dispersió cromàtica

Contràriament a la dispersió modal, la dispersió cromàtica o intramodal ocorre tant en les fibra monomode com a les multimode, malgrat que en les fibres multimode es considera negligible en comparació a la dispersió modal. La dispersió cromàtica està originada per la dependència de l'índex de refracció de la guia amb la longitud d'ona de la ona que es transmet per aquesta guia, que ocasiona que diferents components freqüencials de la llum de la portadora òptica viatgin a velocitats diferents i, consegüentment, arribin en instants de temps diferents. Aquest fet implicarà que quan més amplada de banda de la portadora tinguem, més dispersió cromàtica patirem. La dispersió cromàtica bé determinada per l'expressió:

$$\sigma_t = D \cdot \Delta\lambda \cdot L \quad (4)$$

On D fa referència al paràmetre de dispersió, sent un paràmetre propi de la fibra (bé determinat per la forma de la guia i del material amb què s'hagi fabricat), L fa referència a la longitud de l'enllaç i $\Delta\lambda$ es l'amplada de banda del senyal injectat a la fibra expressat en longitud d'ona. El paràmetre de dispersió el proporciona el fabricant de la fibra de dues maneres: mitjançant una gràfica del paràmetre de dispersió respecte la longitud d'ona o mitjançant uns paràmetres que s'afegeixen a una funció que interpola la gràfica de D respecte a λ .

En les simulacions que realitzarem per comprovar el funcionament de les modulacions multinivell emprarem el valor:

$$D = 17 \frac{ps}{nm \cdot Km} \quad (5)$$

Aquest és el valor més comú en les actuals fibres monomode utilitzades en enllaços interurbans.

6.4 Dispersió de mode de polarització (PMD)

Un altre tipus de dispersió és la dispersió de mode de polarització (PMD). Podem definir aquest tipus de dispersió com el endarreriment diferencial de grup (DGD) entre els dos modes polaritzats, que provoca un eixamplament del pols en sistemes digitals. Si les fibres fossin idealment simètriques, els dos modes ortogonals es propagarien a la mateixa velocitat. Com que les fibres òptiques reals presenten certa birefringència (el desdoblament d'un feix de llum incident en dos feixos polaritzats linealment en direccions perpendiculars entre si com si la fibra tingués dos índexs de refracció diferents) cal contemplar aquest fenomen, ja que sol afectar a sistemes d'altres velocitats com les que són objecte d'aquest estudi.

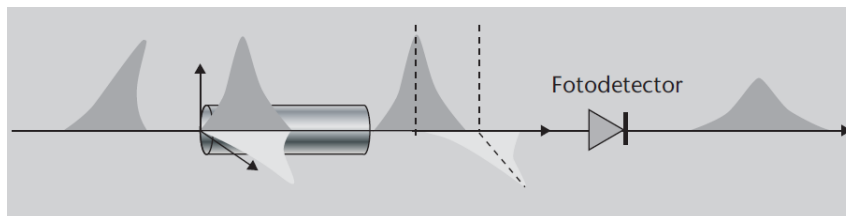


Figura 7: endarreriment del mode en el fenomen de PMD

Els factors que influeixen en la PMD de la fibra són els seus defectes geomètrics (e.g. la excentricitat del nucli), les tensions internes en el nucli de la fibra, les tensions externes en la protecció de la fibra i a les forces de torsió a que es veu sotmesa la fibra. Tot plegat provoca una diferència de propagació en els camps electromagnètics i el pols de sortida arriba distorsionat.

Es defineix el quocient de dispersió de mode de polarització com:

$$PMD = \frac{\Delta\tau}{\sqrt{L}} \quad (6)$$

On $\Delta\tau$ fa referència al retard del grup, L a la distància de l'enllaç i PMD a la dispersió de mode de polarització expressat en $\frac{ps}{km^2}$. Generalment, en enllaços de llarga distància es requereix un valor màxim de $0,06 \frac{ps}{km^2}$.

6.5 Efectes no lineals

Les degradacions del senyal induïdes per les no-linealitats creixen en augmentar la velocitat de transmissió i la potència, depenent també del tipus de senyal transmès, ja que l'índex de degradació no

lineal està determinat pel producte de la longitud de l'enllaç de fibra, el nombre de canals transmesos i la potència del canal.

Els efectes no lineals poden ser els produïts per l'efecte *Kerr* (es defineix com la dependència de l'índex de refracció d'un material amb el camp elèctric aplicat sobre aquest al quadrat), la dispersió de *Brillouin* (una part de la potència del senyal es converteix en un senyal que es propaga contra direcció amb freqüència desplaçada d'11 GHz respecte al senyal transmès) i l'efecte *Raman* (l'efecte Raman és considerable, en presència de molts canals d'alta potència en una amplada de banda àmplia).

La principal causa que pot provocar efectes no lineals no desitjats es la excessiva potència d'emissió òptica de làser. En aquest sentit definirem una potència d'emissió del làser sempre per sota dels 4 mw en les simulacions, per evitar efectes no lineals no desitjats.

Altrament, com que els sistemes OFDM transmetran símbols amb un important marge de potència instantània màxim i mínim (tindrà una elevada PAPR, *Peak to Average Power Ratio*), caldrà incorporar sempre en aquest tipus de sistemes amplificadors amb una alta linealitat.

7 Modulació òptica i sistemes de transmissió OFDM òptics

Enfront la modulació clàssica de portadora única, la modulació OFDM ens permet transmetre grups de N símbols on cada grup es transmet en una freqüència portadora diferent. A continuació descriuré els principis generals d'aquesta estratègia i com s'implementa en les funcions dissenyades per modelitzar aquest tipus de comunicacions.

7.1 Principis de funcionament

En el següent punt exposaré uns conceptes previs bàsics necessaris per comprendre els mecanismes de modulació òptica, explicant els estats de polarització més elementals i explicant l'estratègia més utilitzada per a la representació dels diferents estats de polarització com es l'esfera de Poincaré.

7.1.1 Paràmetres de Stokes i esfera de Poincaré

Els vectors de Stokes són una representació estàndard per a cada estat de polarització. És un conjunt de quatre nombres reals (S_0, S_1, S_2, S_3), anomenats paràmetres de Stokes, que compleixen:

$$S_0 = |E_{0x}|^2 + |E_{0y}|^2 = |E|^2 \quad (7)$$

$$S_1 = |E_{0x}|^2 - |E_{0y}|^2 \quad (8)$$

$$S_2 = 2 \operatorname{Re}(E_{0x} E_{0y}^*) = 2 |E_{0x}| |E_{0y}| \cos \phi \quad (9)$$

$$S_3 = 2 \operatorname{Im}(E_{0x} E_{0y}^*) = 2 |E_{0x}| |E_{0y}| \sin \phi \quad (10)$$

$$S_1^2 + S_2^2 + S_3^2 = S_0^2 \quad (11)$$

On la expressió 11 correspon a la definició d'una esfera de radi S_0 i coordenades S_1 , S_2 i S_3 , anomenada *esfera de Poincaré*:

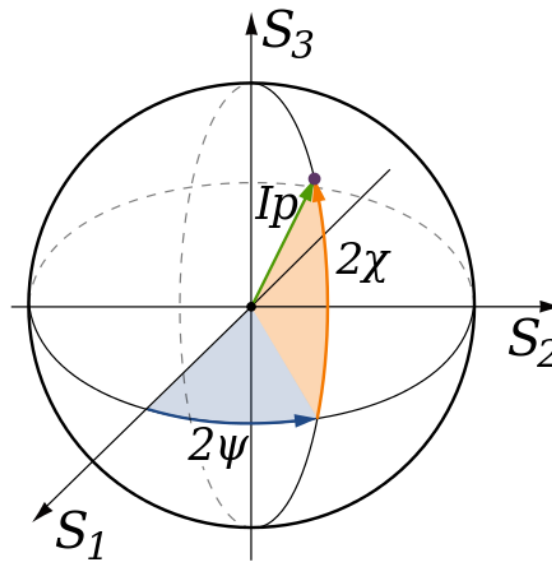


Figura 8: esfera de Poincaré

L'esfera de Poincaré és una eina de representació gràfica que ens permetrà representar un possible estat de polarització amb un punt en la seva superfície o amb l'angle 2χ de manera que:

- Valors de χ positius representen estats de polarització dextrogirs (cap a la dreta). En la esfera es representa per punts del hemisferi superior ($0^\circ < 2\chi \leq 90^\circ$).
- Valors de χ negatius representen estats de polarització levogirs (cap a l'esquerra). En la esfera es representa per punts del hemisferi inferior ($90^\circ \leq 2\chi < 360^\circ$).
- Punts del equador ($0^\circ \leq 2\psi < 360^\circ$ i $\chi=0$) representen estats de polarització lineal. Si els punts estan sobre l'eix O-S₁ representen estats de polarització horitzontal ($\psi=0$), mentre que els punts que estan a sobre de la part negativa del eix O-S₁, representen els estats de polarització vertical ($\psi=90^\circ$).
- El pol nord de l'esfera representa un estat de polarització circular dextrogir, mentre que el pol sud representa un estat de polarització circular levogir.

Per la representació geomètrica de la senyal de manera que es puguin representar els bits amb els que codificarem cada símbol en funció de la fase i l'amplitud de la senyal, s'utilitza un eix de coordenades de manera que l'eix horitzontal fa referència a la fase i l'eix vertical a l'amplitud:

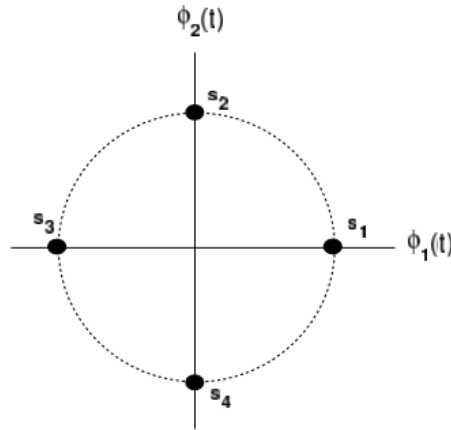


Figura 9: representació geomètrica de la senyal

On cada símbol de la constel·lació vindrà definit per l'expressió:

$$S_i(t) = \sum_{j=1}^N S_{i,j} \cdot \Psi_j(t) \quad i=1, \dots, M < t < T \quad (12)$$

On

$$S_{ij} = \int_0^T S_i(t) \cdot \Psi_j(t) dt \quad (13)$$

7.1.2 Tècniques de modulació avançada

La primera etapa que cal realitzar (tant en OFDM com en portadora única) és la codificació de símbol, on un nombre determinat de bits de la seqüència que es vol transmetre es transforma en uns símbols complexos, que ens permetrà modular en amplitud, quadratura o fase una senyal portadora.

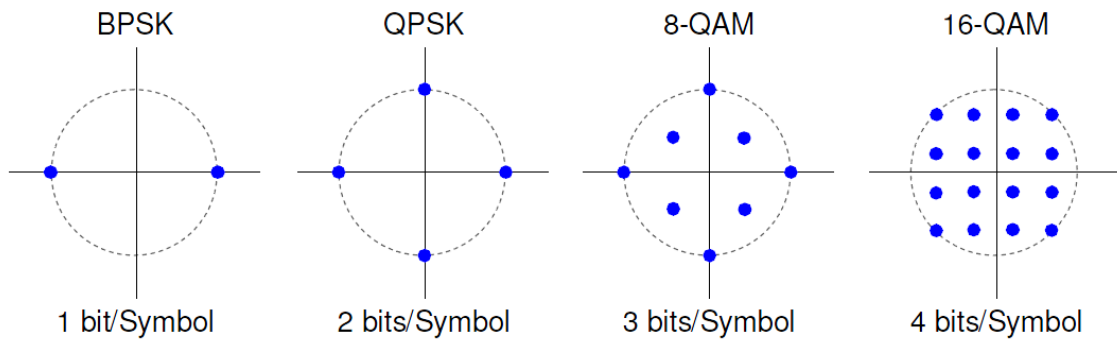


Figura 10: codificació bit a símbol emprades en diferents modulacions

Les tècniques avançades de modulació afegeixen la polarització a la amplitud i la fase, per obtenir més capacitat de transmissió amb la mateixa amplada de banda disponible, aprofitant els dos plans de polarització ortogonals. En sistemes de comunicació òptics, les dues senyals ortogonals són generades pel mateix làser i són modulades d'una manera independent.

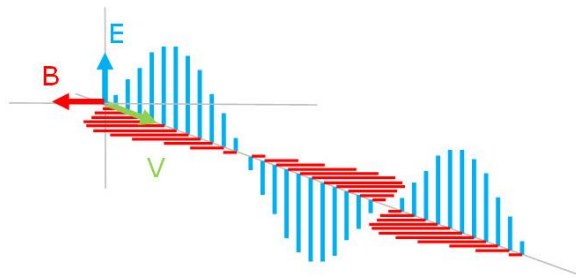


Figura 11: ones ortogonalment polaritzades

La utilització de qualsevol estratègia de modulació (com els de la figura 10) utilitzant la doble polarització de la senyal dona lloc a les constel·lacions 4D (en quatre dimensions), que són les que centraran el nostre anàlisi en aquest projecte.

7.2 Conceptes bàsics del OFDM

La multiplexació per divisió ortogonal de freqüència és un tipus de modulació que s'ha dissenyat per a ser utilitzada en condicions molt severes de canal de comunicacions, de manera que si utilitzéssim una modulació convencional, de tipus QAM, un canal selectiu en freqüència provocaria una distorsió tan acusada en el senyal modulad, que no en permetria la desmodulació sense una equalització realment complicada. Amb la modulació OFDM, s'intenta enviar una informació determinada, obtinguda mitjançant una seqüència de bits, a una velocitat de transmissió donada per un canal selectiu en freqüència, però sense necessitat d'utilitzar un equalitzador complicat al receptor.

Per a assolir aquest objectiu, una modulació OFDM es forma com la suma de N modulacions lineals i utilitza N freqüències portadores diferents ortogonals entre elles. N és un nombre enter relativament alt i potència de 2. Així, la seqüència original de símbols a transmetre es reparteix entre les modulacions, disposades en tota l'amplada de banda assignada a aquest tipus de modulació. L'espectre, o amplada de banda total disponible per a aquest tipus de modulació, es subdivideix en diversos intervals contigus per a assignar un interval a cada una de les modulacions utilitzades.

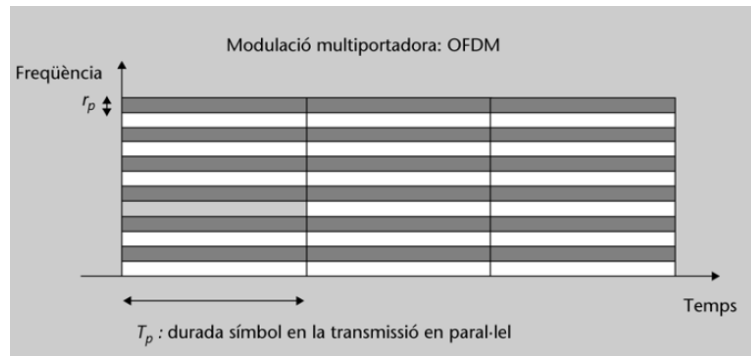


Figura 12: ocupació temps-freqüència de la modulació OFDM

Com veurem en els següents apartats, la implementació pràctica de la modulació OFDM es basa en l'algoritme IFFT per fer-la computacionalment eficient i la desmodulació OFDM es basa en l'algoritme FFT.

7.3 Processat de la senyal en sistemes OFDM

A continuació es mostra l'esquema general del processat de senyal OFDM que analitzarem en els següents apartats:

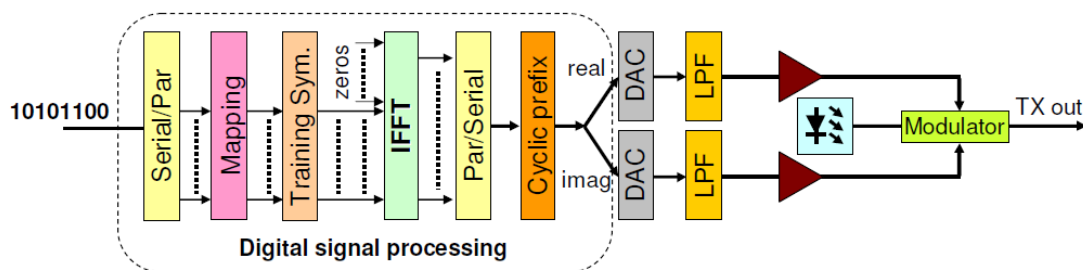


Figura 13: bloc corresponent al processat digital de la senyal

Veiem els dos primers blocs que corresponen al convertidor sèrie a paral·lel i el bloc de mapatge. Per modular el senyal binari que volem transmetre primerament caldrà convertir un flux de dades serialitzades en una matriu de dades mitjançant un convertidor de sèrie a paral·lel.

7.3.1 Símbols d'entrenament

Per poder recuperar de forma adequada la informació en recepció, caldran mecanismes per revertir els canvis produïts pel canal en la senyal transmesa, procés conegut amb el nombre d'equalització. Per poder realitzar aquest procés, caldrà tenir coneixement de la resposta del canal. El procés que permet obtenir el comportament del canal és lo que coneixem com estimació del canal.

L'efecte que produeix el canal sobre cada subportadora OFDM es tradueix en multiplicar el senyal d'entrada per una constant aleatòria complexa que dependrà de cada freqüència de portadora, més el soroll additiu AWGN. Caldrà determinar el valor d'aquesta constant per aconseguir culminar amb èxit el procés d'equalització.

Els mètodes no cecs són els més habituals per l'estimació de canal i es basen en la inserció de seqüències d'entrenament (símbols coneguts pel transmissor i pel receptor) que s'introduiran cada certa freqüència en la seqüència transmesa. Això permet al receptor, a partir dels símbols d'entrenament rebuts, tractar d'estimar quin ha estat el comportament del canal durant la transmissió.

En les modelitzacions realitzades en aquest treball utilitzo un patró d'entrenament de tipus bloc on típicament s'afegeix un símbol pilot cada 25 símbols OFDM transmesos, lo que implica un overhead del 4 % respecte a la informació transmesa.

7.3.2 Transformada Inversa de Fourier (IFFT)

L'algoritme IFFT l'utilitzem per enviar cada símbol QAM a transmetre a la freqüència que ha d'ocupar en l'equivalent passabaix.

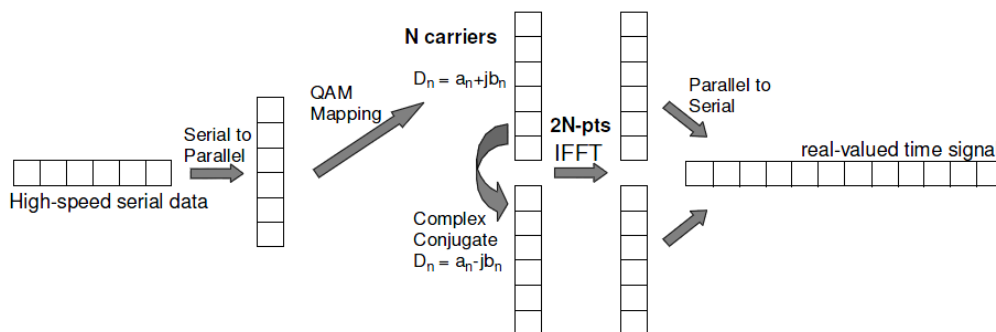


Figura 14: procés de realització de l'IFFT

IFFT significa 'operació inversa a l'FFT' i, per tant, amb l'IFFT es realitza una transformació del domini freqüencial al domini temporal. Es podria dir que el senyal de sortida del bloc IFFT és el senyal temporal que s'ha de transmetre (en la versió d'equivalent passabaix i discretitzada) perquè l'espectre del senyal esmentat coincideixi amb la trama de N símbols QAM que s'envia en cada símbol OFDM.

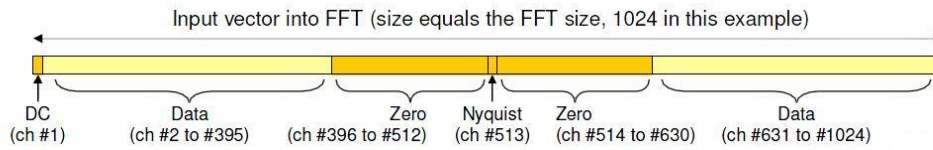


Figura 15: vector d'entrada de l'algorisme FFT

De fet, la modulació OFDM basada en l'algorisme IFFT, i la desmodulació OFDM basada en l'algorisme FFT, són una necessitat pràctica i d'implementació que s'utilitza en tots els transmissors i receptors d'OFDM.

7.3.3 Prefix cíclic

El canal de comunicacions produeix, entre altres alteracions, una dispersió temporal o *delay spread* la qual cosa provoca un nivell d'ISI important que degrada enormement la probabilitat d'error de símbol. Per tal de contrarestar aquest efecte i a l'hora preservar l'ortogonalitat entre les diferents portadores, cal introduir un temps de guarda entre símbol i símbol OFDM durant el qual es transmet una repetició parcial del símbol OFDM següent. Per aconseguir aquest efecte, l'últim tram de durada donada de la forma d'ona obtinguda en el modulador a la sortida del convertidor paral·lel en sèrie (P/S) s'estén cíclicament i es transmet a l'inici del símbol.

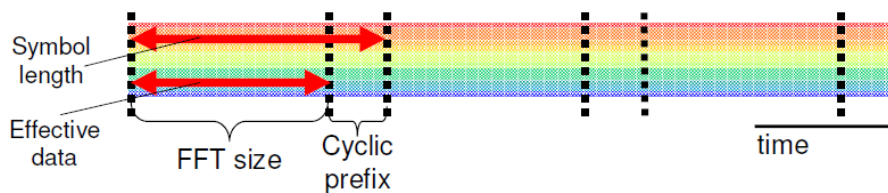


Figura 16: prefix cíclic

7.4 Modulació de la senyal

En la etapa de modulació es passa al domini òptic les components de fase i quadratura codificats en forma de nombres complexos en el domini elèctric. Per fer-ho, s'utilitzen moduladors Mach-Zehnder, basats en el principi d'interferometria. Com es pot veure en la figura següent, la llum entrant es divideix en dues branques. El modulador de fase present en cadascuna de les branques serveix per a imposar un desfasament relatiu entre branques. D'aquesta manera, es pot aconseguir a voluntat una interferència constructiva o destructiva al recombinar la llum de les dues branques a la sortida del modulador.

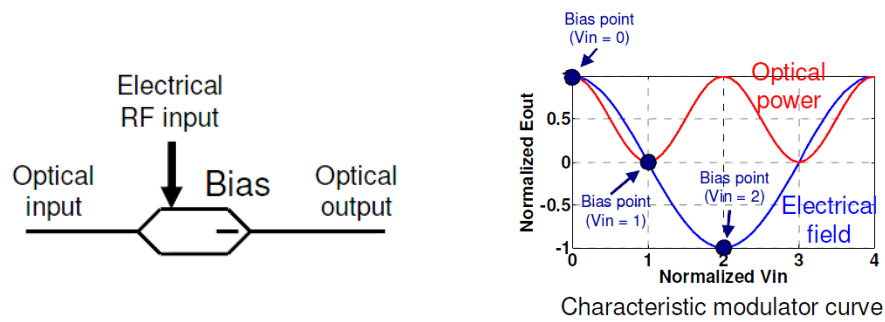


Figura 17: esquema del interferòmetre Mach-Zehnder estàndard

Es defineix el camp òptic de sortida mitjançant la següent expressió:

$$E_0 = E_i \cdot \cos\left(\frac{\pi}{2} \cdot \frac{(V_{RF} + V_{BIAS})}{V\pi}\right) \quad (14)$$

On E_0 fa referència al camp òptic de sortida, E_i al camp òptic d'entrada, V_{RF} al senyal elèctric d'entrada i V_π a la tensió necessària per produir un canvi de fase de la senyal en π radians, en una de les dues guies del modulador.

Les configuracions més comuns per modular la senyal OFDM són amb moduladors MZM estàndards o amb MZ niuats connectats mitjançant un super MZ, connectant les dues entrades elèctriques a la component en fase i quadratura de la senyal OFDM respectivament:

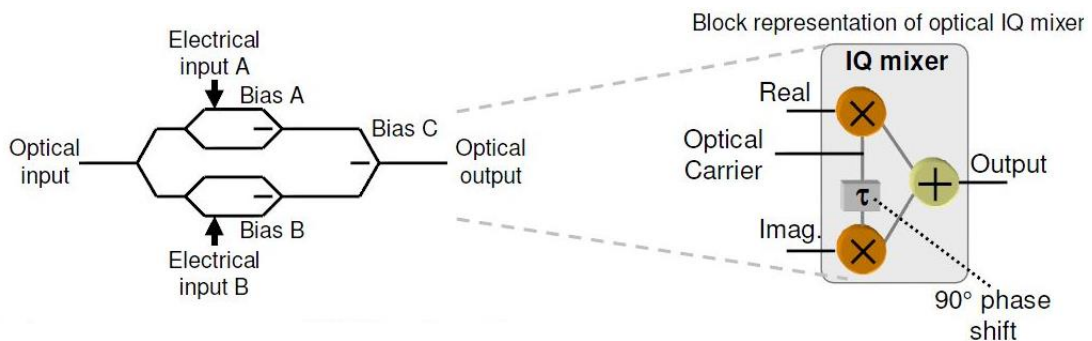


Figura 18: esquema del modulador IQ-MZM

Com veiem en la figura 18, la llum entrant es divideix en les dues branques, la de fase i la de quadratura (Q). En cada camí, el modulador Mach-Zehnder corresponent opera en mode push-pull i, per tant, realitza una modulació en amplitud o intensitat tal com s'ha descrit anteriorment. El modulador de fase present en una de les branques serveix per a imposar un desfasament relatiu entre branques. D'aquesta manera, es pot aconseguir qualsevol modulació complexa en el pla IQ després de recombinar la llum de les dues branques a la sortida del modulador. El camp elèctric de sortida vindrà determinat per la següent expressió:

$$E_0 = E_i \cdot \cos\left(\frac{\pi}{2} \cdot \frac{V_I + V_{BIAS}}{V_\pi}\right) \cdot e^{j\omega L D I^t} + E_i \cdot \cos\left(\frac{\pi}{2} \cdot \frac{V_Q + V_{BIAS}}{V_\pi}\right) \cdot e^{(j\omega L D I^t + \frac{\pi}{2})} \quad (15)$$

7.5 Detecció coherent

Els sistemes coherents es diferencien respecte als sistemes basats amb la detecció directa principalment en que el senyal rebut es mescla amb un làser local de referència.

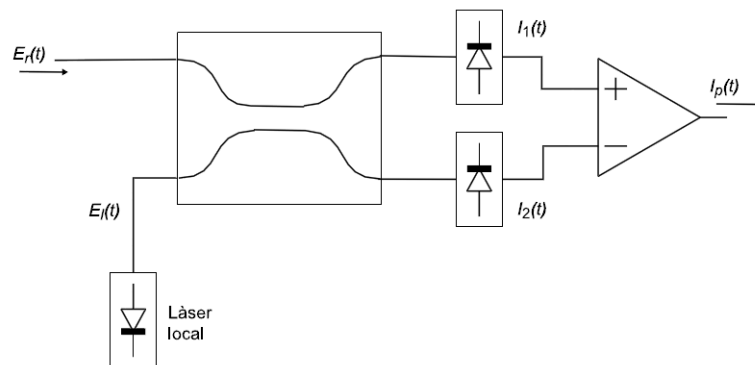


Figura 19: Receptor coherent utilitzant fotodetecció balancejada

Aquesta mescla idealment es realitza en un acoblador i la combinació de tots dos senyals es fotodetecta tot seguit, tal i com podem veure en la figura 19. Així, el corrent després de la fotodetecció conté tota la informació de potència, freqüència i fase combinades entre el senyal rebut i l'oscil·lador local, utilitzat com a referència.

Per la recepció de sistemes OFDM òptics amb modulació 4D multinivell com el que estem analitzant es recomanable utilitzar sistemes coherents amb diversitat de polarització ja que permet detectar de forma immediata i exacta l'estat de fase i polarització de cada símbol, contràriament als sistemes directes, que requereixen un processat més complex de la senyal.

7.6 Recepció: Equalització i estimació de canal

Una vegada convertida la senyal òptica al domini elèctric, s'ha de realitzar el procés invers que el realitzat en transmissió. Primerament caldrà sincronitzar el senyal rebut, extreure el prefix cíclic, realitzar la conversió sèrie a paral·lel i aplicar el FFT a la senyal rebuda.

El procés d'equalització es realitza en el domini de la freqüència, és a dir, una vegada que el senyal ha estat processat mitjançant l'algorisme FFT i és un pas previ a la detecció de símbols. Es podria dir que l'equalització entesa com la compensació de l'efecte del canal sobre la modulació OFDM és senzilla a causa de l'absència d'ISI i d'ICI. Consisteix en què cada sortida de les diferents portadores es divideix per la mostra de la funció de transferència del canal per la qual ha estat afectada. En aquest punt, es reprèn l'anàlisi del processament dels senyals considerant la presència de soroll. L'operació d'equalització és directament una divisió que es realitza per a tota la seqüència de símbols OFDM.

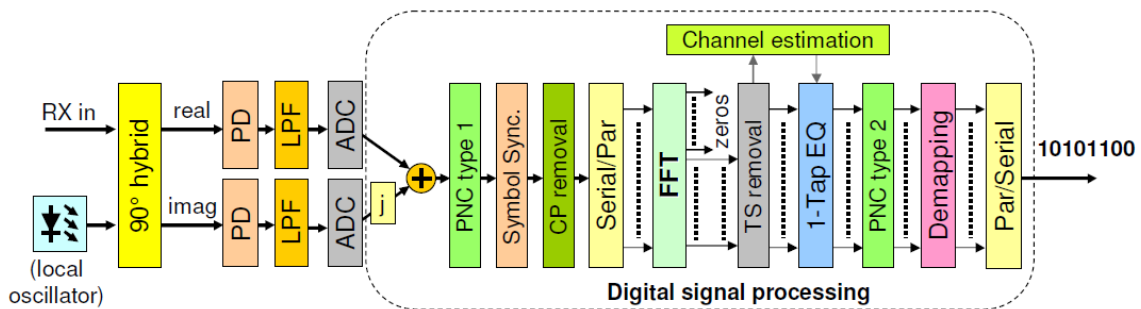


Figura 20: bloc del processament digital en recepció

Les simulacions poden tenir activat o no el bloc d'equalització, com veurem en els següents capítols.

8 Modulació 4D multinivell

La modulació QPSK distingeix entre 4 fases diferents de la portadora i aporta una capacitat de transmissió de 2 bits per símbol. Si utilitzem la doble polarització de l'ona portadora, podem duplicar aquesta capacitat de transmissió i obtenir quatre graus de llibertat: fase, amplitud, polarització X i polarització Y. D'aquesta manera es millora l'eficiència espectral i la velocitat de transmissió requerida, a l'hora que es redueix la vulnerabilitat de la senyal als problemes de distorsió i atenuació. La contrapartida serà una major complexitat en alguns dels elements que intervenen en la modulació, desmodulació i transmissió. A continuació centrarem el nostre anàlisi en tres modulacions 4D: DP-QPSK, 6P-QPSK i PS-QPSK.

8.1 Modulació DP-QPSK

Dins de les modulacions que podem denominar com 4D, la denominada DP-QPSK proposa modular cada polarització en fase i quadratura, de manera que podem obtenir fins a setze combinacions de fases i quadratures per totes dues polaritzacions on la fase prendrà els valors:

$$\varphi_x = \frac{m\pi}{4}$$

$$\varphi_y = \frac{m\pi}{4}$$
(16)

On m pot prendre uns valors de -3, -1, 1 i 3. Tot plegat ens permet disposar de quatre estats de polarització ($\pm 45^\circ$, LHC i RHC) amb quatre possibles valors de fase en cada estat:

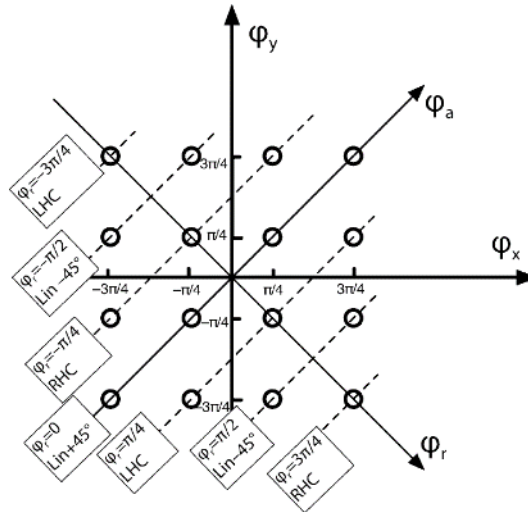


Figura 21: valors de fase emprats per la modulació DP-QPSK. Els eixos diagonals mostren les fases absoluta i relativa

Veiem com la fase $\varphi_r \in (-\pi, \pi]$ descriu l'el·lipticitat de l'estat de polarització, on la polarització lineal correspon als estats de $\varphi_r = 0, \pm\pi/2, \pi$, la polarització circular correspon als estats de $\varphi_r = \pm\pi/4, \pm 3\pi/4$ i els altres valors de φ_r correspon a la polarització el·líptica.

Finalment, podem expressar la senyal com un vector 4-dimensional amb els component reals (cal no confondre aquests vectors amb els vectors tridimensionals de Stokes):

$$s = \begin{pmatrix} E_{x,r} \\ E_{x,i} \\ E_{y,r} \\ E_{y,i} \end{pmatrix} = \begin{pmatrix} \|E\| \cos \varphi_x \sin \theta \\ \|E\| \sin \varphi_x \sin \theta \\ \|E\| \cos \varphi_x \cos \theta \\ \|E\| \sin \varphi_x \cos \theta \end{pmatrix}$$
(17)

On la potència òptica transmesa s'obté mitjançant la següent expressió:

$$P = \|s\|^2 = \|E\|^2 = E_{x,r}^2 + E_{x,i}^2 + E_{y,r}^2 + E_{y,i}^2$$
(18)

Podem concloure descrivint la modulació DP-QPSK com una modulació QPSK en polarització X i Y, on la part real i la part imaginària dels components de polarització prendran uns valors que seran 1 o menys 1.

8.2 Proposta de modulació multinivell: 6P-QPSK

Per aconseguir augmentar els nivells de modulació sense incrementar la potència mitjana d'emissió ni reduir la distància mínima de la constel·lació, es proposa un nou model de modulació anomenat 6P-QPSK.

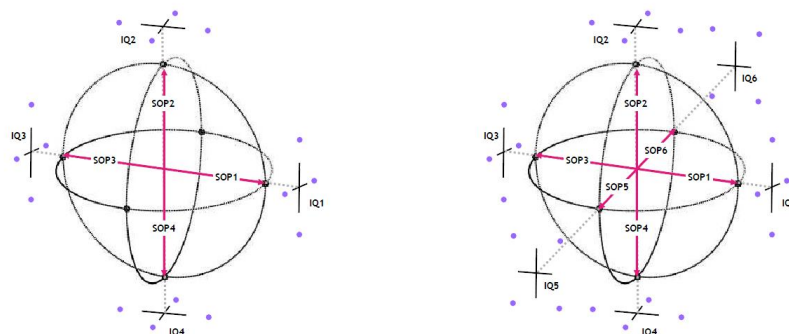


Figura 22: nou escenari de transformació proposat

Com podem veure a la figura 22, dels 16 nivells de codificació disponibles a la modulació DP-QPSK passem als 24 disponibles a la proposta de modulació 6P-QPSK.

8.2.1 Descripció

Com he exposat en el anterior punt, a la modulació DP-QPSK s'utilitzen quatre estats de polarització ($\pm 45^\circ$, LHC i RHC) i quatre nivells de fase per polarització. Una simple rotació a la polarització pot transformar aquest estats en x, y, LHC i RHC sense que es perdin prestacions. Tot plegat ens permet utilitzar dos estats de polarització addicionals amb quatre nivells de fase diferents per estat, sense que canviï la distància mínima entre punts i la potència mitjana necessària per transmetre la senyal.

Si partim dels 16 vectors que podem utilitzar en la modulació DP-QPSK com a conseqüència de la combinació de signes del conjunt $(\pm 1, \pm 1, \pm 1, \pm 1)$ i que anomenem C_1 , i combinem aquest escenari amb els nivells addicionals que poden estar format pel conjunt C_2 que està definit pels vuit vectors $(\pm 2, 0, 0, 0)$ amb qualsevol signe i qualsevol permutació de les coordenades, podem formar un conjunt de 24 nivells $C_3 = C_1 \cup C_2$. Tenim finalment una modulació formada per quatre estats de fase i sis estats de polarització (x, y, $\pm 45^\circ$, LHC, RHC).

Altrament, la teoria del nombre d'oscil·lació ens confirma que el màxim nombre d'esferes de radi 1 que poden tocar simultàniament a la esfera unitària en un espai euclidià K_4 (4-dimensional) és 24, per lo que podem afirmar que el nombre de símbols que poden ser codificats mantenint una distància mínima entre ells sense que calgui modular l'amplitud és 24.

Teòricament, utilitzant el conjunt C_3 la relació entre energia per bit i potència de soroll (E_b/N_0) de la modulació DP-QPSK millora en termes de SER:

$$\log_2(24)/\log_2(16) = 0.59dB \quad (19)$$

Si afegim el vector tot-zeros de manera que $C_4=C_3 \cup \{(0, 0, 0, 0)\}$, obtindrem un conjunt de 25 nivells amb una $d_{\min}=2$, $E_s= 4 \times 24/25$ i $\gamma=(25/96)\log_2 25=0.83$ dB, (on d_{\min} fa referència a la distància mínima de la constel·lació, E_s és la energia mitjana per símbol i γ a l'eficiència asimptòtica de potència) sent aquesta la forma més densa d'encabir 25 esferes en quatre dimensions. Aquesta opció es troba amb importants dificultats per implementar-la en la pràctica per lo que centrarem el nostre estudi en la constel·lació C_3 .

Caldrà resoldre, però, dos problemes importants per implementar la constel·lació C_3 : el mapatge de bit a símbol i la implementació o adaptació del modulador. A continuació analitzaré aquest dos aspectes.

8.2.2 Mapping 6P-QPSK

Els autors de l'article proposen l'ús de dos símbols seqüencials, lo que permetria codificar fins a $24^2=576$ nivells. Proposen, però, transmetre nou bits sobre dos símbols que correspondria a una eficiència de $\gamma=9/8=0.51$ dB. L'esquema del mapatge proposat pels autors és el següent:

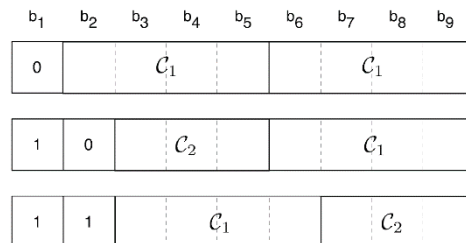


Figura 23: mapatge 6P-QPSK proposat

Veiem com els dos primers bits de cada paraula de nou bits definiran quin tipus de conjunt C_1 i C_2 s'utilitzarà per codificar els altres bits. Podem sintetitzar aquest mapatge amb el següent algorisme:

```

si  $b_1=0$  llavors
    map( $[b_2, b_3, b_4, b_5], C_1$ )
    map( $[b_6, b_7, b_8, b_9], C_1$ )
fsi
si  $b_1=1$  i  $b_2=0$  llavors
    map( $[b_3, b_4, b_5], C_2$ )
    map( $[b_6, b_7, b_8, b_9], C_1$ )
fsi
si  $b_1=1$  i  $b_2=1$  llavors
    map( $[b_3, b_4, b_5, b_6], C_1$ )
    map( $[b_7, b_8, b_9], C_2$ )
fsi
    
```

On la funció $\text{map}()$ fa referència a la codificació bit \rightarrow símbol corresponent.

Veiem com aquesta codificació anomenada 6P-QPSK té una capacitat de codificació de bit per símbol de 4.5 mentre que emprant C_3 tenim una capacitat de $\log_2 24 = 4,58$ bits per símbol.

8.2.3 Adaptació dels moduladors I/Q al 6P-QPSK

Cal adaptar la modulació del DP-QPSK pels vuit nivells afegits de la modulació 6P-QPSK. Com he exposat en el punt 8.1, l'estàndard DP-QPSK emprava moduladors I/Q per transmetre les dades que no poden utilitzar-se directament en la modulació 6P-QPSK, ja que l'amplitud de sortida amb un únic component de polarització present és (en termes normalitzats) $\sqrt{2}$ menor que quan tots dos són presents.

La implementació més directa de cara a modular el senyal òptic amb la constel·lació proposada és mitjançant l'ús de dos moduladors IQ en una configuració de diversitat de polarització, tal i com es pot veure en la figura 24(a). En l'article de l'Erik Agrell i d'en Magnus Karlsson també es proposa definir un element modulador de polarització (PolM) que actuarà com un retardador de polarització d'entrada de x ó y. En aquesta alternativa només s'utilitzen moduladors de fase, figura 24(b), forçant que tots els nivells tinguin la mateixa amplitud.

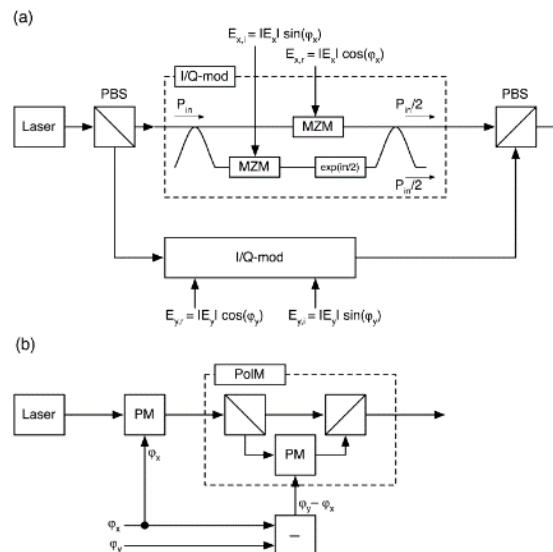


Figura 24: esquemes de transmissors proposats basats en moduladors IQ

8.2.4 BER i SER del nou escenari

Podem avaluar el SER de les modulacions C_1 (DP-QPSK), C_2 (PS-QPSK) i C_3 mitjançant les expressions:

$$C_1 \Rightarrow SER_1 = 1 - \left[1 - \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_s}{4N_0}} \right) \right]^4 \quad (20)$$

$$C_2 \Rightarrow SER_2 = 1 - \frac{1}{\sqrt{\pi}} \int_0^{\infty} (1 - \operatorname{erfc} \cdot x)^3 e^{-\left(x - \sqrt{\frac{E_s}{N_0}}\right)^2} dx \quad (21)$$

$$C_3 \Rightarrow SER_3 = 1 - \frac{1}{\sqrt{\pi}} \int_0^{\infty} (1 - \operatorname{erfc} \cdot x)^2 \cdot \left(x - \sqrt{\frac{E_s}{2N_0}}\right) e^{-\left(x - \sqrt{\frac{E_s}{2N_0}}\right)^2} dx \quad (22)$$

L'equació (11) on E_b/N_0 és la relació d'energia per bit / densitat espectral de potència de soroll es deriva fàcilment de la geometria d'una constel·lació de forma cúbica. L'equació (12) es proposada per varis autors [11, pag. 210],[12, pag 201] mitjançant el reconeixement de la constel·lació com una constel·lació de tipus 8-ària bi-ortogonal. Finalment, l'equació (13) és d'un desenvolupament més complex donada la geometria més irregular de la constel·lació i els autors de l'article no aporten cap dada de l'origen ni del seu desenvolupament.

A continuació es mostra el SER de les tres modulacions en funció del E_b/N_0 (energia per bit/densitat espectral de potència de soroll):

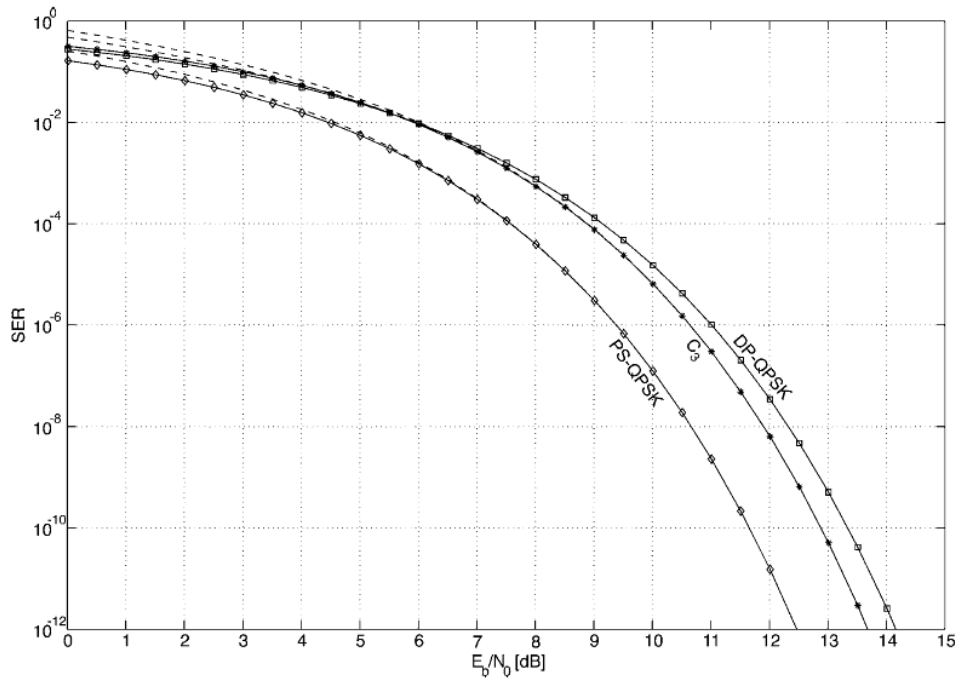


Figura 25: SER versus E_b/N_0 per les modulacions DP-QPSK, PS-QPSK i 6P-QPSK

El càlcul del BER per la modulació 6P-QPSK el podem calcular en base al SER mitjançant relació següent:

$$BER_{6P-QPSK} \approx \frac{5}{18} SER \quad (23)$$

A continuació podem veure els valors estimats de BER vs E_b/N_0 per les tres modulacions. Veiem com el valor de E_b/N_0 per un BER de 10^{-3} per la modulació PS-QPSK és 5,82 dB i per DP-QPSK i 6P-QPSK és de 6,78 dB:

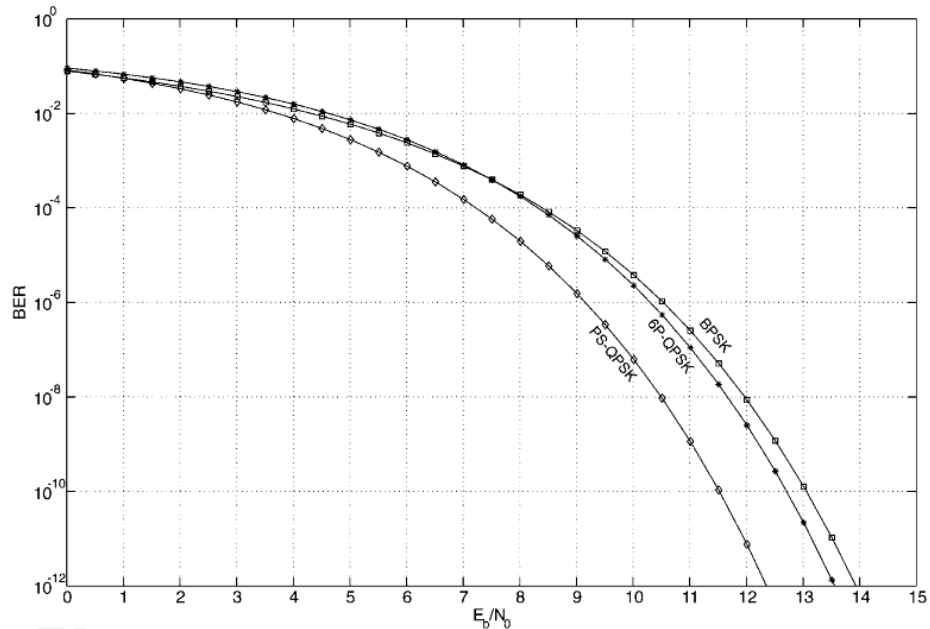


Figura 26: BER versus E_b/N_0 per les modulacions DP-QPSK, PS-QPSK i 6P-QPSK

En el present treball es mesurarà el BER comparant directament els bits transmesos i els bits rebuts *back-to-back* i comptabilitzant els bits erronis.

8.3 Modulació PS-QPSK

La modulació PS-QPSK proposa codificar símbols de tres bits en la constel·lació de vectors dels conjunt C2 ($\pm 2,0,0,0$). Donat que la distància entre els punt de la constel·lació augmenta en un factor $\sqrt{2}$ l'eficiència de potència asimptòtica millora en $\gamma=3/2=1,76$ dB respecte a la modulació DP-QPSK. Aquesta millora es complementa amb una disminució de complexitat al disminuir de 16 a 8 els nivells de modulació. En contrapartida, l'eficiència espectral es redueix de 4 a 3 bits per símbol respecte a la modulació DP-QPSK. Per codificar els vuit vectors del conjunt C2, els autors de l'article objecte d'estudi proposen un mapatge de tres bits en un espai complex determinat com exposaré en l'apartat d'implementació.

9 Implementació

Per realitzar la modelització de la modulació 6P-QPSK i PS-QPSK he creat una sèrie de scripts implementats en Matlab per modelitzar el seu funcionament. A continuació es mostra el processos que intervenen en la transmissió de les dades:

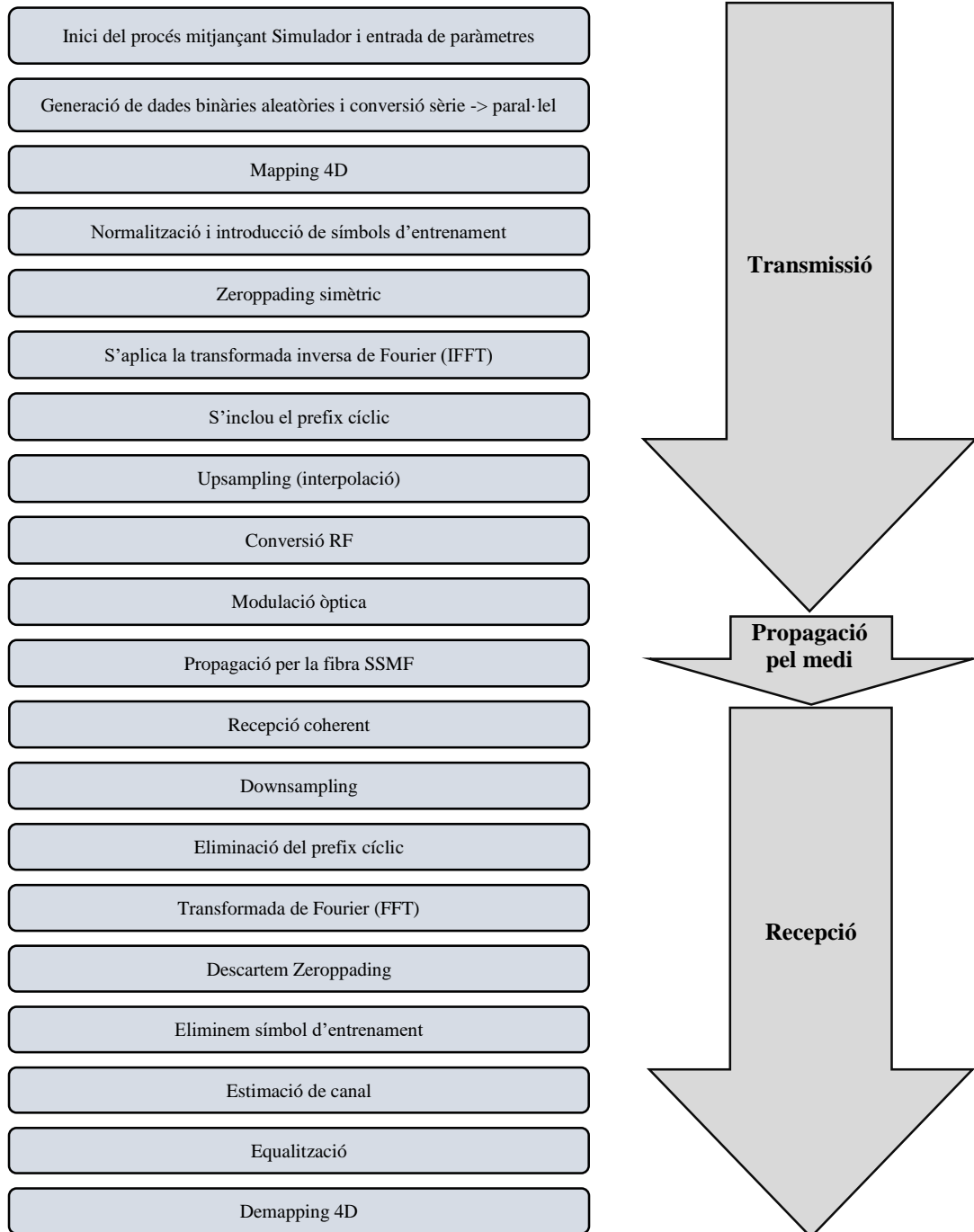


Figura 27: processos que intervenen en la transmissió ODFM 4D

A continuació descriuré les diferents classes implementades en Matlab que intervenen en la simulació de la transmissió d'un flux de dades. Donat que la filosofia del treball ha estat comparar les

prestacions d'aquestes modulacions amb la DP-QPSK, els scripts de prova han executat dos o tres tipus de modulacions (depenent de la prova) i s'han comparat la representació gràfica i els resultats de totes dues.

9.1 Blocs del sistema

Com he esmentat en l'anterior punt, la simulació dels diferents escenaris objecte d'estudi es realitzen mitjançant varies funcions que s'executaran en un entorn Matlab. A continuació es relacionen els scripts que en formen part:

- **Simulació:** S'inclouen una sèrie de scrips anomenats simulació.m que permeten definir les variables d'entrada de la simulació i inicien el procés de transmissió de dades.
- **Odfm4DSimulator / Odfm6PSimulator / OdfmPSSimulator:** És el script central en el que es va executant tot el procés de transmissió i recepció de dades. Es defineixen tres versions, la que executa la modulació 4D-QPSK, la que executa la modulació 6P-QPSK i la que executa la modulació PS-QPSK.
- **genOpField:** és el que s'encarrega de modular òpticament en quatre dimensions mitjançant IQ-MZM (modulació directa).
- **DCM:** és el script que modelitza la fibra compensadora de dispersió. El valor de compensació es fixa en $-100 \cdot 10^{-6} \text{ s/m}^2$.
- **demoludatorDigOFDM / demoludatorDigOFDM6P/ demoludatorDigOFDMPS:** aquest script s'encarrega del procés de desmodulació. Primerament efectua el downsampling, posteriorment representa l'espectre de la senyal rebuda, elimina el prefix cíclic, aplica la transformada de Fourier, descarta el zero padding, realitza l'estimació del canal i l'equalització i finalment efectua el demapatge de la constel·lació rebuda. Es defineixen tres versions, la que executa la desmodulació 4D-QPSK, la que executa la desmodulació 6P-QPSK i la que executa la desmodulació PS-QPSK.
- **EDFA:** modelitza un amplificador de fibra dopada amb Erbi (EDFA). Es defineix un guany fix de 20 dB.
- **FiberTx:** aquest script simula el comportament de una senyal òptica transmesa per una fibra òptica de tipus SSMF. És el script que s'encarrega de cridar al EDFA, sspropv2 i DCF.
- **hibridMixer:** modelitza un detector coherent amb fotodetectors balancejats.
- **IQ_MZM:** simula la modulació del camp elèctric E_{signal} mitjançant interferòmetre *IQ-Mach-Zehnder* emprant els paràmetres $V_I(\text{fase})$, V_Q (quadratura) del senyal d'entrada.

- **laser_cw**: modelitza el funcionament d'un làser CW.
- **modulator_MZM**: modelitza un modulador *Mach-Zehnder* ideal.
- **OSNR**: Afegeix soroll a la senyal.

A continuació podem veure les relacions que s'estableixen entre les diferents classes que intervenen en la simulació:

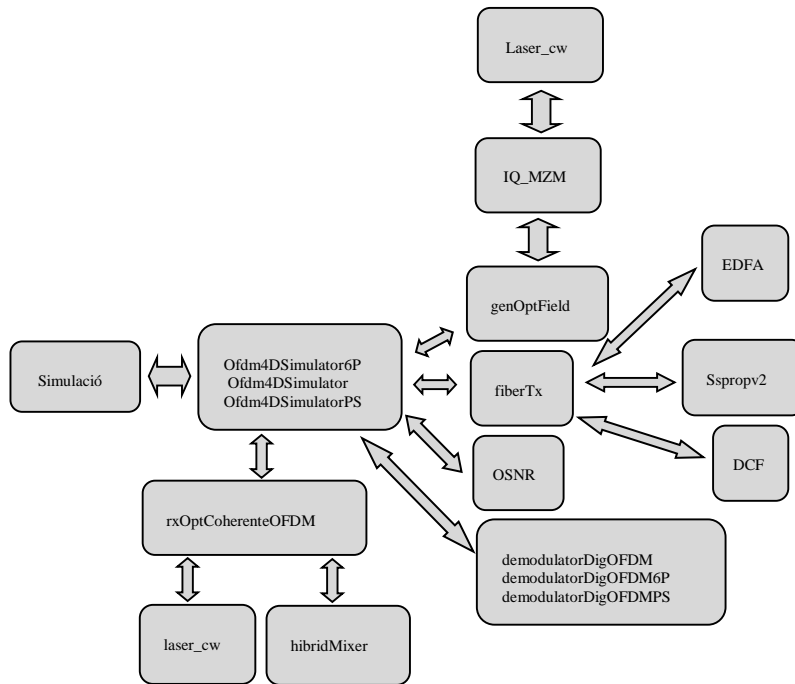


Figura 28: funcions i relacions del escenari de simulació

9.2 Paràmetres de la simulació OFDM

Com ja he anat esmentant en els anteriors punts, en els scripts i funcions implementats per modelitzar la transmissió OFDM òptica multinivell es defineixen una sèrie de paràmetres que ens permetran simular diferents escenaris. Els paràmetres relatius a la configuració OFDM són els següents:

- `OFDMParameters.simbolosOFDM`: És un valor sencer que defineix el nombre de símbols OFDM de cada bloc a transmetre.
- `OFDMParameters.subcarriers`: És un valor sencer que defineix el nombre de subportadores OFDM a emprar.
- `OFDMParameters.constelacion`: És una cadena de caràcters que fan referència al tipus de modulació emprada.
- `OFDMParameters.NFFT`: És un valor sencer que fa referència al valor de mostreig.
- `OFDMParameters.RSymb`: És el nombre de símbols a transmetre per segon.

- `OFDMParameters.RFfreqHz`: És la freqüència de transmissió.
- `OFDMParameters.PC`: És el percentatge de prefix cíclic.
- `OFDMParameters.SimParam.simulations`: Defineix el nombre de simulacions a realitzar.
- `OFDMParameters.SimParam.BEROSNR`: Defineix el valor de SNR mínim sobre el que es realitza l'estudi.
- `OFDMParameters.SimParam.BERFIBER`: Activa el processat d'estudi d'abast del Sistema.
- `OFDMParameters.SimParam.ecualization`: Activa el bloc d'equalització.
- `OFDMParameters.SimParam.fiber`: Activa la transmissió mitjançant un canal òptic.

9.3 Paràmetres de la fibra òptica i del làser

També es defineixen una sèrie de paràmetres relacionats amb la fibra òptica i el làser:

- `laserParam.power`: Potència en Vats d'emissió del làser.
- `laserParam.loPower`: Potència en Vats de l'oscil·lador local.
- `iberParam.fiberLength`: Longitud de cada tram de fibra.
- `fiberParam.tramos`: Nombre de trams de fibra òptica.
- `fiberParam.EDFA`: Dada booleana que indica si existeixen amplificadors EDFA.
- `fiberParam.waveLength`: Longitud d'ona d'emissió en metres.
- `fiberParam.attenuation`: Atenuació de la fibra en dB per kilòmetre.
- `fiberParam.dispersionChrom`: Dispersió cromàtica en segons partit per metre quadrat.
- `fiberParam.FN_dB`: Figura de soroll dels amplificadors EDFA en dB.
- `fiberParam.birr`: Factor de birefringència de la fibra.
- `fiberParam.PMD`: Factor de dispersió del mode de polarització, en segons partit per arrel quadrada de metre.
- `fiberParam.lc`: Distància de correlació, relacionat amb el PMD.

9.4 Codificació

El script *ofdm4DSimulator6P* és l'encarregat de simular la modulació 6P-QPSK. Les principals diferències respecte al script *ofdm4DSimulator* en la fase de generació de la senyal binària i del conversor sèrie a paral·lel són les següents:

- Cal adaptar el nombre de bits aleatoris generats per a cada simulació, perquè ara tindrem 4.5 bits per símbol en el cas de la modulació 6P-QPSK i 3 en el cas de la modulació PS-QPSK:

```
t_data=rand(1,ofdmParam.subcarriers*4.5*(ofdmParam.simbolsOFDM-1))>0.5; %6P-QPSK  
t_data=rand(1,ofdmParam.subcarriers*3*(ofdmParam.simbolsOFDM-1))>0.5; %PS-QPSK
```

- En la generació del senyal OFDM i el procés de conversió sèrie a paral·lel també serà necessari adaptar el nou rati bits – símbol i la nova paraula de 9 bits en el cas de 6P-QPSK:

```
data = t_data_Polar(dataCounter:dataCounter+ofdmParam.subcarriers*4.5-1);  
dataCounter=dataCounter+(ofdmParam.subcarriers*4.5);
```

```
s_p_data = reshape(data,9,length(data)/9);
```

I de 3 bits en el cas de PS-QPSK:

```
data = t_data_Polar(dataCounter:dataCounter+ofdmParam.subcarriers*3-1);  
dataCounter=dataCounter+(ofdmParam.subcarriers*3);
```

```
s_p_data = reshape(data,3,length(data)/3);
```

La conversió de dades sèrie a paral·lel es realitza en un bucle for-end. Dins del cos d'aquest bucle, s'implementarà el mecanisme de mapatge que veurem en el següent punt.

9.4.1 Mapatge 6P-QPSK i implementació

A diferència del procés de mapatge de DP-QPSK, que es podria fer directament de bit a coordenada complexa, el procés de mapatge de 6P-QPSK és d'una certa complexitat. En l'article *Power-Efficient Modulation Formats in Coherent Transmission Systems* se'ns proposa una codificació de bit a

símbol determinada, que és el que s'implementa en aquest treball. Primerament, els bits codificats al conjunt C_1 s'invertiran de manera que $0 \rightarrow 1$ i $1 \rightarrow -1$. En quant als vectors del conjunt C_2 , els autors proposen la següent codificació de vectors a bits:

(0,0,0,2) -> 000
(0,0,2,0) -> 001
(0,2,0,0) -> 010
(2,0,0,0) -> 011
(0,0,0,-2) -> 111
(0,0,-2,0) -> 110
(0,-2,0,0) -> 101
(-2,0,0,0) -> 100

La implementació d'aquest mapatge i que es mostra a continuació simplement afegeix el nombre complex que correspon a la paraula d'entrada depenent dels dos primers bits de més pes del "supersímbol":

```
=====
% Mapping 4D (6P-QPSK)
=====

dataPolX=[];
dataPolY=[];
for jj=1: (length(data)/9)
    if(s_p_data(9,jj)==-1)
        Xj1 = complex(-s_p_data(2,jj) , -s_p_data(1,jj));
        Xj2 = complex(-s_p_data(4,jj) , -s_p_data(3,jj));
        Xj3 = complex(-s_p_data(6,jj) , -s_p_data(5,jj));
        Xj4 = complex(-s_p_data(8,jj) , -s_p_data(7,jj));
    end;
    if(s_p_data(9,jj)==1 && s_p_data(8,jj)==-1)
        Xj1 = complex(-s_p_data(2,jj) , -s_p_data(1,jj));
        Xj2 = complex(-s_p_data(4,jj) , -s_p_data(3,jj));

    if(s_p_data(7,jj)==-1) && (s_p_data(6,jj)==-1) && (s_p_data(5,jj)==-1)
        Xj4 = 0+0i;
        Xj3 = 0+2i;
    end;
    if(s_p_data(7,jj)==-1) && (s_p_data(6,jj)==-1) && (s_p_data(5,jj)==1)
        Xj4 = 0+0i;
        Xj3 = 2+0i;
    end;
    if(s_p_data(7,jj)==-1) && (s_p_data(6,jj)==1) && (s_p_data(5,jj)==-1)
        Xj4 = 0+2i;
        Xj3 = 0+0i;
    end;
    if(s_p_data(7,jj)==1) && (s_p_data(6,jj)==-1) && (s_p_data(5,jj)==-1)
        Xj4 = -2+0i;
        Xj3 = 0+0i;
    end;
    if(s_p_data(7,jj)==1) && (s_p_data(6,jj)==1) && (s_p_data(5,jj)==1)
        Xj4 = 0+0i;
        Xj3 = 0-2i;
    end;
    if(s_p_data(7,jj)==1) && (s_p_data(6,jj)==1) && (s_p_data(5,jj)==-1)
        Xj4 = 0+0i;
        Xj3 = -2+0i;
    end;
    if(s_p_data(7,jj)==1) && (s_p_data(6,jj)==-1) && (s_p_data(5,jj)==1)
        Xj4 = 0-2i;
        Xj3 = 0+0i;
    end;
end;
```

```
if(s_p_data(7,jj)==-1)&& (s_p_data(6,jj)==1)&&(s_p_data(5,jj)==1)
Xj4 = 2+0i;
Xj3 = 0+0i;
end;

end;
if(s_p_data(9,jj)==1 && s_p_data(8,jj)==1)
if(s_p_data(3,jj)==-1)&& (s_p_data(2,jj)==-1)&&(s_p_data(1,jj)==-1)
Xj2 = 0+0i;
Xj1 = 0+2i;
end;
if(s_p_data(3,jj)==-1)&& (s_p_data(2,jj)==-1)&&(s_p_data(1,jj)==1)
Xj2 = 0+0i;
Xj1 = +2+0i;
end;
if(s_p_data(3,jj)==-1)&& (s_p_data(2,jj)==1)&&(s_p_data(1,jj)==-1)
Xj2 = 0+2i;
Xj1 = 0+0i;
end;
if(s_p_data(3,jj)==1)&& (s_p_data(2,jj)==-1)&&(s_p_data(1,jj)==-1)
Xj2 =-2+0i;
Xj1 =0+0i;
end;
if(s_p_data(3,jj)==1)&& (s_p_data(2,jj)==1)&&(s_p_data(1,jj)==1)
Xj2 = 0+0i;
Xj1 = 0-2i;
end;
if(s_p_data(3,jj)==1)&& (s_p_data(2,jj)==1)&&(s_p_data(1,jj)==-1)
Xj2 = 0+0i;
Xj1 = -2+0i;
end;
if(s_p_data(3,jj)==1)&& (s_p_data(2,jj)==-1)&&(s_p_data(1,jj)==1)
Xj2 = 0-2i;
Xj1 = 0+0i;
end;
if(s_p_data(3,jj)==-1)&& (s_p_data(2,jj)==1)&&(s_p_data(1,jj)==1)
Xj2 = 2+0i;
Xj1 = 0+0i;
end;
Xj3 = complex(-s_p_data(5,jj) , -s_p_data(4,jj));
Xj4 = complex(-s_p_data(7,jj) , -s_p_data(6,jj));
end
dataPolY = [dataPolY Xj3];
dataPolX = [dataPolX Xj4];
dataPolY = [dataPolY Xj1];
dataPolX = [dataPolX Xj2];

end
```

Una vegada creada la matriu de símbols, es normalitza multiplicant el contingut de la matriu pel factor de normalització:

```
NormFactor = 1 /sqrt( mean ( mean ( abs (MPolY) .^ 2 ) ) ) ;
MPolX = MPolX * NormFactor;
MPolY = MPolY * NormFactor;
```

A continuació s'introdueixen els símbols d'entrenament i posteriorment l'oversampling afegint zeros, com a procés previ a la realització de la transformada IFFT en dues dimensions. La funció de IFFT ve definida per la funció *iff2OFDM* i s'aplica a cada símbol, considerant cada columna de la matriu de polarització X i Y. Posteriorment s'afegeix el prefix cíclic al percentatge definit en la variable *OFDMPParameters.PC*, que normalment serà del 6,25 %. Finalment es procedirà a modular la senyal mitjançant la funció *IQ_MZM*. Donat que el modulador de la simulació no té limitació d'amplitud de

sortida (al contrari de lo que succeiria en un cas real), no caldrà adaptar la modulació de la funció IQ_MZM als nous vectors afegits en la proposta 6P-QPSK.

9.4.2 Mapatge PS-QPSK i implementació

En aquest cas el mapatge es realitza codificant tres bits als vectors del conjunt C2, d'acord a lo proposat en l'anterior punt:

(0,0,0,2) -> 000
(0,0,2,0) -> 001
(0,2,0,0) -> 010
(2,0,0,0) -> 011
(0,0,0,-2) -> 111
(0,0,-2,0) -> 110
(0,-2,0,0) -> 101
(-2,0,0,0) -> 100

A diferència de la implementació de la modulació 6P-QPSK, aquesta codificació de bit a espai complex es realitza directament, per lo que la seva implementació és més senzilla respecte al mapatge proposat per la modulació 6P-QPSK:

```
=====
% Mapping 4D (PS-QPSK)
=====
dataPolX=[];
dataPolY=[];
for (jj=1: length(data)/3)

    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==-1)
        Xj1 = 0+0i;
        Xj2 = 0+2i;

    end;
    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==1)
        Xj1 = 0+0i;
        Xj2 = 2+0i;
    end;
    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==1) && (s_p_data(1,jj)==-1)
        Xj1 = 0+2i;
        Xj2 = 0+0i;
    end;
    if(s_p_data(3,jj)==1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==-1)
        Xj1 = -2+0i;
        Xj2 = 0+0i;
    end;
    if(s_p_data(3,jj)==1) && (s_p_data(2,jj)==1) && (s_p_data(1,jj)==1)
        Xj1 = 0+0i;
        Xj2 = 0-2i;
    end;
    if(s_p_data(3,jj)==1) && (s_p_data(2,jj)==1) && (s_p_data(1,jj)==-1)
        Xj1 = 0+0i;
        Xj2 = -2+0i;
    end;
    if(s_p_data(3,jj)==1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==1)
        Xj1 = 0-2i;
        Xj2 = 0+0i;
    end;
    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==1) && (s_p_data(1,jj)==1)
        Xj1 = 2+0i;
        Xj2 = 0+0i;
    end;
dataPolY = [dataPolY Xj2];
dataPolX = [dataPolX Xj1];
```

end

Com en el cas de la implementació de la modulació 6P-QPSK, no caldrà adaptar la funció IQ_MZM a la nova modulació PS-QPSK.

9.5 Transmissió per la fibra i etapa de recepció

Per a emular la fibra s'utilitza la funció `sspropv2`, que forma part dels mòduls lliures `SSPROP` proporcionat per la Universitat de Maryland. Aquesta funció és instanciada des de la funció `fiberTX()` que emula el comportament físic de la fibra. El que fa la funció `sspropv2()` bàsicament és solucionar de forma numèrica les equacions de *Manakov* que descriuen el comportament de la fibra monomode. Aquesta funció tindrà en compte els efectes mutus de les dues senyals de polarització, fet que ens permetrà avaluar l'efecte de la distorsió PMD com veurem en l'apartat de les proves i simulacions. L'etapa de recepció constarà d'un receptor coherent homodí modelitzat en la funció `rxOptCoherenteOFDM()` i de dos blocs *mixer* híbrids de desfasament de 90° en ambdues polaritzacions modelitzat per la funció `hibridMixer()`.

9.5.1 Algorisme de demapping 6P-QPSK i implementació

La funció `demodulatorDigOFDM6P` és qui s'encarrega de desmodular el senyal OFDM i el *demapping* de la senyal rebuda. En el cas de 6P-QPSK no podem realitzar un demapatge directe com es realitzava en la modulació DP-QPSK, sinó tindrem que calcular la distància euclidiana entre un punt rebut i els diferents vectors del conjunt C_3 . He definit una sèrie de funcions auxiliars per a realitzar aquest procés i clarificar el codi. La funció `slicingB` retorna el punt de la constel·lació més proper al punt rebut. Per fer-ho, calcula la distància euclidiana entre tots els punt definit en el vector `mapDist` i el punt rebut $(\overline{X}, \overline{Y})$. Aquesta distància bé determinada per la següent expressió:

$$d(C_3, X, Y) = \sqrt{(C_{Xr} - X_r)^2 + (C_{Xi} - X_i)^2 + (C_{Yr} - Y_r)^2 + (C_{Yi} - Y_i)^2} \quad (24)$$

On C_3 fa referència al punt del conjunt C_3 , $Y_{r/i}$ fa referència al punt de la coordenada complexa en el pla Y i, $X_{r/i}$ fa referència al punt de la coordenada complexa en el pla X , sent r/i el seu valor real o complex.

La implementació d'aquest càlcul es realitza de la següent manera:

```
function res = slicingB(Xcomplex,Ycomplex)
    % Retorna el punt de la constel·lació mes proper del punt rebut
    mapDist=[-2,0,0,0,9;2,0,0,0,9;0,-2,0,0,9;0,2,0,0,9;0,0,-2,0,9;0,0,2,0,9;
            0,0,0,-2,9;0,0,0,2,9;1,1,1,1,9;1,1,1,-1,9;1,1,-1,1,9;1,1,-1,-1,9;
            1,-1,1,1,9;1,-1,1,-1,9;1,-1,-1,1,9;1,-1,-1,-1,9;-1,1,1,1,9;
            -1,1,1,-1,9;-1,1,-1,1,9;-1,1,-1,-1,9;-1,-1,1,1,9;-1,-1,1,-1,9;
            -1,-1,-1,1,9;-1,-1,-1,-1,9];

    for jj=1:length(mapDist)
        mapDist(jj,5)=sqrt(((mapDist(jj,1)-real(Xcomplex))^2)+((mapDist(jj,2)-
            imag(Xcomplex))^2)+((mapDist(jj,3)-real(Ycomplex))^2)+((mapDist(jj,4)-
            imag(Ycomplex))^2));
    end;
```

```
[M, I]=min(mapDist(:,5));  
res=mapDist(I,[1,2,3,4]);  
end
```

També es defineix la funció *esC2* que retorna *true* si el vector d'entrada correspon algun dels vectors del conjunt C_2 i *false* en cas contrari:

```
function res=esC2(vector)  
    % Retorna true si el punt rebut correspon a un dels vectors C2  
    A=[-2,0,0,0];B=[0,-2,0,0];C=[0,0,-2,0];D=[0,0,0,-2];  
    E=[2,0,0,0]; F=[0,2,0,0]; G=[0,0,2,0]; H=[0,0,0,2];  
    if isequal(vector,A)||isequal(vector,B)||isequal(vector,C)||isequal(vector,D)  
        ||isequal(vector,E)||isequal(vector,F)||isequal(vector,G)||isequal(vector,H)  
        res=true;  
    else  
        res=false;  
    end;  
end
```

La funció *deMapC2* realitza la codificació símbol (vector) a binari segons la proposta exposada en el punt 9.4.1

```
function res=deMapC2(vector)  
    % Demapeja el vector C2 rebut  
    A=[-2,0,0,0];B=[0,-2,0,0];C=[0,0,-2,0];D=[0,0,0,-2];  
    E=[2,0,0,0];F=[0,2,0,0];G=[0,0,2,0];H=[0,0,0,2];  
    I=[0,0,0];J=[0,0,1];K=[0,1,0];L=[1,0,0];  
    M=[1,1,1];N=[1,1,0];O=[1,0,1];P=[0,1,1];  
  
    if(isequal(vector,A))res=L;  
    end;  
    if(isequal(vector,B))res=O;  
    end;  
    if(isequal(vector,C))res=N;  
    end;  
    if(isequal(vector,D))res=M;  
    end;  
    if(isequal(vector,E))res=P;  
    end;  
    if(isequal(vector,F))res=K;  
    end;  
    if(isequal(vector,G))res=J;  
    end;  
    if(isequal(vector,H))res=I;  
    end;  
end
```

La funció *inverC1* inverteix els bits del vector d'entrada segons la proposta esmentada en l'anterior punt:

```
function res=invertC1(vector)  
    % Inverteix els bits del vector  
    aux=[];  
    if((vector(1))<=0) aux(1)=1;  
    end;  
    if((vector(1))>0) aux(1)=0;  
    end;  
    if((vector(2))<=0) aux(2)=1;  
    end;  
    if((vector(2))>0) aux(2)=0;  
    end;  
    if((vector(3))<=0) aux(3)=1;  
    end;  
    if((vector(3))>0) aux(3)=0;  
    end;  
    if((vector(4))<=0) aux(4)=1;  
    end;  
end
```

```
        if((vector(4))>0) aux(4)=0;  
        end;  
    res=aux;  
end
```

Finalment, el procés de *demapping* s'implementa de la següent manera:

```
%=====
% DEMAPPING
%=====
% Demapatge 6PQPSK
Frames = ofdmParam.symbolsOFDM+floor(ofdmParam.symbolsOFDM/25);
rx_bindata=[];
for si=1:Frames-1-floor(ofdmParam.symbolsOFDM/25)
    for ii=1:2:(ofdmParam.subcarriers)
        if(esC2(slicingB(ODataPolX_Est(si,ii),ODataPolY_Est(si,ii))))
            auxC1=invertC1(slicingB(ODataPolX_Est(si,ii+1),ODataPolY_Est(si,ii+1)));
            rx_bindata=[rx_bindata auxC1(4)];
            rx_bindata=[rx_bindata auxC1(3)];
            rx_bindata=[rx_bindata auxC1(2)];
            rx_bindata=[rx_bindata auxC1(1)];
            auxC2=deMapC2(slicingB(ODataPolX_Est(si,ii),ODataPolY_Est(si,ii)));
            rx_bindata=[rx_bindata auxC2(3)];
            rx_bindata=[rx_bindata auxC2(2)];
            rx_bindata=[rx_bindata auxC2(1)];
            rx_bindata = [rx_bindata 0];
            rx_bindata = [rx_bindata 1];
        elseif(esC2(slicingB(ODataPolX_Est(si,ii+1),ODataPolY_Est(si,ii+1))))
            auxC2=deMapC2(slicingB(ODataPolX_Est(si,ii+1),ODataPolY_Est(si,ii+1)));
            rx_bindata=[rx_bindata auxC2(3)];
            rx_bindata=[rx_bindata auxC2(2)];
            rx_bindata=[rx_bindata auxC2(1)];
            auxC1=invertC1(slicingB(ODataPolX_Est(si,ii),ODataPolY_Est(si,ii)));
            rx_bindata=[rx_bindata auxC1(4)];
            rx_bindata=[rx_bindata auxC1(3)];
            rx_bindata=[rx_bindata auxC1(2)];
            rx_bindata=[rx_bindata auxC1(1)];
            rx_bindata = [rx_bindata 1];
            rx_bindata = [rx_bindata 1];
        else
            auxC1=invertC1(slicingB(ODataPolX_Est(si,ii+1),ODataPolY_Est(si,ii+1)));
            rx_bindata=[rx_bindata auxC1(4)];
            rx_bindata=[rx_bindata auxC1(3)];
            rx_bindata=[rx_bindata auxC1(2)];
            rx_bindata=[rx_bindata auxC1(1)];
            auxC2=invertC1(slicingB(ODataPolX_Est(si,ii),ODataPolY_Est(si,ii)));
            rx_bindata=[rx_bindata auxC2(4)];
            rx_bindata=[rx_bindata auxC2(3)];
            rx_bindata=[rx_bindata auxC2(2)];
            rx_bindata=[rx_bindata auxC2(1)];
            rx_bindata = [rx_bindata 0];
        end;
    end;
end;
rx_data.bindata = rx_bindata;
```

El resultat és un vector anomenat *rx_data.bindata* que correspon a la informació binària rebuda. Si comparem aquesta informació amb la generada en l'emissor en forma de vector binari anomenat *data* podem quantificar el BER (Bit Error Rate) en recepció.

9.5.2 Algoritme de *demapping* PS-QPSK i implementació

Per implementar el *demapping* en la funció *demodulatorDigOFDMPS.m*, utilitzarem la funció de càlcul de la distància euclidiana *slicingB(Xcomplex,Ycomplex)* utilitzada en la desmodulació 6P-QPSK però modificada ja que solament ens cal calcular la distància entre el punt rebut i els diferents vectors del conjunt C_2 :

```
function res = slicingB(Xcomplex,Ycomplex)
% Retorna el punt de la constel·lació mes proper del punt rebut
mapDist=[-2,0,0,0,9;
          2,0,0,0,9;
          0,-2,0,0,9;
          0,2,0,0,9;
          0,0,-2,0,9;
          0,0,2,0,9;
          0,0,0,-2,9;
          0,0,0,2,9;
          ];

for jj=1:length(mapDist)
    mapDist(jj,5)=sqrt(((mapDist(jj,1)-real(Xcomplex))^2)+((mapDist(jj,2)-
    imag(Xcomplex))^2)+((mapDist(jj,3)-real(Ycomplex))^2)+((mapDist(jj,4)-
    imag(Ycomplex))^2));
end;
[M,I]=min(mapDist(:,5));
res=mapDist(I,[1,2,3,4]);

end
```

També utilitzarem la funció *deMapC2(vector)* definida en la funció *demodulatorDigOFDM6P.m*, que ens permetrà codificar el punt de la constel·lació rebut amb la paraula de tres bits corresponent. Finalment, la implementació del *demapping* PS-QPSK proposat és el següent:

```
%=====
% DEMAPPING PS-QPSK
%=====
rx_bindata=[];

for si=1:Frames-1-floor(ofdmParam.simbolosOFDM/25)
    for (ii=1: ofdmParam.subcarriers)
        auxC2=deMapC2(slicingB(ODataPolX_Est(si,ii),ODataPoly_Est(si,ii)));
        rx_bindata=[rx_bindata auxC2(3)];
        rx_bindata=[rx_bindata auxC2(2)];
        rx_bindata=[rx_bindata auxC2(1)];
    end
end

rx_data.bindata = rx_bindata;
```

9.5.3 Implementació de l'efecte PMD en la fibra

Com he esmentat en l'anterior punt, en el script 'FiberTx.m' és on fixen els paràmetres de la fibra abans de cridar al script *sspropv2*, passant els valors dels diferents paràmetres que configuren la fibra. Els més importants són 'alpha', que es correspon amb l'atenuació de la fibra, 'betap' que és un vector que conté els coeficients de les derivades parcials de freqüència (és a dir, la dispersió i els seus increments) i 'gamma', que defineix el coeficient de no-linealitat de la fibra. Quan es realitza la crida a la funció *sspropv2*, s'han de passar els 2 vectors de senyal (un per a cada estat de polarització ortogonal) i dues

vegades els paràmetres ('alpha' i 'betap'), un per a cada estat de polarització. Això seria correcte si assumíssim que la fibra és circular i ideal.

En la realitat, la fibra té una certa birefringència, és a dir, que la seva propagació és diferent dependent de l'estat de polarització en què es propagui. Dit d'una altra manera: 'alpha' i 'betap' no tenen que ser iguals per a cada estat de polarització. Concretament 'alpha' no sol variar massa, mentre que 'betap' sí. De fet, si introduïm coeficients addicionals a 'betap' aconseguim modelar el PMD i altres efectes relacionats amb la polarització. Així hauríem d'entrar diferents valors de 'beta' per a cada estat de polarització ortogonal que ens defineix com es propaga el senyal per la fibra.

Per tal d'implementar l'impacte d'aquest fenomen en la transmissió per la fibra, he definit unes noves variables que fan referència certes característiques de la fibra que són:

```
fiberParam.birr=1e-7;  
fiberParam.PMD=0.02e-12/(10*(3/2));  
fiberParam.lc=100;
```

fiberParam.birr fa referència al factor de birefringència de la fibra. La birefringència o doble refracció és una propietat de la matèria anisòtropa transparent que consisteix en el desdoblament d'un feix de llum incident en dos feixos polaritzats linealment en direccions perpendiculars entre si com si el material tingués dos índexs de refracció diferents. El valor més comú i que utilitzarem en les simulacions és $1e-7$.

fiberParam.PMD fa referència a la dispersió del mode de polarització de la fibra. El valor estàndard que utilitzarem en les simulacions serà de $0.02e-12/(10*(3/2))$.

fiberParam.lc fa referència a la distància de correlació, sent el valor típic entre 10 i 100. En les simulacions utilitzaré el valor 100.

També he modificat el script *fiberTx* per que modifiqui els valors de *betapa* i *betapb* d'acord a aquestes variables i modelitzar el comportament de la fibra:

```
if (isfield(fiberParam,'birr') )  
    beta0=fiberParam.birr; %This is the final value  
    beta1=(fiberParam.PMD*sqrt(2))/(4*sqrt(fiberParam.lc));  
    betapa=[beta0, beta1, beta2, beta3]; % dispersion array  
    betapb=[-beta0, -beta1, beta2, beta3]; % dispersion array  
else  
    betapa=[0,0,beta2, beta3]; % Dispersion array polarització A  
    betapb=[0,0,beta2, beta3]; % Dispersion array polarització B  
end;
```

Veiem que si es defineix el paràmetre *fiberParam.birr* es calcula l'impacte del PMD en la transmissió, i en cas de que no es defineixi aquest paràmetre es processa la senyal sense cap tipus de dispersió PMD.

10 Simulacions

He realitzat una sèrie de simulacions per avaluar les prestacions de les noves modulacions 6P-QPSK i PS-QPSK i comparar-les amb la modulació DP-QPSK. Les simulacions han estat implementades amb Matlab i els codis que s'adjunten a la memòria són els següents:

- *Simulacio1.m*: Nivell de BER/OSNR per un sistema de transmissió OFDM amb modulació 6P-QPSK per 64, 256 i 512 subportadores sense enllaç òptic i amb la mateixa proporció d'oversampling.
- *Simulacio2.m*: Nivell de BER/OSNR per un sistema de transmissió OFDM amb modulació 6P-QPSK per 64, 256 i 512 subportadores sense enllaç òptic i amb un oversampling inversament proporcional al nombre de subportadores.
- *Simulacio3.m*: Comparativa de modulacions 6P-QPSK, PS-QPSK i DP-QPSK a nivell de BER/OSNR per un sistema de transmissió OFDM amb 64 subportadores sense enllaç òptic.
- *Simulacio4.m*: Simulació de dos enllaços òptics OFDM de 50 trams de 100 Km (5.000 Km) de 512 i 256 subportadores respectivament amb modulació 6P-QPSK.
- *Simulacio5.m*: Prova d'abast d'un sistema OFDM de 512 subportadores amb modulació 6P-QPSK.
- *Simulacio6.m*: Prova d'abast d'un sistema OFDM de 512 subportadores amb modulació DP-QPSK.
- *Simulacio7.m*: Prova d'abast d'un sistema OFDM de 256 subportadores amb modulació 6P-QPSK.
- *Simulacio8.m*: Prova d'abast d'un sistema OFDM de 256 subportadores amb modulació DP-QPSK.
- *Simulacio9.m*: Prova dels efectes del PMD en un sistema OFDM de 256 subportadores amb modulació 6P-QPSK.
- *Simulacio10.m*: Comparativa de modulacions 6P-QPSK, PS-QPSK i DP-QPSK a nivell de BER/OSNR per un sistema de transmissió OFDM amb 64 subportadores sense enllaç òptic normalitzant el resultat de les corbes de BER pel bit-símbol de les tres modulacions.
- *Simulacio11.m*: Prova d'abast d'un sistema OFDM de 512 subportadores amb modulació PS-QPSK.

- *Simulacio12.m*: Prova d'abast d'un sistema OFDM de 512 subportadores amb modulació PS-QPSK.
- *Simulacio13.m*: Simulació de dos enllaços òptics OFDM de 50 trams de 100 Km (5.000 Km) per 512 i 40 trams de 125 Km (5.000 Km) per 256 subportadores respectivament amb modulació PS-QPSK.

Per millorar la precisió en els resultats i depenent del tipus de prova, les proves de BER les he realitzat en un interval de SNR variable. També he modificat els scripts per poder extreure per pantalla els resultats numèrics del BER i poder extreure conclusions d'una manera més acurada.

10.1 Proves *Back-to-Back* sense fibra

En aquesta primera simulació anomenada *Simulacio1.m* s'ha modelitzat el funcionament del la modulació 6P-QPSK amb absència dels efectes del canal de comunicacions. S'estudiarà el comportament per 64, 256 i 512 subportadores OFDM, executant la funció `Simulation64subcarriers6P` on 64 fa referència al nombre de subportadores OFDM. Els paràmetres d'entrada de la simulació *Simulacio1.m* seran els següents:

```
OFDMParameters.simbolosOFDM =120;
OFDMParameters.RSymb = 40e6; %SimbolRate (simb/seg)
OFDMParameters.RFfreqHz =6e9; % Hz
OFDMParameters.PC =0; % Percentatge de prefix cíclic
OFDMParameters.SimParam.simulations = 1; %nombre de simulacions
OFDMParameters.SimParam.BEROSNR = 15; %BERMAX sobre el que es realitza l'estudi
OFDMParameters.SimParam.BERFIBER = 0;
OFDMParameters.SimParam.ecualization = 0OFDMParameters.SimParam.fiber = 0;
OFDMParameters.constelacion = '6PPSK';
OFDMParameters.subcarriers = 64;
OFDMParameters.NFFT = OFDMParameters.subcarriers*4;
Simulation64subcarriers6P = ofdm4DSimulator6P(OFDMParameters,laserParam);
OFDMParameters.subcarriers = 256;
OFDMParameters.NFFT = OFDMParameters.subcarriers*4;
Simulation256subcarriers6P = ofdm4DSimulator6P(OFDMParameters,laserParam);
OFDMParameters.subcarriers = 512;
OFDMParameters.NFFT = OFDMParameters.subcarriers*4;
Simulation512subcarriers6P = ofdm4DSimulator6P(OFDMParameters,laserParam);
```

Donat que es manté el nombre de mostres per símbol proporcional al nombre de subportadores, el BER no varia significativament entre les tres execucions. Els paràmetres de transmissió són els següents:

Paràmetres de transmissió per nombre de subportadores	64 sc	256 Sc	512 sc
Total de mostres transmeses incloent preàmbul del 4%	31744	31744	30720
Nombre de mostres per cada símbol OFDM	256	1024	2048
Temps per mostra	9.7656e-11	2.4414e-11	1.2207e-11
Freqüència de mostreig	1.0240e+10	4.0960e+10	8.1920e+10
Gbits /seg.	10.2400	40.9600	81.9200
Símbols/segon Back-to-back	45e6	45e6	45e6
Signal_Bandwidth	2.6102e+09	1.0290e+10	2.0530e+10

Figura 29: Paràmetres de transmissió de *Simulacio1.m*

Si definim la variable `OFDMParameters.SimParam.Spectrum` a 1 se'ns mostra l'espectre OFDM transmès i rebut. A continuació es pot veure aquest espectre per 512 subportadores OFDM:

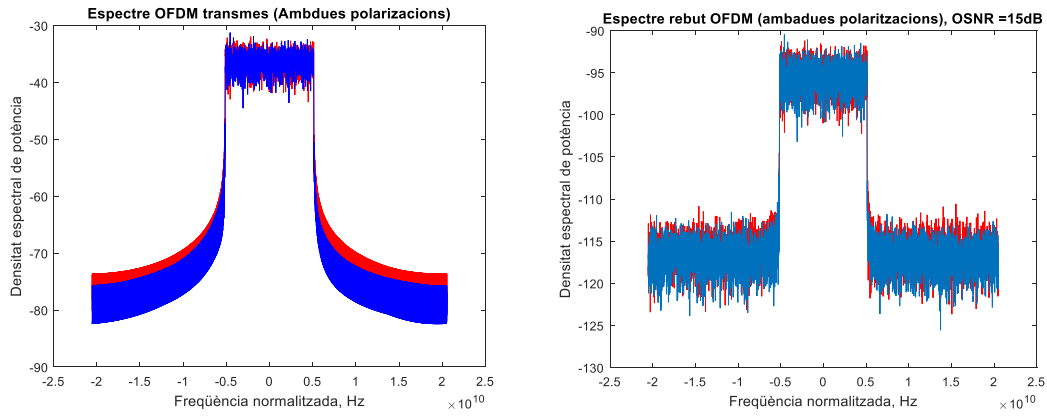


Figura 30: espectre transmès i rebut en la Simulacio1.m

A continuació també podem veure les constel·lacions rebudes i el BER obtingut. Veiem com el mateix rati de mostreig provoca una similitud en la definició de les constel·lacions rebudes:

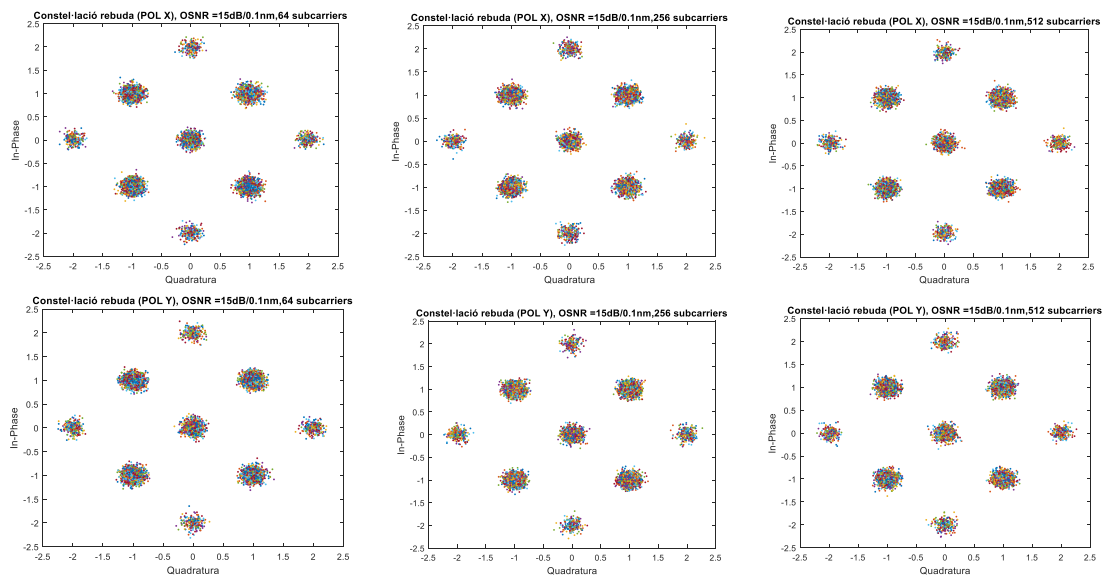


Figura 31: Constel·lacions rebudes en la Simulacio1.m

En la següent gràfica podem comprovar com la diferència de la taxa d'error en absència d'enllaç òptic correspon a la diferència de *oversampling* o sobre mostreig i no al nombre de subportadores emprat:

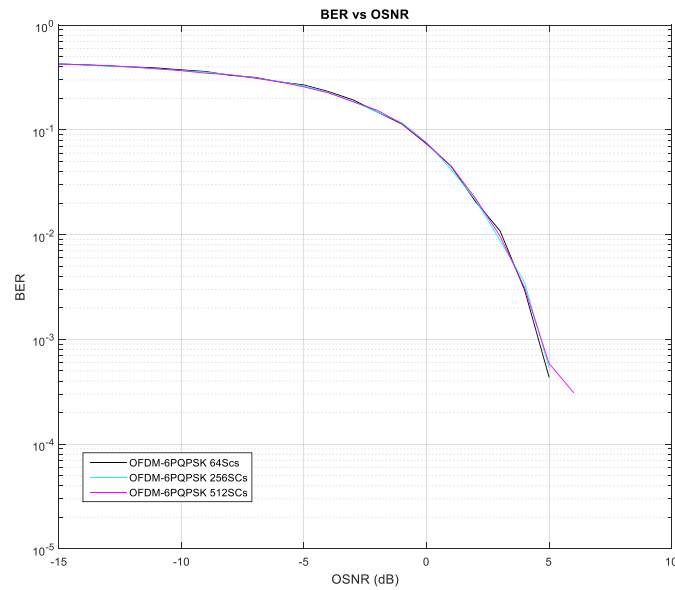


Figura 32: valors de BER obtinguts en la Simulacio1.m

Com he esmentat en l'inici d'aquest apartat, he afegit al codi del simulador una sortida numèrica dels valors de BER. Veiem com els valors són força similars en els tres escenaris :

SNR	BER 6P-QPSK 64 sc.	BER 6P-QPSK 256 sc.	BER 6P-QPSK 512 sc
-15	4.262955e-01	4.276221e-01	4.258743e-01
-14	4.188550e-01	4.164871e-01	4.190848e-01
-13	4.087302e-01	4.064595e-01	4.113033e-01
-12	3.995682e-01	3.997845e-01	3.973524e-01
-11	3.903770e-01	3.808968e-01	3.840216e-01
-10	3.746499e-01	3.688338e-01	3.678075e-01
-9	3.609944e-01	3.562021e-01	3.479973e-01
-8	3.320203e-01	3.362967e-01	3.358755e-01
-7	3.174020e-01	3.138170e-01	3.120350e-01
-6	2.876692e-01	2.894516e-01	2.875434e-01
-5	2.687908e-01	2.633202e-01	2.573165e-01
-4	2.324638e-01	2.280292e-01	2.255704e-01
-3	1.937442e-01	1.864224e-01	1.859499e-01
-2	1.480509e-01	1.479286e-01	1.539249e-01
-1	1.136788e-01	1.170678e-01	1.143043e-01
0	7.350023e-02	7.564057e-02	7.486979e-02
1	4.510971e-02	4.196600e-02	4.513889e-02
2	2.068744e-02	2.158166e-02	2.219742e-02
3	1.088352e-02	8.800287e-03	9.672619e-03
4	2.976190e-03	3.502155e-03	3.131200e-03
5	4.376751e-04	5.387931e-04	5.890377e-04
6	0	0	3.100198e-04
7	0	0	0

Figura 33: valors numèrics de BER obtinguts en la Simulacio1.m

Veiem com les tres modulacions tenen un BER de $2e-2$ a 2.5 dB.

En la segona simulació anomenada *Simulacio2.m* repetirem la mateixa configuració que l'anterior simulació però amb un factor de mostreig fix de 512 pels tres nombres de subportadores. Aquest fet farà que el BER/OSNR sigui major a mesura que augmenta el nombre de subportadores, ja que el factor d'*oversampling* disminueix. Els paràmetres de transmissió de *Simulacio2.m* són els següents:

<i>Paràmetres de transmissió per nombre de subportadores</i>	<i>64 sc</i>	<i>256 Sc</i>	<i>512 sc</i>
<i>Total de mostres transmeses incloent preàmbul del 4%</i>	63488	15872	7680
<i>Nombre de mostres per cada símbol OFDM (64 sc)</i>	512	512	512
<i>Temps per mostra</i>	4.8828e-11	4.8828e-11	4.8828e-11
<i>Freqüència de mostreig</i>	2.0480e+10	2.0480e+10	2.0480e+10
<i>Gbits /seg. Back-to-back</i>	10.2400	40.9600	81.9200
<i>Símbols/segon transmesos</i>	45e6	45e6	45e6
<i>Signal_Bandwidth</i>	2.6051e+09	1.0300e+10	2.0560e+10

Figura 34: paràmetres de transmissió de Simulacio2.m

Veiem com el BER augmenta i la definició en la recepció dels punt de recepció disminueix amb el nombre de subportadores:

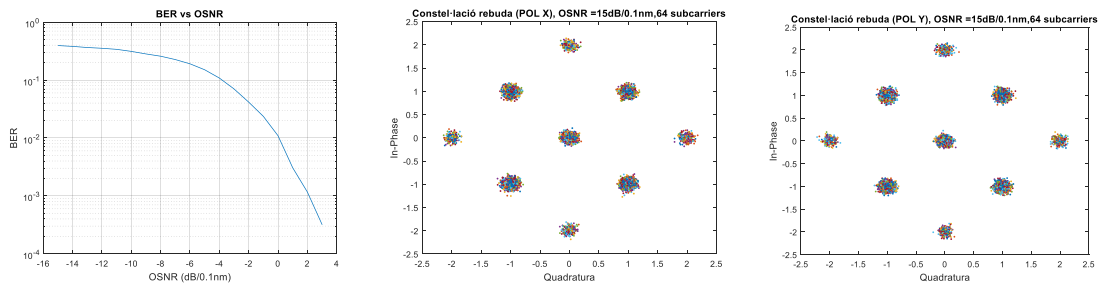


Figura 35: BER vs OSNR i constel·lacions rebudes per Simulacio2.m i 64 subportadores

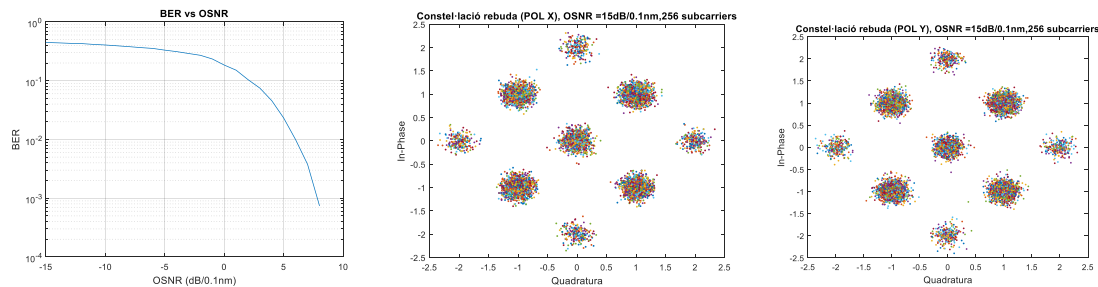


Figura 36: BER vs OSNR i constel·lacions rebudes per Simulacio2.m i 256 subportadores

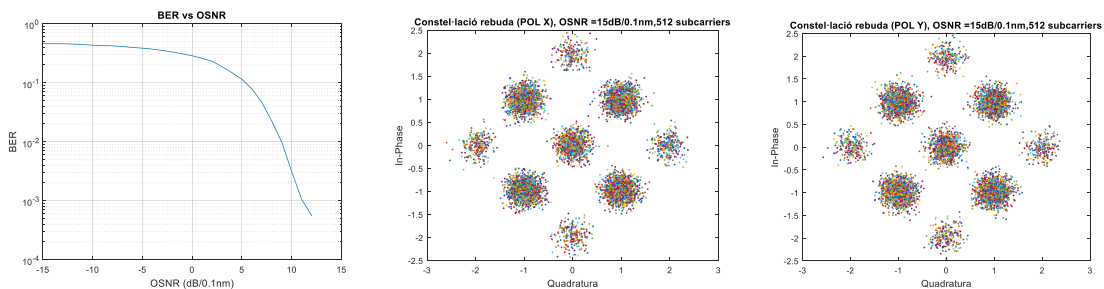


Figura 37: BER vs OSNR i constel·lacions rebudes per Simulacio2.m i 512 subportadores

A continuació podem comprovar la diferència de BER per cada nombre de subportadores:

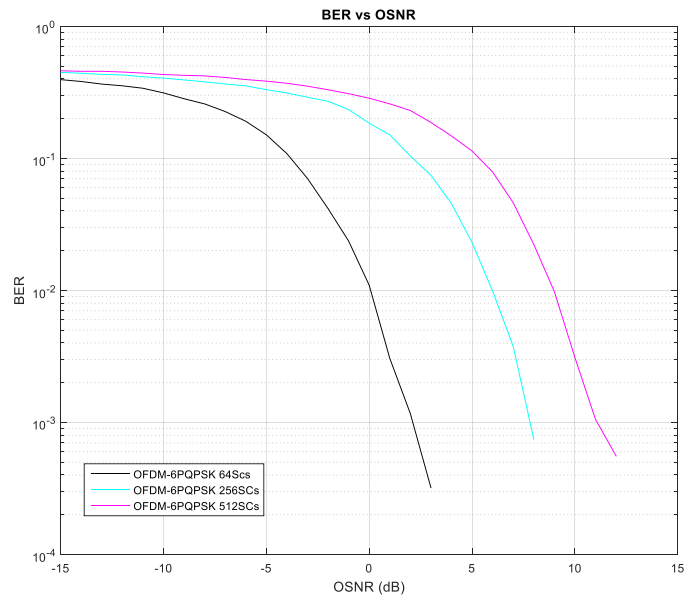


Figura 38: BER vs OSNR de Simulacio2.m per 64, 256 i 512 subportadores

El resultat numèric de BER respecte al OSNR és el següent:

SNR	BER 6P-QPSK 64 sc.	BER 6P-QPSK 256 sc.	BER 6P-QPSK 512 sc
-15	3.944620e-01	4.471683e-01	4.601625e-01
-14	3.817986e-01	4.409124e-01	4.560082e-01
-13	3.649335e-01	4.325311e-01	4.561632e-01
-12	3.541375e-01	4.280412e-01	4.503968e-01
-11	3.394024e-01	4.146312e-01	4.420263e-01
-10	3.129669e-01	4.051125e-01	4.313616e-01
-9	2.827089e-01	3.923910e-01	4.250992e-01
-8	2.583158e-01	3.795199e-01	4.210689e-01
-7	2.260738e-01	3.655711e-01	4.099392e-01
-6	1.909139e-01	3.539571e-01	3.948413e-01
-5	1.507645e-01	3.311183e-01	3.839906e-01
-4	1.085142e-01	3.124102e-01	3.700087e-01
-3	7.040733e-02	2.902000e-01	3.518415e-01
-2	4.163749e-02	2.702646e-01	3.305122e-01
-1	2.363445e-02	2.338961e-01	3.086868e-01
0	1.094188e-02	1.844468e-01	2.852803e-01
1	3.063725e-03	1.508022e-01	2.585565e-01
2	1.167134e-03	1.041068e-01	2.303757e-01
3	3.209617e-04	7.453305e-02	1.869110e-01
4	0	4.546815e-02	1.476625e-01
5	0	2.304837e-02	1.135913e-01
6	0	9.877874e-03	7.874504e-02
7	0	3.771552e-03	4.616195e-02
8	0	7.483238e-04	2.229043e-02
9	0	0	9.796627e-03
10	0	0	3.100198e-03
11	0	0	1.054067e-03
12	0	0	5.580357e-04
13	0	0	0

Figura 39: valors numèrics de BER obtinguts en la Simulacio2.m

Podem comparar els tres escenaris quantificant l'OSNR per a un BER de 2e-2:

BER	OSNR 6P-QPSK 64 sc.	OSNR 6P-QPSK 256 sc.	OSNR 6P-QPSK 512 sc
2e-2	-0.8 dB	5.2 dB	8.2 dB

Figura 40: OSNR per a un BER de 2e-2

En el script anomenat *Simulacio3.m* compararem les modulacions DP-QPSK, 6P-QPSK i PS-QPSK per un sistema OFDM de 64 subportadores. Per a comparar les modulacions cal desactivar la equalització i canviar el factor de remostreig a una mostra per símbol. Definirem un nombre de símbols de 12000 per poder quantificar el BER fins al $1e-5$, malgrat que la simulació pot trigar fins a 30 minuts. Els paràmetres de simulació *Simulacio3.m* seran els següents:

```
OFDMParameters.simbolosOFDM =12000;
OFDMParameters.subcarriers = 64;
OFDMParameters.NFFT = OFDMParameters.subcarriers;
OFDMParameters.RSymb = 40e6; %SimbolRate (simb/seg) en línia
OFDMParameters.RFfreqHz =6e9; % Hz
OFDMParameters.PC =0; % Percentatge de prefix cíclic
OFDMParameters.SimParam.simulations = 1; %nombre de simulacions
OFDMParameters.SimParam.BEROSNR = 15; %BERMAX sobre el que es realitza l'estudi
OFDMParameters.SimParam.BERFIBER = 0;
OFDMParameters.SimParam.ecualization = 0;% No activa el bloc d'equalització i estimació
OFDMParameters.SimParam.fiber = 0; %No activa la transmissió per fibra òptica,
```

Els paràmetres de transmissió per modulació són els següents:

Paràmetres de transmissió per nombre de subportadores i modulació	64 sc (6P-QPSK)	64 sc (DP-QPSK)	64 sc (PS-QPSK)
Total de mostres transmises incloent preàmbul del 4%	79872	79872	79872
Nombre de mostres per cada símbol OFDM (64 sc)	64	64	64
Temps per mostra	3.9062e-10	3.9062e-10	3.9062e-10
Freqüència de mostreig	2.5600e+09	2.5600e+09	2.5600e+09
Gbits /seg.	10.2400	10.2400	10.2400
Símbols/segon Back-to-back	45e6	40e6	30e6
Signal_Bandwidth	2.6413e+09	2.6051e+09	2.6413e+09

Figura 41: paràmetres de transmissió de *Simulacio3.m*

A continuació podem observar els valors de BER obtinguts per les modulacions 6P-QPSK, DP-QPSK i PS-QPSK amb 64 subportadores OFDM:

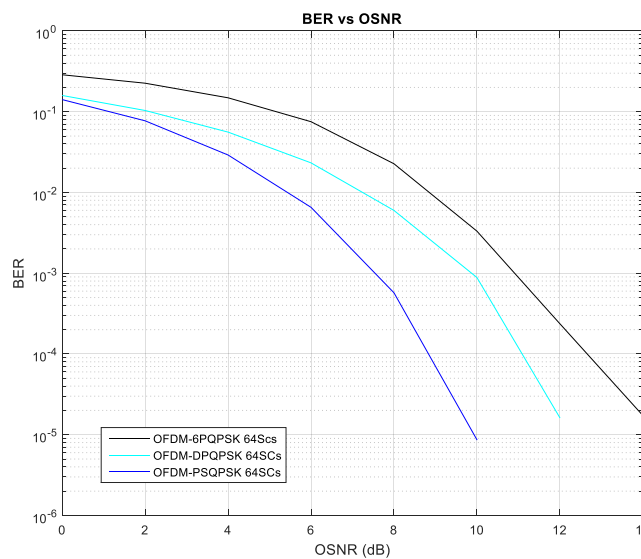


Figura 42: valors de BER obtinguts per les modulacions 6P-QPSK, DP-QPSK i PS-QPSK amb 64 subportadores OFDM

Le valors numèrics de BER per SNR de l'anterior prova són els següents:

SNR	BER 6P-QPSK 64sc.	BER DP-QPSK 64sc.	BER PS-QPSK 64sc.
0	2.869029e-01	1.590453e-01	1.413331e-01
2	2.256220e-01	1.036020e-01	7.717369e-02
4	1.487785e-01	5.594180e-02	2.911280e-02
6	7.525658e-02	2.326809e-02	6.550598e-03
8	2.268673e-02	5.994579e-03	5.733945e-04
10	3.324530e-03	8.861551e-04	8.687795e-06
12	2.374664e-04	1.628962e-05	0
14	1.737559e-05	0	0

Figura 43: valors numèrics de BER / SNR per 6P-QPSK, DP-QPSK i PS-QPSK

Si comparem les tres modulacions tenim un OSNR per a un BER de $2e-2$ de:

BER	OSNR 6P-QPSK 64 sc.	OSNR DP-QPSK 64 sc.	OSNR PS-QPSK 64 sc
$2e-2$	8.4 dB	6.4 dB	4.5 dB

Figura 44: OSNR per a un BER de $2e-2$

Les constel·lacions rebudes per les dues polaritzacions per cadascuna de les modulacions son les següents:

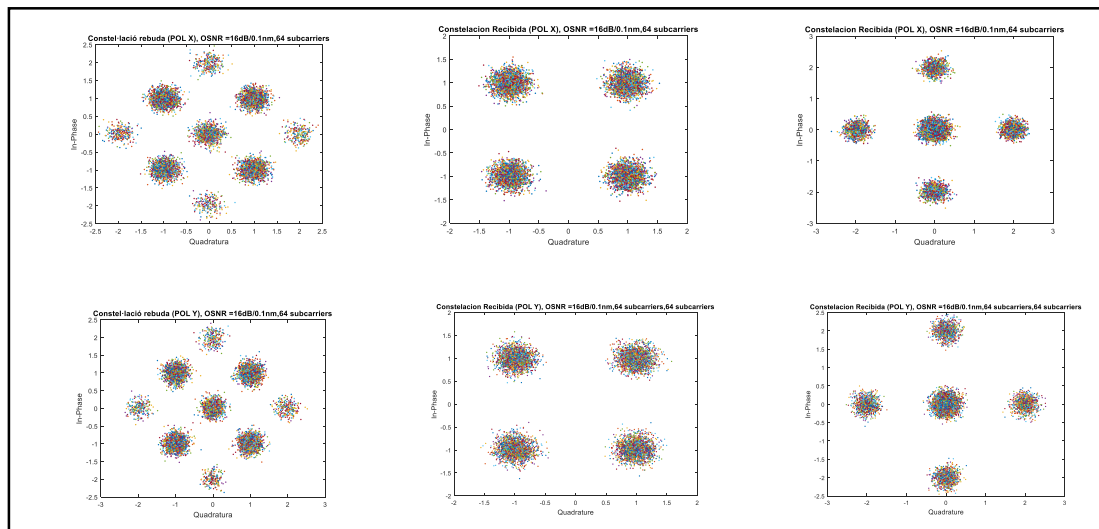


Figura 45: constel·lacions 6P-QPSK, DP-QPSK i PS-QPSK rebudes

Veiem com el BER és menor en la modulació PS-QPSK per l'augment de la distància mínima en un factor de $\sqrt{2}$ respecte a les altres modulacions. Respecte a les modulacions DP-QPSK i PS-QPSK, la primera té un millor comportament en termes absoluts de BER / OSNR. En la prova Simulacio10.m es normalitza la representació del OSNR respecte al nombre de bits per símbol de cada modulació (4,5 en el cas de 6P-QPSK, i 4 bits en el cas de DP-QPSK i 3 en el cas de PS-QPSK):

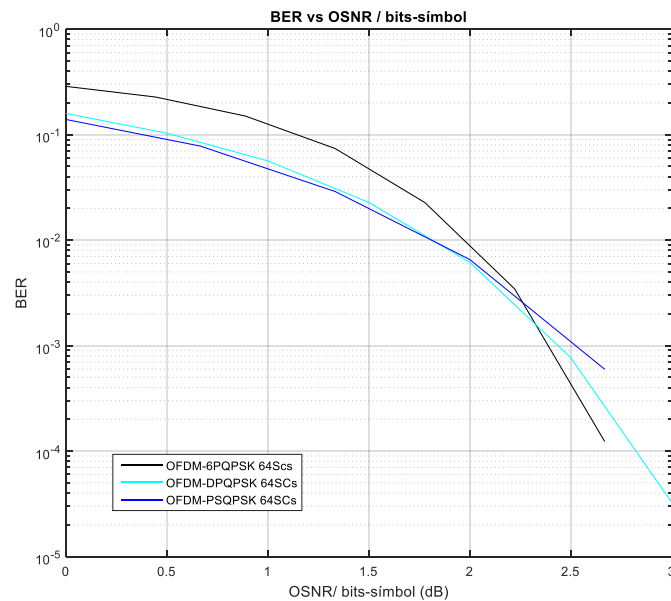


Figura 46: valors de BER (6P-QPSK, DP-QPSK i PS-QPSK) amb 64 subportadores OFDM normalitzat a bit/símbol

Veiem com els resultats en indiquen com la modulació 6P-QPSK té un excel·lent compromís entre BER i eficiència espectral a partir dels 10e-3.

10.2 Prova Back-to-Back amb enllaç òptic

A continuació realitzo una sèrie de simulacions amb enllaços òptics per comprovar el rendiment de la modulació 6P-QPSK i PS-QPSK amb l'efecte del canal òptic. En aquest punt tenim una sèrie de paràmetres addicionals que cal configurar. Per activar en la simulació l'aparició del canal òptic caldrà definir la variable adient de la següent manera:

```
OFDMParameters.SimParam.fiber = 1;
```

Caldrà definir la potència d'emissió de l'emissor i de l'oscil·lador local. Els marges de potència acceptables per evitar l'aparició d'efectes no lineals del sistema són entre -6 dBm i + 6 dBm. Per aquesta primera simulació definirem una potència d'emissió de 2 mW (3dBm) :

```
laserParam.power = 2e-3;  
laserParam.loPower = 2e-3;
```

També caldrà definir la longitud de l'enllaç en Kilòmetres:

```
fiberParam.fiberLength= 400;
```

Ara, cal introduir els paràmetres d'atenuació, de dispersió cromàtica, de birefringència, de PMD i de correlació. També caldrà definir la longitud d'ona de treball:

```
fiberParam.waveLength = 1550e-9;  
fiberParam.attenuation = 0.2;  
fiberParam.dispersionChrom =17;  
fiberParam.birr=1e-7;  
fiberParam.PMD=0.02e-12/(10^(3/2));  
fiberParam.lc=100;
```

També podem utilitzar fibra compensadora de dispersió. Per activar-la caldrà definir-ho d'aquesta manera:

```
fiberParam.DCF = 'si';
```

Podem utilitzar varis trams amb amplificadors EDFA i en cas de que utilitzem amplificador caldrà també definir la figura soroll de l'amplificador:

```
fiberParam.tramos = 3;  
fiberParam.EDFA = 'si';  
fiberParam.FN_dB=5.5;
```

Normalment activarem l'equalització i el prefix cíclic:

```
OFDMPParameters.PC =1/16;  
OFDMPParameters.SimParam.ecualization = 1;
```

10.3 Abast del sistema

En les simulacions *Simulacio5.m*, *Simulacio6.m*, *Simulacio7.m*, *Simulacio8.m*, *Simulació11* i *Simulació12* realitzarem una prova d'abast per a 256 i 512 subportadores amb modulacions 6P-QPSK, DP-QPSK i PS-QPSK i d'aquesta manera podrem definir la distància d'un enllaç en les tres modulacions. La velocitat de transmissió serà de l'ordre dels 122 Gb/seg per 256 subportadores i de 245 Gb/seg per 512 subportadores. Caldrà configurar els paràmetres d'entrada de les simulacions de la següent manera:

```
OFDMPParameters.fixedp = 1;  
OFDMPParameters.SimParam.BEROSNR=0;  
OFDMPParameters.SimParam.BERFIBER =1;
```

Els paràmetre OFDM i de prova seran els següents:

```
OFDMPParameters.simbolosOFDM =40;  
OFDMPParameters.constelacion = 'xxxPSK';  
OFDMPParameters.NFFT = OFDMPParameters.subcarriers*8;  
OFDMPParameters.RSymb = 120e6;  
OFDMPParameters.RFfreqHz =6e9;  
OFDMPParameters.SimParam.simulations = 1;  
OFDMPParameters.subcarriers = xxx;  
fiberParam.waveLength = 1550e-9;  
fiberParam.attenuation = 0.2;  
fiberParam.dispersionChrom =17;  
fiberParam.birr=1e-7;  
fiberParam.PMD=0.02e-12/(10^(3/2));  
fiberParam.lc=100;
```

Els paràmetres de transmissió de la prova d'abast per 512 subportadores són els següents:

<i>Paràmetres de transmissió de la prova d'abast</i>	<i>6P-QPSK 512 sc.</i>	<i>DP-QPSK 512</i>	<i>PS-QPSK 512 sc</i>
--	------------------------	--------------------	-----------------------

		sc.	
Total de mostres transmeses incloent preàmbul del 4%	167936	167936	167936
Total de mostres incloent percentatge de prefix cíclic a afegir	178432	178432	178432
Nombre de mostres per cada símbol OFDM (512 sc)	4096	4096	4096
Temps per mostra	2.0345e-12	2.0345e-12	2.0345e-12
Freqüència de mostreig	4.9152e+11	4.9152e+11	4.9152e+11
Gbits /seg.	245.7600	245.7600	245.7600
Símbols/segon	135e6	120e6	90e6
Signal_Bandwidth	6.1561e+10	6.1561e+10	6.1561e+10
Potència d'emissió	4e-3 W	4e-3 W	4e-3 W

Figura 47: paràmetres de transmissió de Simulacio4.m,i Simulacio5.m i Simulacio11.m

A continuació podem comprovar el resultat de la prova d'abast per 512 subportadores. Veiem que són lleugerament millors en el cas de modulació PS-QPSK en termes de BER/OSNR:

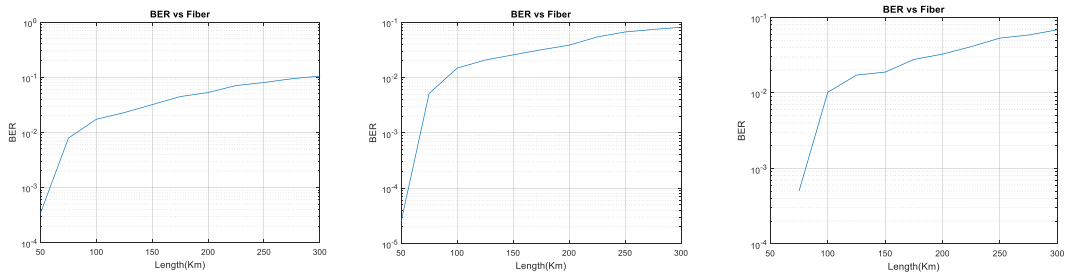


Figura 48: resultat de la prova d'abast per modulacions 6P-QPSK, DP-QPSK, PS-QPSK i 512 subportadores OFDM

Els resultats numèrics de la prova són els següents:

Km.	BER 6P-QPSK 512 sc.	BER DP-QPSK 512 sc	BER PS-QPSK 512 sc
50	3.449964e-04	2.472310e-05	0
75	7.868145e-03	5.123863e-03	5.008013e-04
100	1.727208e-02	1.471643e-02	1.018296e-02
125	2.282541e-02	2.059434e-02	1.721087e-02
150	3.182870e-02	2.555751e-02	1.886351e-02
175	4.448228e-02	3.160230e-02	2.772770e-02
200	5.271768e-02	3.822191e-02	3.266894e-02
225	7.052395e-02	5.380983e-02	4.103232e-02
250	7.999466e-02	6.627027e-02	5.323518e-02
275	9.372774e-02	7.367484e-02	5.846020e-02
300	1.047009e-01	8.046133e-02	6.847623e-02

Figura 49: resultats de BER/Km de Simulacio4.m i Simulacio5.m per modulació 6P-QPSK i DP-QPSK i 512 subportadores OFDM

Si comparem els tres escenaris tenim distància màxima per a BER de 2e-2 de:

BER	Km. 6P-QPSK 512 sc.	Km. DP-QPSK 512 sc.	Km. PS-QPSK 512 sc
2e-2	110 Km	123 Km	155 Km

Figura 50: Km per a un BER de 2e-2

Ara cal decidir quin es el valor de BER acceptable per a determinar la màxima distància d'un enllaç sense amplificador. Si partim de la possibilitat de fer servir un codi de recuperació d'errors tipus FEC (es pot fer servir un esquema de codificació que s'anomena '*soft-decision FEC*', que implica que es fa la correcció alhora que es desmodulen els símbols) podem recuperar errors convertint una BER de $2e-2$ a una $BER < 1e-10$ (considerat com a lliure d'errors). També podem optar per una versió més avançada (i propietària d'alguns fabricants) que permetria recuperar errors a una BER de $4.5e-2$. Per tant, podem definir una BER de referència de $2e-2$ com a acceptable.

Cal esmentar que aquesta prova s'ha realitzat sense fibra compensadora de dispersió (*DCF*) per lo que podem concloure que 100 Km serà una distància adequada per cadascun dels trams que pot constituir un enllaç de molt llarga distància en tres modulacions i 512 subportadores amb els paràmetres de transmissió definits anteriorment.

En les simulacions *Simulacio7.m*, *Simulacio8.m* i *Simulacio12.m* es realitzen les mateixes proves d'abast per sistemes OFDM de 256 subportadores i per modulacions 6P-QPSK, DP-QPSK i PS-QPSK. Els paràmetres de transmissió són els següents:

Paràmetres de transmissió de la prova d'abast	6P-QPSK 256 sc.	DP-QPSK 256 sc.	PS-QPSK 256 sc.
Total de mostres transmeses incloent preàmbul del 4%	169984	169984	169984
Total de mostres incloent percentatge de prefix cíclic a afegir	180608	180608	180608
Nombre de mostres per cada símbol OFDM (64 sc)	2048	2048	2048
Temps per mostra	$4.0690e-12$	$4.0690e-12$	$4.0690e-12$
Freqüència de mostreig	$2.4576e+11$	$2.4576e+11$	$2.4576e+11$
Gbits /seg.	122.8800	122.8800	122.8800
Símbols/segon (Back-to-Back)	135e6	120e6	90e6
Signal Bandwidth	$3.0841e+10$	$3.0841e+10$	$3.0841e+10$
Potència d'emissió	$4e-3$ W	$4e-3$ W	$4e-3$ W

Figura 51: paràmetres de transmissió de *Simulacio6.m* i *Simulacio7.m*

Veiem que en aquest cas la modulació PS-QPSK té unes millors prestacions en termes d'abast i podem constituir enllaços de fins a 300 Km per a una BER màxima de $2e-2$. També podem concloure que totes les modulacions poden constituir un enllaç de 200 km com a mínim:

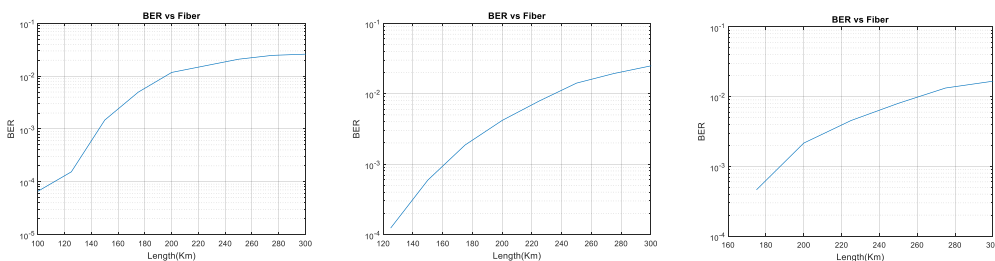


Figura 52: resultat de *Simulacio6.m* i *Simulacio7.m* per modulació 6P-QPSK i DP-QPSK i 256 subportadores OFDM

Els resultats numèrics de la prova són els següents:

Km.	BER 6P-QPSK 256 sc.	BER DP-QPSK 256 sc.	BER PS-QPSK 256 sc.
50	0	0	0
75	0	0	0
100	6.592827e-05	0	0
125	1.538326e-04	1.236155e-04	0
150	1.472398e-03	5.933544e-04	0
175	4.966596e-03	1.866594e-03	4.614979e-04
200	1.174622e-02	4.178204e-03	2.159151e-03
225	1.567994e-02	7.899031e-03	4.549051e-03
250	2.091025e-02	1.415398e-02	8.026767e-03
275	2.470113e-02	1.933347e-02	1.331751e-02
300	2.605265e-02	2.487144e-02	1.666337e-02

Figura 53: resultats de la prova d'abast per modulació 6P-QPSK, DP-QPSK, PS-QPSK i 256 subportadores OFDM

Si comparem els tres escenaris tenim distància màxima per a BER de $2e-2$ de:

BER	Km. 6P-QPSK 256 sc.	Km. DP-QPSK 256 sc.	Km. PS-QPSK 256 sc
$2e-2$	247 Km	275 Km	>300 Km

Figura 54: Km per a un BER de $2e-2$

En el script *Simulacio4.m* modelitzarem dos enllaços de 5000 Km amb 50 trams de 100 Km per modulació 6P-QPSK amb transmissió òptica OFDM de 512 subportadores (245 Gb/seg) i 256 subportadores (122 Gb/seg). Cada 100 Km s'amplificarà la senyal mitjançant un amplificador EDFA amb una figura soroll de 5,5 dB i s'equiparan totes les seccions amb fibres compensadores de dispersió (DCF). Altrament, definirem una potència d'emissió de 4 mw (+6 dBm) que és la màxima potència recomanable per no tenir problemes amb els efectes no lineals. Els paràmetres de configuració de la simulació seran els següents:

```

OFDMParameters.simbolosOFDM = 60;
OFDMParameters.subcarriers = 256;
OFDMParameters.constelacionn = '6P-QPSK';
OFDMParameters.NFFT = OFDMParameters.subcarriers * 8;
OFDMParameters.RSymb = 120e6; %SimbolRate (simb/seg)
OFDMParameters.RFfreqHz = 6e9; % Hz
OFDMParameters.PC = 1/16
OFDMParameters.fixedp = 1;
OFDMParameters.SimParam.simulations = 1;
OFDMParameters.SimParam.BEROSNR = 20;
OFDMParameters.SimParam.BERFIBER = 0;
OFDMParameters.SimParam.ecualization = 1;
OFDMParameters.SimParam.fiber = 1;
OFDMParameters.SimParam.fixedp = 1;
laserParam.power = 4e-3;%W
laserParam.loPower = 4e-3;%W
fiberParam.fiberLength = 100;%Km
fiberParam.tramos = 50;
fiberParam.EDFA = 'si';
fiberParam.DCF = 'si';
fiberParam.waveLength = 1550e-9; % (m)
fiberParam.attenuation = 0.2; %e-3 dB/Km
fiberParam.dispersionChrom = 17; %s/mm^2
fiberParam.FN_dB = 5.5;
fiberParam.birr = 1e-7; % Birefringència
fiberParam.PMD = 0.02e-12 / (10^(3/2));
fiberParam.lc = 100;

```

A continuació veiem els paràmetres de transmissió de la simulació:

<i>Paràmetres de transmissió dels enllaços de 5000 Km</i>	<i>6P-QPSK 256 sc. 5000 Km (50 x 100)</i>	<i>6P-QPSK 512 sc. 5000 Km (50 x 100)</i>
<i>Total de mostres transmises incloent preàmbul del 4%</i>	126976	253952
<i>Total de mostres incloent percentatge de prefix cíclic a afegir</i>	134912	269824
<i>Nombre de mostres per cada símbol OFDM (64 sc)</i>	2048	4096
<i>Temps per mostra</i>	4.0690e-12	2.0345e-12
<i>Freqüència de mostreig</i>	2.4576e+11	4.9152e+11
<i>Gbits /seg.</i>	122.8800	245.7600
<i>Símbols/segon (Back-to-Back)</i>	135e6	135e6
<i>Signal Bandwidth</i>	3.0841e+10	6.1561e+10
<i>Potència d'emissió</i>	4e-3 W	4e-3 W

Figura 55: paràmetres de transmissió de Simulacio8.m

A continuació es mostren les constel·lacions rebudes en els dos sistemes:

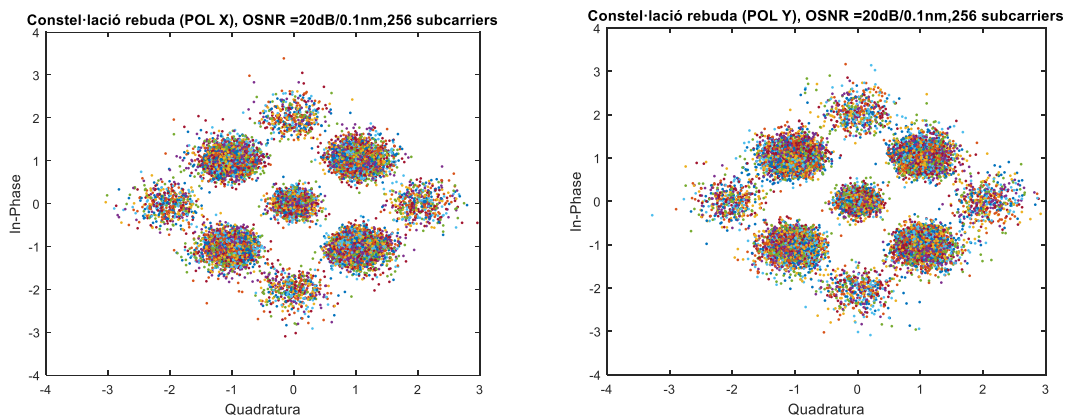


Figura 56: constel·lacions rebudes en Simulacio8.m per 256 subportadores OFDM

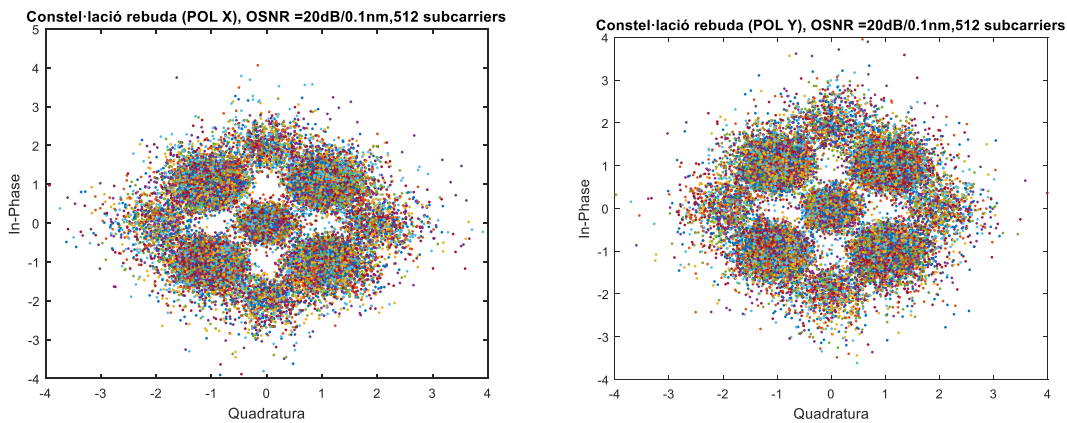


Figura 57: constel·lacions rebudes en Simulacio8.m per 256 i 512 subportadores OFDM

El resultat de la simulació ens confirma que és possible constituir un enllaç OFDM òptic de molt llarga distància amb modulació 6P-QPSK i alta velocitat (més de 245 Gb/seg):

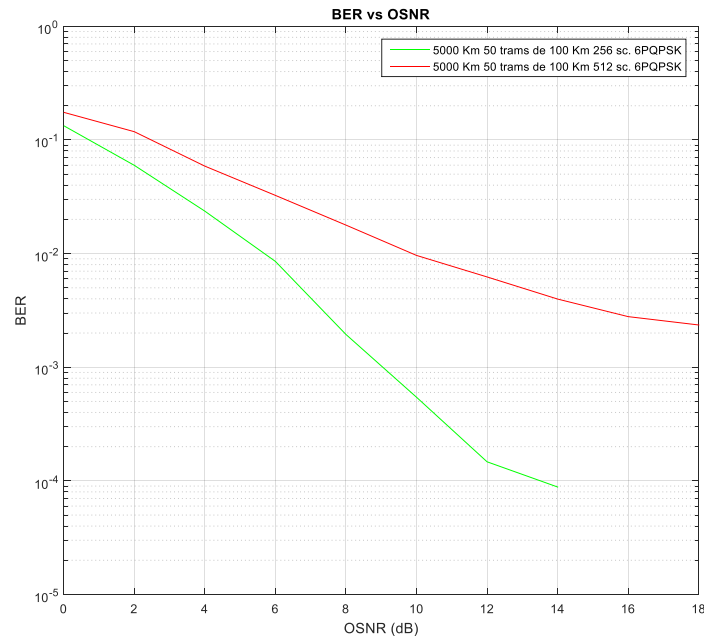


Figura 58: valors de BER per modulació 6P-QPSK amb 256 i 512 subportadores OFDM i 50 trams de 100 Km

Els resultats numèrics de BER/OSNR ens confirmen la viabilitat de constituir enllaços OFDM òptics amb modulació 6P-QPSK d'alta velocitat i llarga distància:

dB	6P-QPSK 256 sc. 5000 Km (50 x 100)	6P-QPSK 512 sc. 5000 Km (50 x 100)
0	1.334452e-01	1.750235e-01
2	5.980756e-02	1.182542e-01
4	2.362877e-02	5.883651e-02
6	8.533427e-03	3.252266e-02
8	1.942090e-03	1.781721e-02
10	5.443738e-04	9.636888e-03
12	1.471281e-04	6.230873e-03
14	8.827684e-05	3.979814e-03
16	0	2.788077e-03
18	0	2.354049e-03

Figura 59: valors de BER per modulació 6P-QPSK amb 256 i 512 subportadores OFDM i 50 trams de 100 Km

Si comparem els dos escenaris tenim un OSNR per a un BER de 2e-2 de:

BER	OSNR 6P-QPSK 256 sc. 5000 Km (50 x 100)	OSNR 6P-QPSK 512 sc. 5000 Km (50 x 100)
2e-2	4.4 dB	7.7 dB

Figura 60: OSNR per a un BER de 2e-2

A continuació repetirem l'anterior simulació però per modulació PS-QPSK. En el script *Simulacio13.m* modelitzarem uns enllaços de 5000 Km amb 50 trams de 100 Km amb modulació PS-QPSK i transmissió òptica OFDM de 512 subportadores (245 Gb/s) i un altre enllaç de 5000 Km però amb 40 trams de 125 Km amb modulació PS-QPSK amb transmissió òptica OFDM de 256 subportadores (122 Gb/s). Cada tram s'amplificarà la senyal mitjançant un amplificador EDFA amb una figura soroll de

5,5 dB i s'equiparan totes les seccions amb fibres compensadores de dispersió (DCF). Altrament, definirem una potència d'emissió de 4 mw (+6 dBm). Els paràmetres de configuració de la simulació seran els següents:

```

OFDMPParameters.simbolosOFDM = 60;
OFDMPParameters.subcarriers = 256;
OFDMPParameters.constelacionn = 'PS-QPSK';
OFDMPParameters.NFFT = OFDMPParameters.subcarriers * 8;
OFDMPParameters.RSymb = 120e6; %SimbolRate (simb/seg)
OFDMPParameters.RFfreqHz =6e9; % Hz
OFDMPParameters.PC =1/16 ; %Percentatge de prefix cíclic
OFDMPParameters.fixedp = 1;
OFDMPParameters.SimParam.simulations = 1
OFDMPParameters.SimParam.BEROSNR = 20;
OFDMPParameters.SimParam.BERFIBER = 0;
OFDMPParameters.SimParam.ecualization = 1;
OFDMPParameters.SimParam.fiber = 1;
OFDMPParameters.SimParam.fixedp = 1;
laserParam.power = 4e-3;%W
laserParam.loPower = 4e-3;%W
fiberParam.fiberLength =125;%Km
fiberParam.tramos =40;
fiberParam.EDFA = 'si';
fiberParam.DCF = 'si';
fiberParam.waveLength = 1550e-9; % (m)
fiberParam.attenuation = 0.2; %e-3 dB/Km
fiberParam.dispersionChrom =17; %s/mm^2
fiberParam.FN_dB=5.5;
fiberParam.birr=1e-7; % Birefringència
fiberParam.PMD=0.02e-12/(10^(3/2));
fiberParam.lc=100;
    
```

A continuació veiem els paràmetres de transmissió de la simulació:

Paràmetres de transmissió dels enllaços de 5000 Km	PS-QPSK 256 sc. 5000 Km (40 x 125)	PS-QPSK 512 sc. 5000 Km (50 x 100)
Total de mostres transmeses incloent preàmbul del 4%	126976	253952
Total de mostres incloent percentatge de prefix cíclic a afegir	134912	269824
Nombre de mostres per cada símbol OFDM (64 sc)	2048	4096
Temps per mostra	4.0690e-12	2.0345e-12
Freqüència de mostreig	2.4576e+11	4.9152e+11
Gbits /seg.	122.8800	245.7600
Símbols/segon (Back-to-Back)	90e6	90e6
Signal Bandwidth	3.0841e+10	6.1561e+10
Potència d'emissió	4e-3 W	4e-3 W

Figura 61: paràmetres de transmissió de Simulacio13.m

A continuació podem veure els punts rebuts pel primer enllaç de 256 subportadores OFDM i 40 trams de 125 Km:

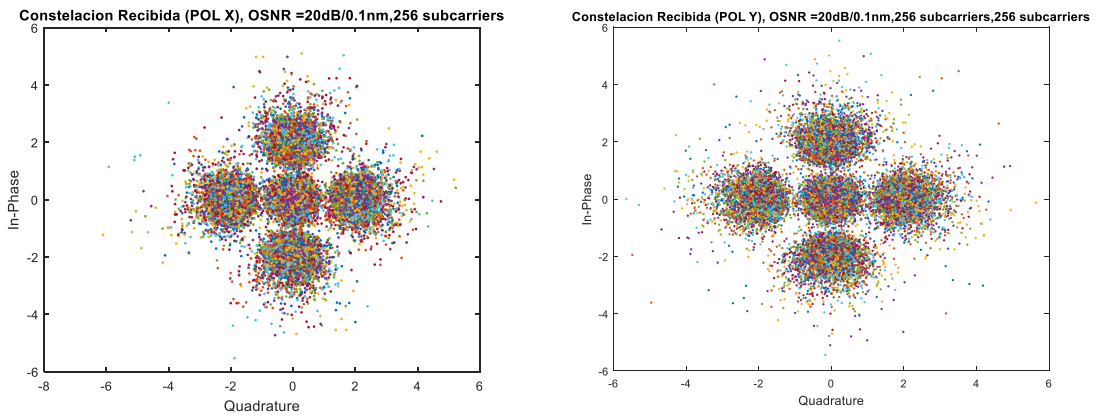


Figura 62: constel·lacions rebudes en Simulacio13.m per 256 subportadores OFDM (40 x 125 Km)

També podem veure els punts rebuts pel segon enllaç de 512 subportadores OFDM i 50 trams de 100 Km:

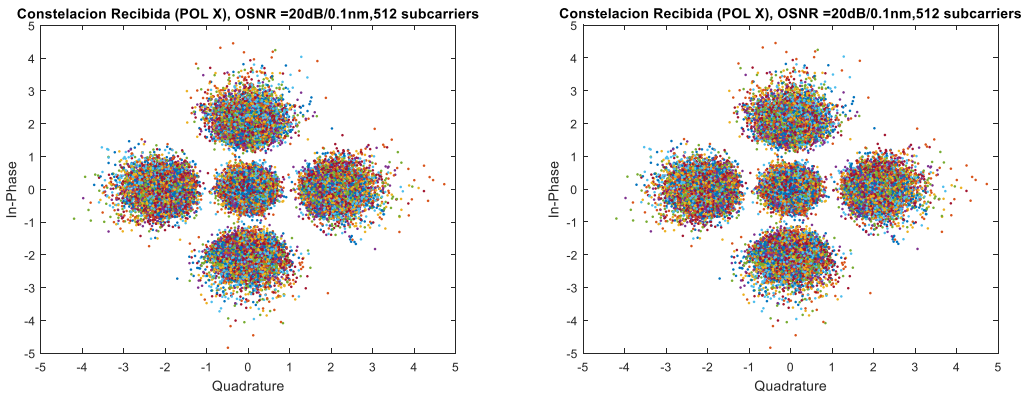


Figura 63: : constel·lacions rebudes en Simulacio13.m per 512 subportadores OFDM (50 x 100 Km)

Finalment, en la gràfica de BER / OSNR podem veure el bon comportament d'aquest tipus de modulació en enllaços de llarga distància:

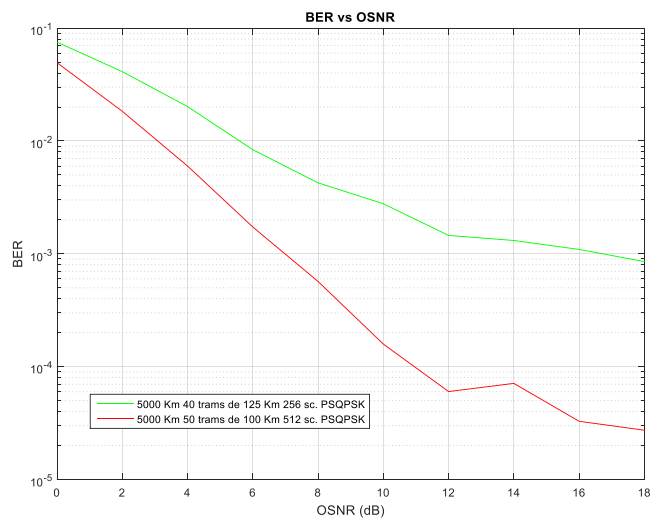


Figura 64:valors de BER per modulació PS-QPSK amb 256 i 512 subportadores OFDM i 40 trams de 125 Km

Els resultats numèrics de BER/OSNR ens confirmen que s'obtenen uns resultats una mica millors en termes de BER en la modulació PS-QPSK respecte a les altres modulacions 4D a l'hora de constituir una enllaç de comunicacions amb molts trams òptics:

dB	PS-QPSK 256 sc. 5000 Km (40x 125)	PS-QPSK 512 sc. 5000 Km (50 x 100)
0	7.521446e-02	4.941352e-02
2	4.115240e-02	1.833859e-02
4	2.018776e-02	6.001619e-03
6	8.370536e-03	1.728817e-03
8	4.256390e-03	5.689776e-04
10	2.779237e-03	1.586572e-04
12	1.455270e-03	6.018032e-05
14	1.313025e-03	7.112220e-05
16	1.094188e-03	3.282563e-05
18	8.534664e-04	2.735469e-05

Figura 65: valors de BER per modulació PS-QPSK amb 256 sc. (40 x 125 km) i 512 sc (50 x 100 Km)

Si comparem els dos escenaris tenim un OSNR per a un BER de 2e-2 de:

BER	OSNR PS-QPSK 256 sc. 5000 Km (40x 125)	OSNR PS-QPSK 512 sc. 5000 Km (50 x 100)
2e-2	1.8 dB	4 dB

Figura 66: OSNR per a un BER de 2e-2

10.4 Impacte de la dispersió del mode de polarització en enllaços de llarga distància

L'objectiu de la següent prova és quantificar l'impacte del PMD en el rendiment d'un enllaç òptic de llarga distància i d'alta velocitat. Com he esmentat en el punt 6.4, la dispersió del mode de polarització provoca una diferència de propagació en els camps electromagnètics i el pols de sortida de l'emissor arriba distorsionat al receptor. El valor típic d'aquesta dispersió és de 0,1 a 0,2 ps/km^{1/2}. En línies generals, el PMD es considera excessiu quant supera els 0,5 ps /km^{1/2}, però depenent de quin és el temps de bit aquest valor varia. Podem afirmar que per evitar un augment de penalització menor de 1 dB cal:

$$\text{penalització} < 1\text{dB} \Rightarrow \Delta t_{\max} \leq \frac{T_{\text{bit}}}{10} \quad (25)$$

Per exemple, els coeficients genèrics de PMD requerits per a evitar una penalització excessiva són:

Bit Rate (Gbits/s)	MaxPMD (ps)	Coefficient de PMD per 400 Km (ps/km ^{1/2})
2,5	40	<2,0
10	10	<0,5
40	2.5	<0,125

Figura 67: coeficients genèrics de PMD requerits per evitar una excessiva penalització

Per tot plegat, actualment s'intenta evitar valors de 0,06 ps/Km^{1/2} en sistemes de llarga distància. Per provar l'impacte d'un PMD excessiu en el nostre sistema, provarem l'enllaç de 122 Gb/seg amb un valor de PMD per sota i per sobre del valor recomanable que serà:

$$122\text{Gbps} \Rightarrow \Delta t_{\max} \leq \frac{T_{\text{bit}}}{10} \Rightarrow \Delta t_{\max} \leq 0,081 \frac{\text{ps}}{\text{Km}^{1/2}} \quad (26)$$

En la primera execució de la funció *ofdmSimulator6P()* definirem el valor de PMD en $0.02e-12/(10^{(3/2)})$ que és un valor que està dins del marge recomanable. En la segona execució de la funció definirem el valor de PMD en $0.2e-12/(10^{(3/2)})$ que està per sobre del valor recomanable. En tots dos casos s'utilitzarà la modulació 6P-QPSK. Els valors d'entrada de la prova *Simulacio9.m* seran els següents:

```

OFDMParameters.symbolosOFDM = 40;
OFDMParameters.subcarriers = 256;
OFDMParameters.constelacion = '6P-QPSK';
OFDMParameters.NFFT = OFDMParameters.subcarriers * 8;
OFDMParameters.RSymb = 120e6; %SimbolRate (simb/seg)
OFDMParameters.RFfreqHz = 6e9; % Hz
OFDMParameters.PC = 1/16 ;
OFDMParameters.fixedp = 1;
OFDMParameters.SimParam.simulations = 1;
OFDMParameters.SimParam.BEROSNR = 20;
OFDMParameters.SimParam.BERFIBER = 0;
OFDMParameters.SimParam.ecualization = 1;
OFDMParameters.SimParam.fiber = 1;
OFDMParameters.SimParam.fixedp = 1;
laserParam.power = 2e-3;%W
laserParam.loPower = 2e-3;%W
fiberParam.fiberLength = 200;%Km
fiberParam.tramos = 1;
fiberParam.EDFA = 'no';
fiberParam.DCF = 'no';
fiberParam.waveLength = 1550e-9; % (m)
fiberParam.attenuation = 0.2; %e-3 dB/Km
fiberParam.dispersionChrom = 17; %s/mm^2
fiberParam.FN_dB = 5.5;
fiberParam.birr = 1e-7; % Birefringència
fiberParam.PMD = 0.02e-12 / (10^(3/2));
fiberParam.lc = 100;
Simulation50006P-QPSK256 = ofdm4DSimulator6P(OFDMParameters, laserParam, fiberParam);
fiberParam.PMD = 0.2e-12 / (10^(3/2)); % [s/m^0.5] PMD primer ordre
Simulation50006P-QPSK256PMD = ofdm4DSimulator6P(OFDMParameters, laserParam, fiberParam);

```

Els paràmetres de transmissió de la simulació de la prova de l'impacte de PMD són els següents:

Paràmetres de transmissió de la prova de PMD	6P-QPSK 256 sc. 200 Km PMD=0.02e-12/(10^(3/2))	6P-QPSK 256 sc. 200 Km PMD=0.1e-12/(10^(3/2))
Total de mostres transmeses incloent preàmbul del 4%	83968	83968
Total de mostres incloent percentatge de prefix cíclic a afegir	89216	89216
Nombre de mostres per cada símbol OFDM (64 sc)	2048	2048
Temps per mostra	4.0690e-12	4.0690e-12
Freqüència de mostreig	2.4576e+11	2.4576e+11
Gbits /seg.	122.8800	122.8800
Símbols/segon (Back-to-Back)	135e6	135e6
Signal Bandwidth	3.0841e+10	3.0841e+10

Figura 68: paràmetres de transmissió de *Simulacio9.m*

A continuació podem veure la constel·lació rebuda per totes dues simulacions realitzades en un enllaç de 200 km. Es pot comprovar l'augment del soroll que produeix un PMD excessiu:

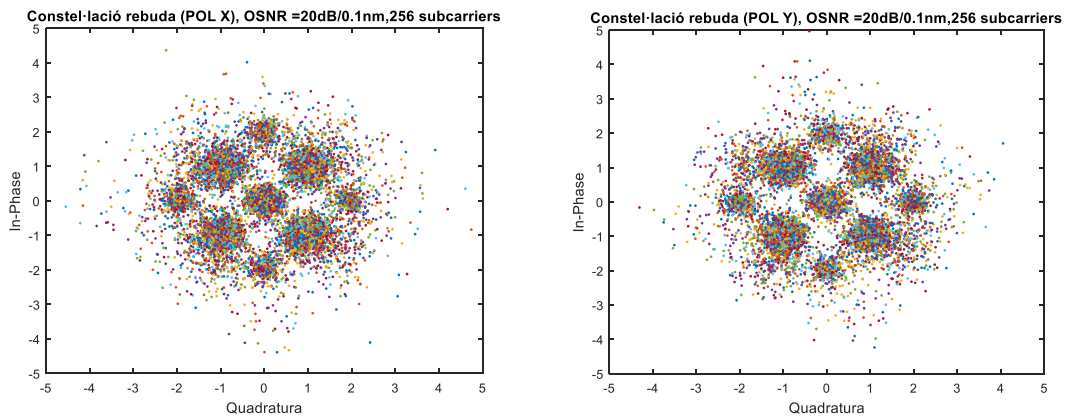


Figura 69: constel·lacions rebudes en Simulacio9.m i $PMD=0.02e-12/(10^{(3/2)})$

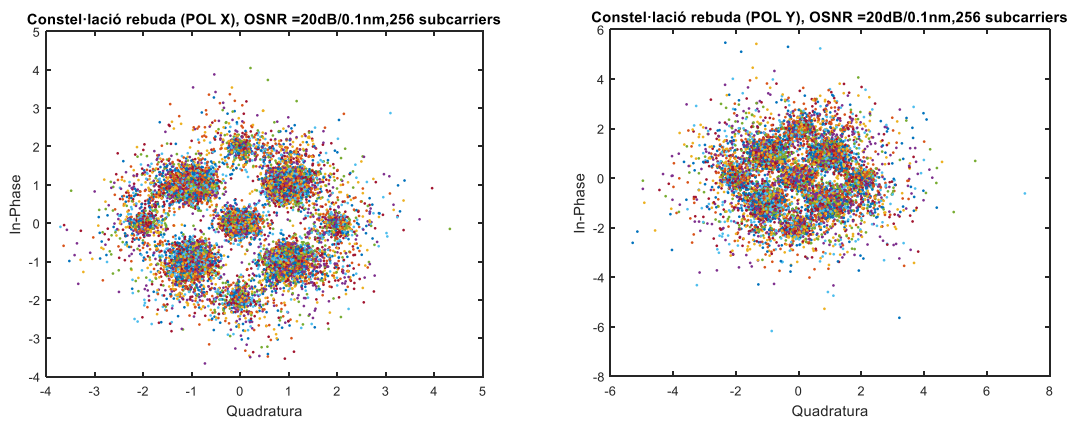


Figura 70: constel·lacions rebudes en Simulacio9.m i $PMD=0.1e-12/(10^{(3/2)})$

En la gràfica de BER/OSNR podem constatar com la diferència de BER entre els dos escenaris és apreciable i augmenta lleugerament a mesura que augmenta el OSNR:

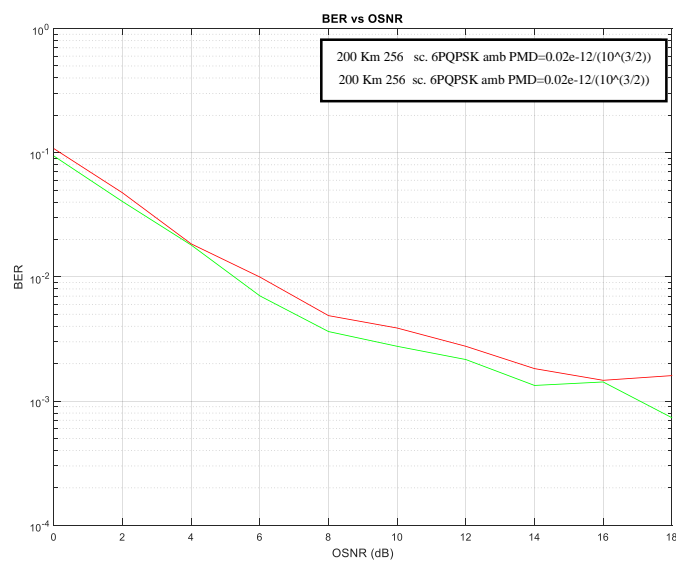


Figura 71: resultat de BER/OSNR per un enllaç de 200 Km, 256 sc OFDM i $PMD=0.02e-12/(10^{(3/2)})/0.1e-12/(10^{(3/2)})$

A continuació es mostren els resultats numèrics de BER/OSNR pels dos valors de PMD definits:

SNR	$PMD=0.02e-12/(10^{(3/2)})$	$PMD=0.2e-12/(10^{(3/2)})$
0	9.406161e-02	1.079282e-01
2	4.042023e-02	4.752048e-02
4	1.805110e-02	1.845175e-02
6	7.033476e-03	9.971510e-03
8	3.628027e-03	4.874466e-03
10	2.759972e-03	3.872863e-03
12	2.159010e-03	2.759972e-03
14	1.335470e-03	1.825142e-03
16	1.424501e-03	1.469017e-03
18	7.345085e-04	1.602564e-03

Figura 72: resultat numèric de BER/OSNR per un enllaç de 200 Km, 256 sc OFDM i $PMD=0.02e-12/(10^{(3/2)})/0.1e-12/(10^{(3/2)})$

Si comparem els dos escenaris tenim un OSNR per a un BER de $2e-2$ de:

BER	OSNR $PMD=0.02e-12/(10^{(3/2)})$	OSNR $PMD=0.2e-12/(10^{(3/2)})$
$2e-2$	3.6 dB	3.8 dB

Figura 73: OSNR per a un BER de $2e-2$

11 Conclusions

En aquest projecte he estudiat i analitzat les modulacions multinivell 6P-QPSK i PS-QPSK objecte d'estudi en l'article *Power-Efficient Modulation Formats in Coherent Transmission Systems* (Agrell, Karlsson, 2009). Mitjançant el programari Matlab, he modelitzat varis sistemes de transmissió òptics OFDM i implementat el mecanisme de mapatge que proposen els autors, quantificant les prestacions en diferents escenaris i comparant els resultats amb els obtinguts amb la modulació DP-QPSK, analitzada i implementada al treball final de Màster *Simulación de sistemas CO-OFDM 4D ópticos* (David Sacie Alcazar). He implementat els efectes de la dispersió del mode de polarització (PMD) en els enllaços òptics i he inclòs els efectes de la figura soroll en els amplificadors EDFA per lo que els resultats obtinguts en les diferents simulacions són força representatius.

Les diferents modelitzacions han confirmat que les prestacions de la modulació 6P-QPSK en termes de BER són lleugerament pitjors a la modulació DP-QPSK però amb una millor eficiència espectral ja que podem codificar 4,5 bits per símbol mentre que la modulació DP-QPSK codifica 4 bits per símbol. També hem comprovat com és possible implementar la modulació 6P-QPSK en enllaços de llarga distància (més de 5000 km amb trams de 100 Km) i d'alta velocitat (més de 255 Gb/s). Les modelitzacions també han demostrat que la modulació PS-QPSK (C_2), malgrat tenir una pitjor eficiència espectral (3 bits per símbol) i un pitjor balanç energètic respecte a les altres dues modulacions, és més robust respecte als errors de bit i permet constituir enllaços amb un major abast.

Altrament, la codificació proposada en l'article on una paraula de 9 bits es codifica en dues modulacions consecutives implica una necessitat de memòria afegit. La complexitat de la codificació i de la desmodulació, que obliga a calcular la distància euclidiana entre els vectors de la constel·lació propers i el punt rebut, fa que el cost computacional de la implementació augmenti en $O(n^2)$ respecte a la

modulació DP-QPSK, on n són els punts de la constel·lació C_3 . En velocitats de transmissió de dades de l'entorn dels 200 Gb/s, aquests requeriments de memòria i de maquinari afegits fan que la proposta de modulació 6P-QPSK proposada en l'article pugui ser incompatible amb la necessitat d'implementar sistemes barats i relativament senzills.

En aquest sentit, la modulació 6P-QPSK-32 proposada per Chen Chen, Chuandong Li, Mahdi Zamani and Zhuhong Zhang en l'article *Coherent Detection of a 32-Point 6PolSK-QPSK Modulation* sembla molt més raonable. Per una part, la nova variable $PolM$ que es proposa en la modulació 6P-QPSK per rotar la polarització 45° s'utilitza també en la modulació 6P-QPSK-32 amb el mateix propòsit, però girant tota la constel·lació C_1 aconseguint 32 punts possibles de mapatge, per lo que podem codificar 5 bits per símbol. El mapatge és molt més senzill, ja que permet utilitzar un sol bit per a rotar (o no) la constel·lació i s'aconsegueix una codificació directa bits \rightarrow símbol en una sola modulació, fet que simplifica el procés de mapping i, malgrat que manté la complexitat en la desmodulació respecte a la proposta 6P-QPSK, requereix menys recursos computacionals i de memòria augmentant encara més l'eficiència espectral respecte a les modulacions 6P-QPSK-24, DP-QPSK i PS-QPSK.

Altrament, hem constatat com aquest tipus de sistemes són força sensibles a la dispersió del mode de polarització (PMD) i variacions de $0.18e-12/(10^{(3/2)})$ en un sistema de 6P-QPSK a 122.8 Gbit/segon pot empitjorar significativament el BER.

12 Propostes d'investigació

A continuació plantejo varies línies d'investigació a realitzar en futurs projectes o estudis:

- Modificar les funcions implementades en Matlab, particularment el modulador IQ, definint una nova variable $PolM$ que rotarà l'estat de polarització del vector X o Y aconseguint els nous 8 vectors del conjunt C_2 . Malgrat que les prestacions de les modulacions 6P-QPSK i PS-QPSK no variaran significativament respecte als resultats obtinguts en aquest projecte, el resultat serà una modelització més real d'un sistema de transmissió òptic coherent amb la modulació 6P-QPSK i PS-QPSK.
- Una vegada implementat el primer punt, implementar la nova proposta de modulació realitzada en l'article *Coherent Detection of a 32-Point 6PolSK-QPSK Modulation* dels autors Chen Chen Chuandong Li, Mahdi Zamani i Zhuhong i comparar les seves prestacions respecte a les demés modulacions 4D, quantificant d'una manera més acurada els resultats de les diferents modelitzacions 4D a nivell de BER, E_b/N_o i sensibilitat afegint-hi els costos computacionals i de memòria de les diferents propostes, ja que considero el BER com un element de mesura no lo suficientment representatiu a l'hora de comparar modulacions multinivell.

- Un altre camp d'investigació pot ser partir de la implementació PS-QPSK i de la variable PolM, rotar la polarització de la constel·lació 45° per aconseguir implementar un sistema PM-QPSK, lo que aconseguirem un 25 % més de *bit rate* malgrat perdre 3 dB de sensibilitat.

13 Glossari

6P-QPSK: 6 polarization Quaternary- Phase Shift Keying

ADC: Conversor analògic/Digital.

APD: Fotodiodo.

AWGN: Additive white Gaussian noise

BER: Tassa de Error de bit.

B-PSK: Binary-Phase Shift Keying.

CO-OFDM: Coherent Orthogonal Frequency Division Modulation.

CP: Prefix cíclic.

CPE: Common phase estimation.

DAC: Conversor digital/analògic.

dB: Decibels.

DD: Detecció directa.

DFT: Discrete Fourier Transform.

DP-QPSK: Dual polarization Quaternary- Phase Shift Keying.

DSP: Digital Signal Processor.

EDFA: Erbium doped fibre amplifier.

FMD: Frequency Division Multiplexing.

FFT: Fast Fourier Transform.

F_s: Sampling Frequency.

ICI: Inter Carrier Interference.

IQ: In phase and Quadrature.

IDFT: Inverse Discrete Fourier Transform.

IFFT: Inverse Fast Fourier Transform.

ISI: Inter Symbol Interference.

LO: Local oscillator.

MATLAB: MATrix LABoratory.

MZM: Mach–Zehnder Modulator

Nfft: Nombre de mostres per símbol OFDM.

OFDM: Orthogonal Frequency Division Multiplexing.

OSNR: Optcial signal to noise ratio.

PAPR:Peak-toAverage Power Ratio.

PMD: Polarition Mode Dispertion.

QAM: Quadrature Amplitude Modulation.

Q-PSK: Quaternary- Phase Shift Keying.

RF: Radio Frecuency.

Rx: Receptor.

SER: Symbol Error Rate.

SE: Spectral efficiency.

SMF: Single Mode Fiber.

MMF: Multi Mode Fiber.

SSMF: Standar Single Mode Fiber.

SNR: Signal-to-noise ratio.

Tx: Transmissor.

WDM:Wavelength division multiplexing.

ZP: Zero Padding.

14 Bibliografia

1. Sander L. Jansen: **SC341 OFDM for Optical Communications**. Nokia Siemens Networks, Germany. 2010
2. Erik Agrell and Magnus Karlsson: **Power-Efficient Modulation Formats in Coherent Transmission Systems**. IEEE- JOURNAL OF LIGHTWAVE TECHNOLOGY, VOL. 27, NO. 22, 2009.
3. H. Bülow, “**Polarization QAM modulation (POL-QAM) for coherent detection schemes**” in Proc. Opt. Fiber Commun. Nat. Fiber Opt. Eng. Conf. (OFC/NFOEC), 2009, Paper OWG2.
4. Chen Chen, Chuandong Li, Mahdi Zamani and Zhuhong Zhang: **Coherent Detection of a 32-Point 6PolSK-QPSK Modulation Format** OFC/NFOEC Technical Digest 2013.
5. J. H. Conway and N. J. A. Sloane, **Sphere Packings**, Lattices and Groups, 3rd ed. New York: Springer-Verlag, 1999.
6. Johannes Karl Fischer, Member, IEEE, Saleem Alreesh, Student Member, IEEE, Robert Elschner, Member, IEEE, Felix Frey, Markus Nölle, Carsten Schmidt-Langhorst, and Colja Schubert: **Bandwidth-Variable Transceivers based on Four-Dimensional Modulation Formats**, JOURNAL OF LIGHTWAVE TECHNOLOGY, vol. 32, n°. 16, agost, 2014
7. David Sacie Alcázar: **Simulación de sistemas CO-OFDM 4D ópticos**, TFM, UOC.
8. Yue-Kai Huang, Ming-Fang Huang, Ezra Ip, Eduardo Mateo and Philip N. Ji: **High-Capacity Fiber Field Trial Using Terabit/s All-Optical OFDM Superchannels With DP-QPSK and DP-8QAM/DP-QPSK Modulation**
9. Corneliu Eugen D. Sterian · Yuanyuan Ma · Matthias Pätzold · Ion Banica · Huaqiang He, **New super-orthogonal space-time trellis codes using differential M-PSK for non coherent Mobile communication systems with two transmit antennas**, Institut Télécom and Springer-Verlag 2010
10. G. Taricco, E. Biglieri, and V. Castellani, “**Applicability of four-dimensional modulations to digital satellites: A simulation study**,” in Proc. IEEE Global Telecommun. Conf. (GLOBECOM), Nov./Dec. 1993, vol. 4, pp. 28–34.
11. S. Benedetto and E. Biglieri, **Principles of Digital Transmission: With Wireless Applications**. Norwell, MA: Kluwer, 1999.

12. M. Simon, S. Hinedi, and W. Lindsey, **Digital Communication Techniques:Signal Design and Detection**. Englewood Cliffs, NJ: Prentice-Hall, 1995
13. Ivan Djordjevic, Hussam G. Batshon, Lei Xu and Ting Wang: **Four-dimensional optical multiband-OFDM for beyond 1.4 Tb/s serial optical transmission** (Ivan Djordjevic). OSA. 2011
14. Kim Roberts, Member, IEEE, Maurice O'Sullivan, and Kuang-Tsan: **Performance of Dual-Polarization QPSK for Optical Transport Systems**. JOURNAL OF LIGHTWAVE TECHNOLOGY. 2009.
15. K. Petermann , **Laser diode modulation and noise** , Kluwer Academic.
16. Ivan Djordjevic: **Four-Dimensional Coded Optical OFDM for Ultra-High-Speed Metro Networks**. 2013
17. J. Karaki, E. Pincemin, D. Grot, T.Guillossou, and Y. Jaouen **Dual-Polarization Multi-Band OFDM versus Single-Carrier DP-QPSK for 100 Gb/seg Long-Haul WDM Transmission over Legacy Infrastructure**. 2012
18. Ariel Leiva L., Marco Tarifeño G., Ricardo Olivares V. **Effects of Polarization Mode Dispersion (PMD) on propagation of optical pulses in optical fibres**

15 Annexes

```

function result = Ofdm4DSimulator6P (ofdmParam,laserParam,fiberParam)
%
%*****Ofdm4DSimulator6P*****
%
% Parametres d'entrada:
%
% ofdmParam: Paràmetres de senyal OFDM
%
%     simbolosOFDM: símbols a transmetre
%     subcarriers: Subportadores en cada símbol
%     constelacion: Modulació digital seleccionada. Valors acceptats: '6P-QPSK'
%     NFFT: mostres transmeses en cada FFT
%     RSymb: Tassa de símbols
%     RFFreqHz: Freqüència de portadora
%     PC: Percentatge de prefix cíclic a insertar, en tant per uno. 1/16
%     (6.25%) per defecte
%     upsamplingFactor: Factor de sobremostreig
%
% ofdmParam.SimParam: Paràmetres de simulació
%
%     simulations: Numnre de simulacions. Per defecte, 1
%     BEROSNR: >0 activa el càlcul de gràfica, en el rang [-BEROSNR,BEROSNR]
%     Spectrum: A 1, activa la representació del espectre
%     Equalizació: a 1 activa la etapa equalitzadora, a 0 la deixa desactivada
%     Fiber: a 1 activa la etapa de fibra Òptica SSMF
%
% laserParam:
%
%     LaserPower: potència de sortida del làser [W]
%     LoPower: potència de sortida del oscil·lador en recepció [W]
%
% fiberParam:
%
%     fiberLength: longitud de la fibra
%     tramos: Nombre de trams SSMF
%     DCF: a 1 activa el tram de fibra compensadora de dispersió
%     wavelength: Longitud d'onda central (m)
%     attenuation: atenuació de la fibra
%     dispersionChrom: Paràmetre de dispersió de la fibra (s/m^2)
%
%*****

%=====
% Càlcul de paràmetres OFDM secundaris
%=====
disp(strcat('Codificant senyal OFDM (6P-QPSK)...',int2str(ofdmParam.subcarriers),'
subportadores'));
SamplesCh=ofdmParam.NFFT; %nombre de mostres per cada símbol OFDM
disp(strcat('Nombre de mostres per cada símbol OFDM:',int2str(SamplesCh)));
N = (ofdmParam.simbolosOFDM+floor(ofdmParam.simbolosOFDM/25)) * SamplesCh; %Total de mostres
transmeses incloent preàmbul 4%
disp('Total de mostres transmeses incloent preàmbul 4%:');
disp(N);
N = N*(1+ofdmParam.PC); %Total de mostres incloent percentatge de prefix cíclic a afegir
disp('Total de mostres incloent percentatge de prefix cíclic a afegir: ');
disp(N);
dt = 1 / ( ofdmParam.RSymb*SamplesCh );%Temps per mostra
disp('Temps per mostra: ');
disp(dt );
df =1/dt; %Freqüència de mostreig
disp('Freqüència de mostreig: ');
disp(df );
result.Rb = df*( ofdmParam.subcarriers /ofdmParam.NFFT) *4*1e-9; %Gbits/seg
disp('Gbits/seg: ');
disp(result.Rb );
time = ( 0 : ( SamplesCh-1) ) *dt ; %Finestra de temps
timepc = ( 0 : ( SamplesCh-1)*(1+ofdmParam.PC) ) *dt ; %Finestra de temps amb prefix cíclic
FREQ=(-(N/ 2) : (N/2-1) ) * 1 / ( dt*N); %Eix de freqüència
disp('length Eix de freqüència: ');
disp(length(FREQ));
Signal_Bandwidth = ( 2 /max ( timepc ) ) + ( ( ofdmParam.subcarriers-1) /max ( time ) ); %
disp('Signal_Bandwidth : ');
disp(Signal_Bandwidth );
fiberParam.Bo = Signal_Bandwidth;
%=====
% Freqüència de mostreig
%=====
t_data=rand(1,ofdmParam.subcarriers*4.5*(ofdmParam.simbolosOFDM-
1))>0.5;%1,ofdmParam.subcarriers*4.5*(ofdmParam.simbolosOFDM+1

```

```

t_data_Polar = 2*t_data-1;
result.polarBinData = t_data_Polar;
dataCounter=1;%Permetrà seleccionar dades per cada símbol OFDM
=====
% Generació senyal OFDM (PROCESAT DIGITAL)
=====
result.spdata = [];
for d=1:ofdmParam.simbolosOFDM-1
    data = t_data_Polar(dataCounter:dataCounter+ofdmParam.subcarriers*4.5-1); %Dades per trama
    dataCounter=dataCounter+(ofdmParam.subcarriers*4.5);

    =====
    % Conversió S/P
    =====
    s_p_data = reshape(data,9,length(data)/9);
    result.spdata = [result.spdata s_p_data];

    =====
    % Mapping 4D (6P-QPSK)
    =====

dataPolX=[];
dataPolY=[];
for jj=1: (length(data)/9)
    if(s_p_data(9,jj)==-1)
        Xj1 = complex(-s_p_data(2,jj) , -s_p_data(1,jj));
        Xj2 = complex(-s_p_data(4,jj) , -s_p_data(3,jj));
        Xj3 = complex(-s_p_data(6,jj) , -s_p_data(5,jj));
        Xj4 = complex(-s_p_data(8,jj) , -s_p_data(7,jj));
    end;
    if(s_p_data(9,jj)==1 && s_p_data(8,jj)==-1)
        Xj1 = complex(-s_p_data(2,jj) , -s_p_data(1,jj));
        Xj2 = complex(-s_p_data(4,jj) , -s_p_data(3,jj));

    if(s_p_data(7,jj)==-1) && (s_p_data(6,jj)==-1) && (s_p_data(5,jj)==-1)
        Xj4 = 0+0i;
        Xj3 = 0+2i;
    end;
    if(s_p_data(7,jj)==-1) && (s_p_data(6,jj)==-1) && (s_p_data(5,jj)==1)
        Xj4 = 0+0i;
        Xj3 = 2+0i;
    end;
    if(s_p_data(7,jj)==-1) && (s_p_data(6,jj)==1) && (s_p_data(5,jj)==-1)
        Xj4 = 0+2i;
        Xj3 = 0+0i;
    end;
    if(s_p_data(7,jj)==1) && (s_p_data(6,jj)==-1) && (s_p_data(5,jj)==-1)
        Xj4 = -2+0i;
        Xj3 = 0+0i;
    end;
    if(s_p_data(7,jj)==1) && (s_p_data(6,jj)==1) && (s_p_data(5,jj)==1)
        Xj4 = 0+0i;
        Xj3 = 0-2i;
    end;
    if(s_p_data(7,jj)==1) && (s_p_data(6,jj)==1) && (s_p_data(5,jj)==-1)
        Xj4 = 0+0i;
        Xj3 = -2+0i;
    end;
    if(s_p_data(7,jj)==1) && (s_p_data(6,jj)==-1) && (s_p_data(5,jj)==1)
        Xj4 = 0-2i;
        Xj3 = 0+0i;
    end;
    if(s_p_data(7,jj)==-1) && (s_p_data(6,jj)==1) && (s_p_data(5,jj)==1)
        Xj4 = 2+0i;
        Xj3 = 0+0i;
    end;

end;
if(s_p_data(9,jj)==1 && s_p_data(8,jj)==1)

    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==-1)
        Xj2 = 0+0i;
        Xj1 = 0+2i;
    end;
    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==1)
        Xj2 = 0+0i;
        Xj1 = +2+0i;
    end;
    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==1) && (s_p_data(1,jj)==-1)
        Xj2 = 0+2i;
        Xj1 = 0+0i;
    end;
    if(s_p_data(3,jj)==1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==-1)
        Xj2 = -2+0i;

```

```

Xj1 =0+0i;
end;
if (s_p_data(3,jj)==1) && (s_p_data(2,jj)==1) && (s_p_data(1,jj)==1)
Xj2 = 0+0i;
Xj1 = 0-2i;
end;
if (s_p_data(3,jj)==1) && (s_p_data(2,jj)==1) && (s_p_data(1,jj)==-1)
Xj2 = 0+0i;
Xj1 = -2+0i;
end;
if (s_p_data(3,jj)==1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==1)
Xj2 = 0-2i;
Xj1 = 0+0i;
end;
if (s_p_data(3,jj)==-1) && (s_p_data(2,jj)==1) && (s_p_data(1,jj)==1)
Xj2 = 2+0i;
Xj1 = 0+0i;
end;

Xj3 = complex(-s_p_data(5,jj) , -s_p_data(4,jj));
Xj4 = complex(-s_p_data(7,jj) , -s_p_data(6,jj));
end
dataPolY = [dataPolY Xj3];
dataPolX = [dataPolX Xj4];
dataPolY = [dataPolY Xj1];
dataPolX = [dataPolX Xj2];

end
if (d == 1)
MPolX = [];
MPolY = [];
end
MPolX = [MPolX transpose([dataPolX])];
MPolY = [MPolY transpose([dataPolY])];
end

%=====
% Normalització
%=====
NormFactor = 1 /sqrt( mean( mean( abs( MPolY ) .^ 2 ) ) );
MPolX = MPolX * NormFactor;
MPolY = MPolY * NormFactor;
%=====
% Símbol d'entrenament (Tipus Bloc).Suposem esvaïment lent del canal
%=====
ST = ( exp(1i* pi .* ( ( 1 : ofdmParam.subcarriers ) .^ 2 ) / (ofdmParam.subcarriers) ) ).';
MpolX_withST = [];
MpolY_withST = [];
MPolX = [ST MPolX];
MPolY = [ST MPolY];
for ii=1:floor((ofdmParam.simbolosOFDM/25))
MpolX_withST = [MpolX_withST MPolX(:, ((ii-1)*25+1):(ii*25)) ST];
MpolY_withST = [MpolY_withST MPolY(:, ((ii-1)*25+1):(ii*25)) ST];
end

if (not(floor(ofdmParam.simbolosOFDM/25) == ( ofdmParam.simbolosOFDM/25)))
MpolX_withST = [MpolX_withST MPolX(:,floor(ofdmParam.simbolosOFDM/25)*25+1:end)];
MpolY_withST = [MpolY_withST MPolY(:,floor(ofdmParam.simbolosOFDM/25)*25+1:end)];
end;
%=====
% ZeroPadding Simetric
%=====

result.Zeropadding =SamplesCh-(ofdmParam.subcarriers);
result.OFDMSignalf_tx.PolX = [ MpolX_withST( ((ofdmParam.subcarriers)/2+1) :
(ofdmParam.subcarriers), :);...
zeros(result.Zeropadding, (ofdmParam.simbolosOFDM)+floor(ofdmParam.simbolosOFDM/25));...
MpolX_withST( 1 : (ofdmParam.subcarriers/2), :)];
result.OFDMSignalf_tx.PolY = [ MpolY_withST( (ofdmParam.subcarriers)/2+1) :
ofdmParam.subcarriers, :);...
zeros(result.Zeropadding,
ofdmParam.simbolosOFDM+floor(ofdmParam.simbolosOFDM/25));...
MpolY_withST( 1 : ofdmParam.subcarriers/2, :)];

%=====
% IFFT2: Càlcul de suportadores en el domini temporal, en dues dimensions
% Garantim ortogonalitat entre ambdues polaritzacions
%=====

OFDMSignal_t_tx =
ifft2OFDM(result.OFDMSignalf_tx,ofdmParam.simbolosOFDM+floor(ofdmParam.simbolosOFDM/25));

%=====

```

```

% PREFIX CÍCLIC
%=====
NumPC = SamplesCh*ofdmParam.PC;
OfdmSignal_t_PolX = [ OFDMSignalt_tx.PolX( ( SamplesCh-NumPC+1) : SamplesCh, : ) ;
OFDMSignalt_tx.PolX ] ;
OfdmSignal_t_PolY = [ OFDMSignalt_tx.PolY( ( SamplesCh-NumPC+1) : SamplesCh, : ) ;
OFDMSignalt_tx.PolY ] ;
result.OFDMSignalt_tx.PolX = reshape ( OfdmSignal_t_PolX ,N, 1 ) ;
result.OFDMSignalt_tx.PolY = reshape ( OfdmSignal_t_PolY ,N, 1 ) ;

%=====
% ESPECTRE DE LA SENYAL TRANSMESA
%=====
if (isfield(ofdmParam.SimParam,'Spectrum') && ofdmParam.SimParam.Spectrum == 1)
[Pxx,W] = pwelch(result.OFDMSignalt_tx.PolX,[],[],length(FREQ),1);
[Pxy,W] = pwelch(result.OFDMSignalt_tx.PolY,[],[],length(FREQ),1);
figure(1);
plot(FREQ/max(FREQ)*Signal_Bandwidth,10*log10(fftshift(Pxx)),'r');
hold on;
plot(FREQ/max(FREQ)*Signal_Bandwidth,10*log10(fftshift(Pxy)),'b');
xlabel('Freqüència normalitzada, Hz')
ylabel('Densitat espectral de potència')
title('Espectre OFDM transmes (Ambdues polaritzacions)');
end;
%=====
% Upsampling (Interpolació)
%=====
if (isfield(ofdmParam,'UpsamplingFactor') && ofdmParam.UpsamplingFactor > 0)
upsampledt.PolX = interp(result.OFDMSignalt_tx.Pol,ofdmParam.UpsamplingFactor);
upsampledt.PolY = interp(result.OFDMSignalt_tx.PolY,ofdmParam.UpsamplingFactor);
result.df = df*ofdmParam.UpsamplingFactor; %Nueva frecuencia de muestreo con oversampling
result.dt = 1/df;
else
upsampledt.PolX = result.OFDMSignalt_tx.PolX;
upsampledt.PolY = result.OFDMSignalt_tx.PolY;
end
%=====
% Conversió RF
%=====
result.OFDMSignalRF_tx.PolX = IQRF (upsampledt.PolX,ofdmParam.RFfreqHz,dt);
result.OFDMSignalRF_tx.PolY = IQRF (upsampledt.PolY,ofdmParam.RFfreqHz,dt);

%=====
% Modulació Òptica: CONF LASER + 2 IQ-Mach-Zehnder niats (DOBLE POL)
%=====
result.EOpt_Tx = genOptField(result.OFDMSignalRF_tx,ofdmParam,laserParam,dt);

if (isfield(ofdmParam.SimParam,'fixedp') && ofdmParam.SimParam.fixedp == 1)
power_resultEOpt_Tx1=mean(abs(result.EOpt_Tx(1,:)).^2)/1e-3;
power_resultEOpt_Tx2=mean(abs(result.EOpt_Tx(2,:)).^2)/1e-3;
optical_power_output_mW = laserParam.power;
factorPower1 = sqrt(optical_power_output_mW/power_resultEOpt_Tx1);
factorPower2 = sqrt(optical_power_output_mW/power_resultEOpt_Tx2);
result.EOpt_Tx(1,:) = result.EOpt_Tx(1,,:)*factorPower1;
result.EOpt_Tx(2,:) = result.EOpt_Tx(2,,:)*factorPower2;
end;
%=====
% Propagació per fibra SSMF
%=====
if (isfield(ofdmParam.SimParam,'fiber') && ofdmParam.SimParam.fiber == 1)
fiberParam.dt = dt;
result.EOpt_Rx = fiberTx(result,fiberParam,ofdmParam);
ofdmParam.longitudFibraKm = fiberParam.fiberLength;
else
result.EOpt_Rx = result.EOpt_Tx;
end;

%=====
% RX COHERENT OFDM: OSC. LOCAL PBS,PBF,PDs BALANCEJATS,MIXER RF COHERENT
%=====

laserParam.dt = dt;
laserParam.RFfreqHz = ofdmParam.RFfreqHz;
ofdmParam.df = df;
ofdmParam.SamplesCh = SamplesCh;
ofdmParam.ZeroPadding = result.ZeroPadding;
ofdmParam.NormFactor = NormFactor;
ofdmParam.FREQ = FREQ;
ofdmParam.Signal_Bandwidth = Signal_Bandwidth;
titleOSNR = '';
if (isfield(ofdmParam.SimParam,'BEROSNR') && ofdmParam.SimParam.BEROSNR>0 )

```

```

%=====
% PROCESAT DIGITAL OFDM Rx + ESTUDI BER/OSNR
%=====
if (isfield(ofdmParam.SimParam,'simulations'))
    simulations = ofdmParam.SimParam.simulations;
else
    simulations = 1;
end
if (ofdmParam.SimParam.BEROSNR >15)
    OSNRmin = 0 ;
    OSNRmax = ofdmParam.SimParam.BEROSNR;
else
    OSNRmin = -ofdmParam.SimParam.BEROSNR;
    OSNRmax = ofdmParam.SimParam.BEROSNR;
end
total_it = ((OSNRmax - OSNRmin) / 2) * simulations;
s=sprintf('Starting OSNR measurement %d simulations, range %d dB<=OSNR<= %d
dB:',simulations,OSNRmin,OSNRmax);
disp(s);
tinit=tic; % needed for displaying progress
for sim = 1:simulations
    i_snr = 1;
    BWOSNR = 3e8 / (0.1e-9);
    for osnr = OSNRmin:2:OSNRmax
        EOpt_Rx(1,:) = OSNR(result.EOpt_Rx(1,:), osnr, BWOSNR, df);

        EOpt_Rx(2,:) = OSNR(result.EOpt_Rx(2,:), osnr, BWOSNR, df);
        result.ElectricFields_Rx = rxOptCoherenteOFDM(EOpt_Rx, laserParam);
        ofdmParam.OSNR = osnr;
        rx_data = demodulatorDigOFDM6P(result.ElectricFields_Rx, ofdmParam);
        %Buffer OSNR/BER
        result.rx_data = rx_data;
        [number(i_snr), BER(sim, i_snr)] = biterr(t_data, rx_data.bindata);

        % Display progress
        if toc(tinit)>5
            s=sprintf('Measuring OSNR: %0.2f%% complete.', 100*((sim-1) * ((OSNRmax -
OSNRmin)/2) + i_snr-1)/total_it));
            disp(s);
            tinit=tic;
        end;
        i_snr=i_snr+1;
    end;
end;

%=====
% GRAFICA BER/OSNR
%=====

% Calculamos promedio sobre todas las simulaciones (ofdmParam.SimParam.simulations)
for col=1:(OSNRmax - OSNRmin)/2;
    ber(1,col)=0;
    for row = 1:ofdmParam.SimParam.simulations;
        ber(1,col)=ber(1,col)+BER(row,col);
    end;
end

ber=ber./ofdmParam.SimParam.simulations;
figure;
ii=OSNRmin:2:OSNRmax-2;
result.ii = ii;
result.ber = ber;
for aup=1:length(result.ii)
    s=sprintf('SNR %0.1f: ber %d ', result.ii(aup), result.ber(aup));
    disp(s);
end;
semilogy(ii,ber);
title('BER vs OSNR');
ylabel('BER');
xlabel('OSNR (dB/0.1nm)');
grid on
titleOSNR = strcat(' OSNR = ',int2str(ofdmParam.SimParam.BEROSNR), 'dB/0.1nm,
',int2str(ofdmParam.subcarriers), ' subcarriers');
else
    result.ElectricFields_Rx = rxOptCoherenteOFDM(result.EOpt_Rx, laserParam);
%=====
% Processat digital de la senyal en Rx.
%=====

rx_data = demodulatorDigOFDM6P(result.ElectricFields_Rx, ofdmParam);
result.rx_data = rx_data;
result.tdata = t_data;
[i_snr, result.BerBER] = biterr(t_data, rx_data.bindata);
end

```

```

%=====
% Representació de la constel·lació rebuda
%=====

figure;
plot ( real ( rx_data.ODataPolX_Est), imag ( rx_data.ODataPolX_Est ) , ' . ' );
title(strcat('Constel·lació rebuda (POL X)',titleOSNR));
ylabel('In-Phase');
xlabel('Quadratura');
figure;
plot ( real ( rx_data.ODataPolY_Est), imag ( rx_data.ODataPolY_Est ) , ' . ' );
title(strcat('Constel·lació rebuda (POL Y)',titleOSNR));
ylabel('In-Phase');
xlabel('Quadratura');

%=====
% PROCESAT BERvsFiberLength
%=====
if (isfield(ofdmParam.SimParam,'fiber') && ofdmParam.SimParam.fiber == 1)
    if (isfield(ofdmParam.SimParam,'BERFIBER') && ofdmParam.SimParam.BERFIBER == 1)
        lengthf = fiberParam.fiberLength;
        itt = 1;
        for ii=50:25:lengthf
            fiberParam.fiberLength = ii;
            fiberln(itt) = fiberParam.fiberLength;
            result.EOpt_Rx = fiberTx(result,fiberParam,ofdmParam);
            result.ElectricFields_Rx = rxOptCoherenteOFDM(result.EOpt_Rx,laserParam);
            rx_data = demodulatorDigOFDM6P(result.ElectricFields_Rx,ofdmParam);
            result.rx_data = rx_data;
            result.tdata = t_data;
            [number(itt),BER(itt)] = biterr(t_data,rx_data.bindata);
            itt=itt+1;

            end
            for aup=1:itt-1
                s=sprintf('Distància %0.1f: BER %d ', fiberln(aup),BER(aup));
                disp(s);
            end;
        %=====
        % GRÀFICA BER/OSNR
        %=====
        figure;
        result.BER = BER;
        semilogy(fiberln,BER);
        title('BER vs Fiber');
        ylabel('BER');
        xlabel('Length(Km)');
        grid on
        end;
    end;

end

%=====
% Funció: ifft2OFDM: Càlcul de subportadores en el domini temporal per n símbols OFDM-2D
%=====
function result = ifft2OFDM(signalf,frames)
    result.PolX= [];
    result.PolY= [];
    for ii =1:frames
        M2xSamplesCh = [signalf.PolX(:,ii) signalf.PolY(:,ii) ];
        M2xSamplesCh_t = ifft2(M2xSamplesCh);
        result.PolX = [result.PolX M2xSamplesCh_t(:,1)];
        result.PolY = [result.PolY M2xSamplesCh_t(:,2)];
    end

end

%=====
% Funció: IQRf(signal)
%=====
function iqresult = IQRf (signal,RFfreqHz,dt)
    I = real ( signal );
    Q = imag ( signal );
    iqresult.VI = I;% .* cos( ang' );
    iqresult.VQ = Q;% .* sin( ang' );

end

function rx_data = demodulatorDigOFDM(Exr , ofdmParam)
%=====
% demodulatorDig6POFDM()
% Paràmetres d'entrada:

```

```

% Erx: Vectors amb senyal elèctrica detectada (ambdues polaritzacions)
% ofdmParam: paràmetre de senyal OFDM
%
% DOWNSAMPLE + ESPECTRO + REMOVE CP + FFT2 + REMOVE ZEROPADDING + ECUALIZATION + SLICING + DEMAPPING
%
%=====
%
%=====
% Downsample
%=====

if (isfield(ofdmParam,'UpsamplingFactor') && ofdmParam.UpsamplingFactor > 0)
    EoutxDownSamp = decimate(Erx(:,1),ofdmParam.UpsamplingFactor,8,'fir');
    EoutyDownSamp = decimate(Erx(:,2),ofdmParam.UpsamplingFactor,8,'fir');
else
    EoutxDownSamp = Erx(:,1);
    EoutyDownSamp = Erx(:,2);
end

%=====
% Espectre senyal Rx
%=====
if (isfield(ofdmParam.SimParam,'Spectrum') && ofdmParam.SimParam.Spectrum == 1)

    if (not( isfield (ofdmParam,'OSNR') ) || ofdmParam.OSNR >= 0 )
        figure(5);
        [Pxx,W] = pwelch(EoutxDownSamp,[],[],length(ofdmParam.FREQ),1);

plot(ofdmParam.FREQ/max(ofdmParam.FREQ)*ofdmParam.Signal_Bandwidth,10*log10(fftshift(Pxx)), 'r');
        hold on;
        [Pxx,W] = pwelch(EoutyDownSamp,[],[],length(ofdmParam.FREQ),1);

plot(ofdmParam.FREQ/max(ofdmParam.FREQ)*ofdmParam.Signal_Bandwidth,10*log10(fftshift(Pxx)));
        xlabel('Frequència normalitzada, Hz')
        ylabel('Densitat espectral de potència')
        titleOsnr = '';
        titleFiber = '';
        if (isfield(ofdmParam.SimParam,'BEROSNR') && ofdmParam.SimParam.BEROSNR>0 )
            titleOsnr = strcat(' OSNR = ',int2str(ofdmParam.OSNR), 'dB');
        end
        if (isfield(ofdmParam,'fiberLength'))
            titleFiber = strcat(' Fibra SSMF: ',int2str(ofdmParam.fiberLength),' Kms');
        end
        title(strcat('Espectre rebut OFDM (ambdues polaritzacions)',titleOsnr,titleFiber));
        hold off;
    end;
end;

%=====
% Eliminem prefix cíclic
%=====
PC = ofdmParam.PC;
NumPC = ofdmParam.SamplesCh*PC;
Frames = ofdmParam.simbolosOFDM+floor(ofdmParam.simbolosOFDM/25);
ChsCP=ofdmParam.SamplesCh*(1+PC);
Esampx= reshape ( EoutxDownSamp , ChsCP , Frames );
Esampy= reshape ( EoutyDownSamp , ChsCP , Frames );
Esampx = Esampx ( ( NumPC+1 ) : ( ChsCP ) , : );
Esampy = Esampy ( ( NumPC+1 ) : ( ChsCP ) , : );

%=====
% FFT2
%=====
ODataPolX= [];
ODataPolY= [];
for ii =1:Frames
    M2xSamplesCh_t = [Esampx(:,ii) Esampy(:,ii)];
    M2xSamplesCh_f = fft2(M2xSamplesCh_t);
    ODataPolX = [ODataPolX; M2xSamplesCh_f(:,1).'];
    ODataPolY = [ODataPolY; M2xSamplesCh_f(:,2).'];
end

%=====
% DESCARTEM ZERO PADDING
%=====
Zeropadding = ofdmParam.Zeropadding;
ODataPolX_NoZeros = [ODataPolX(:,(ofdmParam.subcarriers/2 +Zeropadding+ 1):((
ofdmParam.subcarriers/2 +Zeropadding+1)+ofdmParam.subcarriers/2-1) )...
    ODataPolX(:,1:ofdmParam.subcarriers/2)
];
ODataPolY_NoZeros = [ODataPolY(:,(ofdmParam.subcarriers/2 +Zeropadding+ 1):((
ofdmParam.subcarriers/2 +Zeropadding+1)+ofdmParam.subcarriers/2-1) )...
    ODataPolY(:,1:ofdmParam.subcarriers/2)
];

```

```

ODataPolX_NoZeros = ODataPolX_NoZeros * sqrt ( 1 / mean ( mean ( abs ( ODataPolX_NoZeros ) .^
2 ) ) ) ;
ODataPolY_NoZeros = ODataPolY_NoZeros * sqrt ( 1 / mean ( mean ( abs ( ODataPolY_NoZeros ) .^
2 ) ) ) ;

%=====
% Estimació de canal + equalització
%=====
% Eliminem símbols d'entrenament
ST = ( exp ( 1i * pi .* ( ( 1 : ofdmParam.subcarriers ) .^ 2 ) / ofdmParam.subcarriers ) ).';

STMatrixPolX = [ODataPolX_NoZeros(1,:)];
STMatrixPolY = [ODataPolY_NoZeros(1,:)];

if (ofdmParam.simbolosOFDM<25)
    step = ofdmParam.simbolosOFDM-1;
    ODataPolXWithoutST = ODataPolX_NoZeros ( 2 : end, : );
    ODataPolYWithoutST = ODataPolY_NoZeros ( 2 : end, : );
else
    ODataPolXWithoutST = [ODataPolX_NoZeros(2:25,:)];
    ODataPolYWithoutST = [ODataPolY_NoZeros(2:25,:)];

    for ii=1:floor(ofdmParam.simbolosOFDM/25)
        STMatrixPolX = [STMatrixPolX ;ODataPolX_NoZeros((ii*(25+1)),:)];
        STMatrixPolY = [STMatrixPolY ;ODataPolY_NoZeros((ii*(25+1)),:)];
        if ii<(floor(ofdmParam.simbolosOFDM/25))
            ODataPolXWithoutST = [ODataPolXWithoutST;
ODataPolX_NoZeros((ii*(25+1)+1):ii*(25+1)+25,: )];
            ODataPolYWithoutST = [ODataPolYWithoutST;
ODataPolY_NoZeros((ii*(25+1)+1):ii*(25+1)+25,: )];
        else
            ODataPolXWithoutST = [ODataPolXWithoutST; ODataPolX_NoZeros((ii*(25+1)+1):end,:
)];
            ODataPolYWithoutST = [ODataPolYWithoutST; ODataPolY_NoZeros((ii*(25+1)+1):end,:
)];
        end
    end

end;
if (isfield(ofdmParam.SimParam,'equalization') && ofdmParam.SimParam.equalization == 0)
    ODataPolX_Est = ODataPolXWithoutST;
    ODataPolY_Est = ODataPolYWithoutST;
else
    %Estimació de canal
    for ii=1:floor(ofdmParam.simbolosOFDM/25)+1
        STMatrixPolX(ii,:) = STMatrixPolX(ii,:)./ST.';
        STMatrixPolY(ii,:) = STMatrixPolY(ii,:)./ST.';
    end
    %Ecuació per blocs de 25 símbols
    for ii=1:ofdmParam.simbolosOFDM-1
        ODataPolXWithoutST(ii,:) = ODataPolXWithoutST(ii,:)./STMatrixPolX(floor(ii/25)+1,:);
        ODataPolYWithoutST(ii,:) = ODataPolYWithoutST(ii,:)./STMatrixPolY(floor(ii/25)+1,:);
    end
    ODataPolX_Est = ODataPolXWithoutST;
    ODataPolY_Est = ODataPolYWithoutST;
end;

if (isfield(ofdmParam,'NormFactor'))
    ODataPolX_Est = ODataPolX_Est/ofdmParam.NormFactor;
    ODataPolY_Est = ODataPolY_Est/ofdmParam.NormFactor;
end
rx_data.ODataPolX_Est = ODataPolX_Est;
rx_data.ODataPolY_Est = ODataPolY_Est;

%=====
% DEMAPPING
%=====
% Demapateje 6PKPSK
Frames = ofdmParam.simbolosOFDM+floor (ofdmParam.simbolosOFDM/25);
rx_bindata=[];
for si=1:Frames-1-floor (ofdmParam.simbolosOFDM/25)
    for ii=1:2:(ofdmParam.subcarriers)
        if (esC2 (slicingB (ODataPolX_Est (si,ii),ODataPolY_Est (si,ii))))
auxC1=invertC1 (slicingB (ODataPolX_Est (si,ii+1),ODataPolY_Est (si,ii+1)));
            rx_bindata=[rx_bindata auxC1 (4)];
            rx_bindata=[rx_bindata auxC1 (3)];
            rx_bindata=[rx_bindata auxC1 (2)];
            rx_bindata=[rx_bindata auxC1 (1)];
            auxC2=deMapC2 (slicingB (ODataPolX_Est (si,ii),ODataPolY_Est (si,ii)));
            rx_bindata=[rx_bindata auxC2 (3)];
            rx_bindata=[rx_bindata auxC2 (2)];
            rx_bindata=[rx_bindata auxC2 (1)];
            rx_bindata = [rx_bindata 0];

```



```

        rx_bindata = [rx_bindata 1];

        elseif(esC2(slicingB(ODataPolX_Est(si,ii+1),ODataPolY_Est(si,ii+1))))
auxC2=deMapC2(slicingB(ODataPolX_Est(si,ii+1),ODataPolY_Est(si,ii+1)));
        rx_bindata=[rx_bindata auxC2(3)];
        rx_bindata=[rx_bindata auxC2(2)];
        rx_bindata=[rx_bindata auxC2(1)];
        auxC1=invertC1(slicingB(ODataPolX_Est(si,ii),ODataPolY_Est(si,ii)));
        rx_bindata=[rx_bindata auxC1(4)];
        rx_bindata=[rx_bindata auxC1(3)];
        rx_bindata=[rx_bindata auxC1(2)];
        rx_bindata=[rx_bindata auxC1(1)];

        rx_bindata = [rx_bindata 1];
        rx_bindata = [rx_bindata 1];

    else
        auxC1=invertC1(slicingB(ODataPolX_Est(si,ii+1),ODataPolY_Est(si,ii+1)));

        rx_bindata=[rx_bindata auxC1(4)];
        rx_bindata=[rx_bindata auxC1(3)];
        rx_bindata=[rx_bindata auxC1(2)];
        rx_bindata=[rx_bindata auxC1(1)];
        auxC2=invertC1(slicingB(ODataPolX_Est(si,ii),ODataPolY_Est(si,ii)));

        rx_bindata=[rx_bindata auxC2(4)];
        rx_bindata=[rx_bindata auxC2(3)];
        rx_bindata=[rx_bindata auxC2(2)];
        rx_bindata=[rx_bindata auxC2(1)];
        rx_bindata = [rx_bindata 0];

    end;
end;
end;
rx_data.bindata = rx_bindata;

function res = slicingB(Xcomplex,Ycomplex)
% Retorna el punt de la constel·lació mes proper del punt rebut
mapDist=[-2,0,0,0,9;
         2,0,0,0,9;
         0,-2,0,0,9;
         0,2,0,0,9;
         0,0,-2,0,9;
         0,0,2,0,9;
         0,0,0,-2,9;
         0,0,0,2,9;
         1,1,1,1,9;
         1,1,1,-1,9;
         1,1,-1,1,9;
         1,1,-1,-1,9;
         1,-1,1,1,9;
         1,-1,1,-1,9;
         1,-1,-1,1,9;
         1,-1,-1,-1,9;
         -1,1,1,1,9;
         -1,1,1,-1,9;
         -1,1,-1,1,9;
         -1,1,-1,-1,9;
         -1,-1,1,1,9;
         -1,-1,1,-1,9;
         -1,-1,-1,1,9;
         -1,-1,-1,-1,9];

for jj=1:length(mapDist)
    mapDist(jj,5)=sqrt(((mapDist(jj,1)-real(Xcomplex))^2)+((mapDist(jj,2)-
imag(Xcomplex))^2)+((mapDist(jj,3)-real(Ycomplex))^2)+((mapDist(jj,4)-imag(Ycomplex))^2));
end;
[M,I]=min(mapDist(:,5));
res=mapDist(I,[1,2,3,4]);
end
function res=esC2(vector)
% Retorna true si el punt rebut correspon a un dels vectors C2
A=[-2,0,0,0];B=[0,-2,0,0];C=[0,0,-2,0];D=[0,0,0,-2];
E=[2,0,0,0];F=[0,2,0,0];G=[0,0,2,0];H=[0,0,0,2];
if
(isequal(vector,A)||isequal(vector,B)||isequal(vector,C)||isequal(vector,D)||isequal(vector,E)||is
equal(vector,F)||isequal(vector,G)||isequal(vector,H))
    res=true;
else
    res=false;
end;
end
end

```

```

function res=deMapC2(vector)
% Demapeja el vector C2 rebut
A=[-2,0,0,0];B=[0,-2,0,0];C=[0,0,-2,0];D=[0,0,0,-2];
E=[2,0,0,0];F=[0,2,0,0];G=[0,0,2,0];H=[0,0,0,2];
I=[0,0,0,0];J=[0,0,1,0];K=[0,1,0,0];L=[1,0,0,0];
M=[1,1,1,1];N=[1,1,0,0];O=[1,0,1,1];P=[0,1,1,1];

if(isequal(vector,A)) res=L;
end;
if(isequal(vector,B)) res=O;
end;
if(isequal(vector,C)) res=N;
end;
if(isequal(vector,D)) res=M;
end;
if(isequal(vector,E)) res=P;
end;
if(isequal(vector,F)) res=K;
end;
if(isequal(vector,G)) res=J;
end;
if(isequal(vector,H)) res=I;
end;

end
function res=invertC1(vector)
% Inverteix els bits del vector
aux=[];
if((vector(1))<=0) aux(1)=1;
end;
if((vector(1))>0) aux(1)=0;
end;
if((vector(2))<=0) aux(2)=1;
end;
if((vector(2))>0) aux(2)=0;
end;
if((vector(3))<=0) aux(3)=1;
end;
if((vector(3))>0) aux(3)=0;
end;
if((vector(4))<=0) aux(4)=1;
end;
if((vector(4))>0) aux(4)=0;
end;
res=aux;

end
end

%*****OFDM4DSIMULATOR*****
%
% Parametros de entrada:
%
% ofdmParam: Parametros de senial OFDM
%
%   simbolosOFDM: simbols a transmetir
%   subcarriers: Subportadoras en cada simbols
%   constelacion: Modulacion digital seleccionada. Valores aceptados: 'PS-QPSK'
%   NFFT: muestras transmitidas en cada FFT
%   RSymb: Tasa de simbols
%   RFfreqHz: Frecuencia de portadora
%   PC: Porcentaje de prefijo ciclico a insertar, en tanto por uno. 1/16 (6.25%) por defecto
%   upsamplingFactor: Factor de sobremuestreo
%
% ofdmParam.SimParam: Parametros de simulacions
%
%   simulations: Numero de simulacions. Por defecto, 1
%   BEROSNR: >0 activa el calculo de grafica, en el rango [-BEROSNR,BEROSNR]
%   Spectrum: A 1, activa la representacion del espectro
%   equalization: a 1 activa la etapa equalizadora, a 0 deja desactivada esta etapa
%   Fiber: a 1 activa la etapa de fibra Optica SSMF
%
% laserParam:
%
%   LaserPower: potencia de salida del laser [W]
%   LoPower: potencia de salida del oscilador en recepcion [W]
%
% fiberParam:
%
%   fiberLength: longitud de la fibra
%   tramos: Numero de tramos SSME
%   DCF: a 1 activa el tramo de fibra compensadora de dispersion
%   wavelength: Longitud de onda central (m)

```

```

%      attenuation: atenuacion de la fibra
%      dispersionChrom: Parametro de dispersion de la fibra (s/m^2)
%
%*****

%=====
% Calculo de parametros OFDM secundarios
%=====
disp(strcat('Codificant senyal OFDM (PS-QPSK)...',int2str(ofdmParam.subcarriers),'
subportadores'));
SamplesCh=ofdmParam.NFFT; %nombre de mostres per cada simbol OFDM
disp(strcat('Nombre de mostres per cada simbolo OFDM:',int2str(SamplesCh)));
N = (ofdmParam.simbolosOFDM+floor(ofdmParam.simbolosOFDM/25)) * SamplesCh; %Total de mostres
transmeses incloent preàmbul 4%
disp('Total de mostres transmeses incloent preàmbul 4%:');
disp(N);
N = N*(1+ofdmParam.PC); %Total de mostres incloent percentatge de prefix cíclic a afegir
disp('Total de mostres incloent percentatge de prefix cíclic a afegir: ');
disp(N);
dt = 1 / ( ofdmParam.RSymb*SamplesCh );%Temps per mostra
disp('Temps per mostra: ');
disp(dt );
df =1/dt; %Frequència de mostreig
disp('Frequència de mostreig: ');
disp(df );
result.Rb = df*( ofdmParam.subcarriers /ofdmParam.NFFT) *4*1e-9; %Gbits/seg
disp('Gbits/seg: ');
disp(result.Rb );
time = ( 0 : ( SamplesCh-1) ) * dt ; %Finestra de temps
timepc = ( 0 : ( SamplesCh-1) * (1+ofdmParam.PC) ) * dt ; %Finestra de temps amb prefix cíclic
FREQ=(-(N/ 2) : (N/2-1) ) * 1 / ( dt*N); %Eix de freqüència
disp('length Eix de freqüència: ');
disp(length(FREQ));
Signal_Bandwidth = ( 2 /max ( timepc ) ) + ( ( ofdmParam.subcarriers-1) /max ( time ) ); %
disp('Signal_Bandwidth : ');
disp(Signal_Bandwidth );
fiberParam.Bo = Signal_Bandwidth;

%=====
% Generacion de datos binarios aleatorios
%=====
t_data=rand(1,ofdmParam.subcarriers*3*(ofdmParam.simbolosOFDM-1))>0.5;
t_data_Polar = 2*t_data-1;
result.polarBinData = t_data_Polar;
dataCounter=1;%PermitirÃ; seleccionar datos para cada simbolo OFDM
%=====
% Generacion senial OFDM (PROCESADO DIGITAL)
%=====
result.spdata = [];
for d=1:ofdmParam.simbolosOFDM-1
    data = t_data_Polar(dataCounter:dataCounter+ofdmParam.subcarriers*3-1); %Datos por trama
    dataCounter=dataCounter+ofdmParam.subcarriers*3;

%=====
% Conversion S/P
%=====
s_p_data = reshape(data,3,length(data)/3);
result.spdata = [result.spdata s_p_data];

%=====
% Mapping 4D (PS-QPSK)
%=====
dataPolX=[];
dataPolY=[];

for (jj=1: length(data)/3)

    %*****
    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==-1)
        Xj1 = 0+0i;
        Xj2 = 0+2i;
    end;
    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==1)
        Xj1 = 0+0i;
        Xj2 = 2+0i;
    end;
    if(s_p_data(3,jj)==-1) && (s_p_data(2,jj)==1) && (s_p_data(1,jj)==-1)
        Xj1 = 0+2i;
        Xj2 = 0+0i;
    end;
    if(s_p_data(3,jj)==1) && (s_p_data(2,jj)==-1) && (s_p_data(1,jj)==-1)
        Xj1 = -2+0i;
    end;
end;

```

```

Xj2 = 0+0i;
end;
if(s_p_data(3,jj)==1)&& (s_p_data(2,jj)==1)&&(s_p_data(1,jj)==1)
Xj1 = 0+0i;
Xj2 = 0-2i;
end;
if(s_p_data(3,jj)==1)&& (s_p_data(2,jj)==1)&&(s_p_data(1,jj)==-1)
Xj1 = 0+0i;
Xj2 = -2+0i;
end;
if(s_p_data(3,jj)==1)&& (s_p_data(2,jj)==-1)&&(s_p_data(1,jj)==1)
Xj1 = 0-2i;
Xj2 = 0+0i;
end;
if(s_p_data(3,jj)==-1)&& (s_p_data(2,jj)==1)&&(s_p_data(1,jj)==1)
Xj1 = 2+0i;
Xj2 = 0+0i;
end;

dataPolY = [dataPolY Xj2];
dataPolX = [dataPolX Xj1];

end
if (d == 1)
    MPolX = [];MPolY = [];
end
MPolX = [MPolX transpose([dataPolX])];
MPolY = [MPolY transpose([dataPolY])];

end

%=====
% Normalizacion
%=====
NormFactor = 1 /sqrt( mean ( mean ( abs (MPolX) .^ 2 ) ) );
MPolX = MPolX * NormFactor;
MPolY = MPolY * NormFactor;

%=====
% Simbolo de entrenamiento (Tipo Bloque).Suponemos desvanecimiento lento del canal
%=====
ST = ( exp (1i* pi .* ( ( 1 : ofdmParam.subcarriers ) .^ 2 ) / ofdmParam.subcarriers ) ).';
MpolX_withST = [];
MpolY_withST = [];
MPolX = [ST MPolX];
MPolY = [ST MPolY];
for ii=1:floor(ofdmParam.simbolosOFDM/25)
    MpolX_withST = [MpolX_withST MPolX(:, ((ii-1)*25+1):(ii*25)) ST];
    MpolY_withST = [MpolY_withST MPolY(:, ((ii-1)*25+1):(ii*25)) ST];
end
if (not(floor(ofdmParam.simbolosOFDM/25) == ( ofdmParam.simbolosOFDM/25)))
    MpolX_withST = [MpolX_withST MPolX(:,floor(ofdmParam.simbolosOFDM/25)*25+1:end)];
    MpolY_withST = [MpolY_withST MPolY(:,floor(ofdmParam.simbolosOFDM/25)*25+1:end)];
end;

%=====
% ZeroPadding Simetrico
%=====

result.Zeropadding =SamplesCh-ofdmParam.subcarriers;
result.OFDMSignalf_tx.PolX = [ MpolX_withST( (ofdmParam.subcarriers/2+1) :
ofdmParam.subcarriers, :);...
    zeros(result.Zeropadding,
ofdmParam.simbolosOFDM+floor(ofdmParam.simbolosOFDM/25));...
    MpolX_withST( 1 : ofdmParam.subcarriers/2, :)];

result.OFDMSignalf_tx.PolY = [ MpolY_withST( (ofdmParam.subcarriers/2+1) :
ofdmParam.subcarriers, :);...
    zeros(result.Zeropadding,
ofdmParam.simbolosOFDM+floor(ofdmParam.simbolosOFDM/25));...
    MpolY_withST( 1 : ofdmParam.subcarriers/2, :)];

%=====
% IFFT2: Calculo de subportadoras en el dominio temporal, en dos dimensiones
% Garantizamos ortogonalidad entre ambas polarizaciones
%=====

OFDMSignal_tx =
ifft2OFDM(result.OFDMSignalf_tx,ofdmParam.simbolosOFDM+floor(ofdmParam.simbolosOFDM/25));

%=====
% PREFIJO CICLICO
%=====

```

```

NumPC = SamplesCh*ofdmParam.PC;
OfdmSignal_t_PolX = [ OFDMSignalt_tx.PolX( ( SamplesCh-NumPC+1) : SamplesCh, : ) ;
OFDMSignalt_tx.PolX ] ;
OfdmSignal_t_Poly = [ OFDMSignalt_tx.Poly( ( SamplesCh-NumPC+1) : SamplesCh, : ) ;
OFDMSignalt_tx.Poly ] ;
result.OFDMSignalt_tx.PolX = reshape ( OfdmSignal_t_PolX ,N, 1 ) ;
result.OFDMSignalt_tx.Poly = reshape ( OfdmSignal_t_Poly ,N, 1 ) ;
%=====
% ESPECTRO DE LA SEÑAL TRANSMITIDA
%=====
if (isfield(ofdmParam.SimParam,'Spectrum') && ofdmParam.SimParam.Spectrum == 1)
[Pxx,W] = pwelch(result.OFDMSignalt_tx.PolX,[],[],length(FREQ),1);
[Pxy,W] = pwelch(result.OFDMSignalt_tx.Poly,[],[],length(FREQ),1);
figure (1);
plot(FREQ/max(FREQ)*Signal_Bandwidth,10*log10(fftshift(Pxx)),'r');
hold on;
plot(FREQ/max(FREQ)*Signal_Bandwidth,10*log10(fftshift(Pxy)),'b');
xlabel('frecuencia normalizada, Hz')
ylabel('Densidad espectral de potencia')
title('Espectro OFDM transmitido (Ambas polarizaciones)');

end;
%=====
% Upsampling (Interpolacion)
%=====
if (isfield(ofdmParam,'UpsamplingFactor') && ofdmParam.UpsamplingFactor > 0)
upsampled_t.PolX = interp(result.OFDMSignalt_tx.PolX,ofdmParam.UpsamplingFactor);
upsampled_t.Poly = interp(result.OFDMSignalt_tx.Poly,ofdmParam.UpsamplingFactor);
result.df = df*ofdmParam.UpsamplingFactor; %Nueva frecuencia de muestreo con oversampling
result.dt = 1/df;
else
upsampled_t.PolX = result.OFDMSignalt_tx.PolX;
upsampled_t.Poly = result.OFDMSignalt_tx.Poly;
end
%=====
% RF Up Conversion
%=====
result.OFDMSignalRF_tx.PolX = IQRF (upsampled_t.PolX,ofdmParam.RFfreqHz,dt);
result.OFDMSignalRF_tx.Poly = IQRF (upsampled_t.Poly,ofdmParam.RFfreqHz,dt);

%=====
% Modulacion Optica: CONF LASER + 2 IQ-Mach-Zehnder Anidados (DOBLE POL)
%=====
result.EOpt_Tx = genOptField(result.OFDMSignalRF_tx,ofdmParam,laserParam,dt);

if (isfield(ofdmParam.SimParam,'fixedp') && ofdmParam.SimParam.fixedp == 1)
power_resultEOpt_Tx1=mean(abs(result.EOpt_Tx(1,:)).^2)/1e-3;
power_resultEOpt_Tx2=mean(abs(result.EOpt_Tx(2,:)).^2)/1e-3;
optical_power_output_mW = laserParam.power;
factorPower1 = sqrt(optical_power_output_mW/power_resultEOpt_Tx1);
factorPower2 = sqrt(optical_power_output_mW/power_resultEOpt_Tx2);
result.EOpt_Tx(1,:) = result.EOpt_Tx(1,)*factorPower1;
result.EOpt_Tx(2,:) = result.EOpt_Tx(2,)*factorPower2;
end;
%=====
% Propagacion por fibra SSMF
%=====
if (isfield(ofdmParam.SimParam,'fiber') && ofdmParam.SimParam.fiber == 1)
fiberParam.dt = dt;
result.EOpt_Rx = fiberTx(result,fiberParam,ofdmParam);
ofdmParam.longitudFibraKm = fiberParam.fiberLength;%for title in graphic
else
result.EOpt_Rx = result.EOpt_Tx;
end;
%=====
% RX COHERENTE OFDM: OSC. LOCAL PBS,PBF,PDs BALANCEADOS,MIXER RF COHERENTE
%=====
%Set params
laserParam.dt = dt;
laserParam.RFfreqHz = ofdmParam.RFfreqHz;
ofdmParam.df = df;
ofdmParam.SamplesCh = SamplesCh;
ofdmParam.Zeropadding = result.Zeropadding;
ofdmParam.NormFactor = NormFactor;
ofdmParam.FREQ = FREQ;
ofdmParam.Signal_Bandwidth = Signal_Bandwidth;
titleOSNR = '';
if (isfield(ofdmParam.SimParam,'BEROSNR') && ofdmParam.SimParam.BEROSNR>0 )
%=====

```

```

% PROCESADO DIGITAL OFDM Rx + ESTUDIO BER/OSNR
%=====
if (isfield(ofdmParam.SimParam,'simulacions'))
    simulacions = ofdmParam.SimParam.simulacions;
else
    simulacions = 1;
end
if (ofdmParam.SimParam.BEROSNR >15)
    OSNRmin = 0 ;
    OSNRmax = ofdmParam.SimParam.BEROSNR;
else
    OSNRmin = -ofdmParam.SimParam.BEROSNR;
    OSNRmax = ofdmParam.SimParam.BEROSNR;
end
total_it = ((OSNRmax - OSNRmin) / 2) * simulacions;
s=sprintf('Starting OSNR measurement %d simulacions, range %d dB<=OSNR<= %d
dB:',simulacions,OSNRmin,OSNRmax);
disp(s);
tinit=tic; % needed for displaying progress
for sim = 1:simulacions
    i_snr = 1;
    BWOSNR = 3e8 / (0.1e-9);
    for osnr = OSNRmin:2:OSNRmax
        EOpt_Rx(1,:) = OSNR(result.EOpt_Rx(1,:),osnr,BWOSNR,df);

        EOpt_Rx(2,:) = OSNR(result.EOpt_Rx(2,:),osnr,BWOSNR,df);
        result.ElectricFields_Rx = rxOptCoherenteOFDM(EOpt_Rx,laserParam);
        ofdmParam.OSNR = osnr;
        rx_data = demodulatorDigOFDMPS(result.ElectricFields_Rx,ofdmParam);
        %Buffer OSNR/BER
        result.rx_data = rx_data;
        [number(i_snr),BER(sim,i_snr)] = biterr(t_data,rx_data.bindata);

        % Display progress
        if toc(tinit)>5
            s=sprintf('Measuring OSNR: %0.2f%% complete.', 100*((sim-1)* ((OSNRmax -
OSNRmin)/2) + i_snr-1)/total_it);
            disp(s);
            tinit=tic;
        end;
        i_snr=i_snr+1;
    end;
end;
%=====
% GRAFICA BER/OSNR
%=====

% Calculamos promedio sobre todas las simulaciones (ofdmParam.SimParam.simulacions)
for col=1:(OSNRmax - OSNRmin)/2;
    ber(1,col)=0;
    for row = 1:ofdmParam.SimParam.simulations;
        ber(1,col)=ber(1,col)+BER(row,col);
    end;
end
ber=ber./ofdmParam.SimParam.simulations;
figure;
ii=OSNRmin:2:OSNRmax-2;
result.ii = ii;
result.ber = ber;
for aup=1:length(result.ii)
    s=sprintf('SNR %0.1f: ber %d ', result.ii(aup),result.ber(aup));
    disp(s);
end;
semilogy(ii,ber);
title('BER vs OSNR');
ylabel('BER');
xlabel('OSNR (dB/0.1nm)');
grid on
titleOSNR = strcat(' OSNR = ',int2str(ofdmParam.SimParam.BEROSNR), 'dB/0.1nm,
',int2str(ofdmParam.subcarriers), ' subcarriers');
else
    result.ElectricFields_Rx = rxOptCoherenteOFDM(result.EOpt_Rx,laserParam);
%=====
% Procesado digital de la seÑal en Rx.
%=====

rx_data = demodulatorDigOFDMPS(result.ElectricFields_Rx,ofdmParam);
result.rx_data = rx_data;
result.tdata = t_data;
[i_snr,result.BerBER] = biterr(t_data,rx_data.bindata);
end

%=====
% Representacion de la constelacion recibida

```

```

%=====
%scatterplot(rx_data.ODataPolX_Est(1,:),1,0,'r. ');title(strcat('Constelacion Recibida (POL
X)',titleOSNR));
%scatterplot(rx_data.ODataPolY_Est(1,:),1,0,'b. ');title(strcat('Constelacion Recibida (POL
Y)',titleOSNR));

figure;
plot ( real ( rx_data.ODataPolX_Est), imag ( rx_data.ODataPolX_Est ) , ' . ' );
title(strcat('Constelacion Recibida (POL X)',titleOSNR));
ylabel('In-Phase');
xlabel('Quadrature');
figure;
plot ( real ( rx_data.ODataPolY_Est), imag ( rx_data.ODataPolY_Est ) , ' . ' );
title(strcat('Constelacion Recibida (POL Y)',titleOSNR, ' ',int2str(ofdmParam.subcarriers),
subcarriers));
ylabel('In-Phase');
xlabel('Quadrature');

%=====
% PROCESADO BERvsFiberLength
%=====
if (isfield(ofdmParam.SimParam,'fiber') && ofdmParam.SimParam.fiber == 1)
    if (isfield(ofdmParam.SimParam,'BERFIBER') && ofdmParam.SimParam.BERFIBER == 1)
        lengthf = fiberParam.fiberLength;
        itt = 1;
        for ii=50:25:lengthf
            fiberParam.fiberLength = ii;
            fiberln(itt) = fiberParam.fiberLength;
            result.EOpt_Rx = fiberTx(result,fiberParam,ofdmParam);
            result.ElectricFields_Rx = rxOptCoherenteOFDM(result.EOpt_Rx,laserParam);
            rx_data = demodulatorDigOFDMPS(result.ElectricFields_Rx,ofdmParam);
            result.rx_data = rx_data;
            result.tdata = t_data;
            [number(itt),BER(itt)] = biterr(t_data,rx_data.bindata);
            itt=itt+1;
        end
        for aup=1:itt-1
            s=sprintf('Distància %0.1f: BER %d ', fiberln(aup),BER(aup));
            disp(s);
        end;
    end;
end;

%=====
% GRAFICA BER/OSNR
%=====
figure;
result.BER = BER;
semilogy(fiberln,BER);
title('BER vs Fiber');
ylabel('BER');
xlabel('Length(Km)');
grid on
end;
end;

end

%=====
% Funcion: ifft2OFDM: Calculo de subportadoras en el dominio temporal para n simbolos OFDM-2D
%=====
function result = ifft2OFDM(signalf,frames)
    result.PolX= [];
    result.PolY= [];
    for ii =1:frames
        M2xSamplesCh = [signalf.PolX(:,ii) signalf.PolY(:,ii) ];
        M2xSamplesCh_t = ifft2(M2xSamplesCh);
        result.PolX = [result.PolX M2xSamplesCh_t(:,1)];
        result.PolY = [result.PolY M2xSamplesCh_t(:,2)];
    end
end

%=====
% Funcion: IQRf(signal)
%=====
function iqresult = IQRf (signal,RFfreqHz,dt)
    I = real ( signal );
    Q = imag ( signal );
    %ang =2* pi *RFfreqHz*( 1 :length(I))*dt;
    iqresult.VI = I;% .* cos( ang' );
    iqresult.VQ = Q;% .* sin( ang' );
end

function rx_data = demodulatorDigOFDMPS(Err , ofdmParam)

```

```

=====
% demodulatorDigOFDM()
% Paràmetres d'entrada:
% Erx: Vectors amb senyal elèctrica detectada (ambdues polaritzacions)
% ofdmParam: paràmetre de senyal OFDM
%
% DOWNSAMPLE + ESPECTRO + REMOVE CP + FFT2 + REMOVE ZEROPADDING + ECUALIZATION + SLICING + DEMAPPING
%
=====

%=====
% Downsample
%=====

if (isfield(ofdmParam,'UpsamplingFactor') && ofdmParam.UpsamplingFactor > 0)
    EoutxDownSamp = decimate(Erx(:,1),ofdmParam.UpsamplingFactor,8,'fir');
    EoutyDownSamp = decimate(Erx(:,2),ofdmParam.UpsamplingFactor,8,'fir');
else
    EoutxDownSamp = Erx(:,1);
    EoutyDownSamp = Erx(:,2);
end

%=====
% Espectre senyal Rx
%=====
if (isfield(ofdmParam.SimParam,'Spectrum') && ofdmParam.SimParam.Spectrum == 1)

    if (not( isfield (ofdmParam,'OSNR') ) || ofdmParam.OSNR >= 0 )
        figure(5);
        [Pxx,W] = pwelch(EoutxDownSamp,[],[],length(ofdmParam.FREQ),1);

plot(ofdmParam.FREQ/max(ofdmParam.FREQ)*ofdmParam.Signal_Bandwidth,10*log10(fftshift(Pxx)), 'r');
        hold on;
        [Pyy,W] = pwelch(EoutyDownSamp,[],[],length(ofdmParam.FREQ),1);

plot(ofdmParam.FREQ/max(ofdmParam.FREQ)*ofdmParam.Signal_Bandwidth,10*log10(fftshift(Pxx)));
        xlabel('Freqüència normalitzada, Hz')
        ylabel('Densitat espectral de potència')
        titleOsnr = '';
        titleFiber = '';
        if (isfield(ofdmParam.SimParam,'BEROSNR') && ofdmParam.SimParam.BEROSNR>0 )
            titleOsnr = strcat(' OSNR = ',int2str(ofdmParam.OSNR), 'dB');
        end
        if (isfield(ofdmParam,'fiberLength'))
            titleFiber = strcat(' Fibra SSMF: ',int2str(ofdmParam.fiberLength),' Kms');
        end
        title(strcat('Espectre rebut OFDM (ambdues polaritzacions)',titleOsnr,titleFiber));
        hold off;
    end;
end;

%=====
% Eliminem prefix cíclic
%=====
PC = ofdmParam.PC;
NumPC = ofdmParam.SamplesCh*PC;
Frames = ofdmParam.simbolosOFDM+floor(ofdmParam.simbolosOFDM/25);
ChsCP=ofdmParam.SamplesCh*(1+PC);
Esampx= reshape ( EoutxDownSamp , ChsCP , Frames );
Esampy= reshape ( EoutyDownSamp , ChsCP , Frames );
Esampx = Esampx ( ( NumPC+1 ) : ( ChsCP ) , : );
Esampy = Esampy ( ( NumPC+1 ) : ( ChsCP ) , : );

%=====
% FFT2
%=====
ODataPolX= [];
ODataPolY= [];
for ii =1:Frames
    M2xSamplesCh_t = [Esampx(:,ii) Esampy(:,ii)];
    M2xSamplesCh_f = fft2(M2xSamplesCh_t);
    ODataPolX = [ODataPolX; M2xSamplesCh_f(:,1).'];
    ODataPolY = [ODataPolY; M2xSamplesCh_f(:,2).'];
end

%=====
% DESCARTEM ZERO PADDING
%=====
Zeropadding = ofdmParam.Zeropadding;
ODataPolX_NoZeros = [ODataPolX(:,(ofdmParam.subcarriers/2 +Zeropadding+ 1):((
ofdmParam.subcarriers/2 +Zeropadding+1)+ofdmParam.subcarriers/2-1) )...
    ODataPolX(:,1:ofdmParam.subcarriers/2)
    ];
ODataPolY_NoZeros = [ODataPolY(:,(ofdmParam.subcarriers/2 +Zeropadding+ 1):((
ofdmParam.subcarriers/2 +Zeropadding+1)+ofdmParam.subcarriers/2-1) )...

```



```

ODataPolY(:,1:ofdmParam.subcarriers/2)
];

ODataPolX_NoZeros = ODataPolX_NoZeros * sqrt ( 1 / mean ( mean ( abs ( ODataPolX_NoZeros ) .^
2 ) ) ) ;
ODataPolY_NoZeros = ODataPolY_NoZeros * sqrt ( 1 / mean ( mean ( abs ( ODataPolY_NoZeros ) .^
2 ) ) ) ;
%=====
% Estimació de canal + equalització
%=====
% Eliminem símbols d'entrenament
ST = ( exp (1i* pi .* ( 1 : ofdmParam.subcarriers ) .^ 2 ) / ofdmParam.subcarriers ) .';

STMatrixPolX = [ODataPolX_NoZeros(1,:)];
STMatrixPolY = [ODataPolY_NoZeros(1,:)];

if (ofdmParam.simbolosOFDM<25)
    step = ofdmParam.simbolosOFDM-1;

    ODataPolXWithoutST = ODataPolX_NoZeros ( 2 : end, :);
    ODataPolYWithoutST = ODataPolY_NoZeros ( 2 : end, :);
else
    ODataPolXWithoutST = [ODataPolX_NoZeros(2:25,:)];
    ODataPolYWithoutST = [ODataPolY_NoZeros(2:25,:)];

    for ii=1:floor(ofdmParam.simbolosOFDM/25)
        STMatrixPolX = [STMatrixPolX ;ODataPolX_NoZeros((ii*(25+1)),:);];
        STMatrixPolY = [STMatrixPolY ;ODataPolY_NoZeros((ii*(25+1)),:);];
        if ii<(floor(ofdmParam.simbolosOFDM/25))
            ODataPolXWithoutST = [ODataPolXWithoutST;
ODataPolX_NoZeros((ii*(25+1)+1):ii*(25+1)+25,: )];
            ODataPolYWithoutST = [ODataPolYWithoutST;
ODataPolY_NoZeros((ii*(25+1)+1):ii*(25+1)+25,: )];
        else
            ODataPolXWithoutST = [ODataPolXWithoutST; ODataPolX_NoZeros((ii*(25+1)+1):end,:
)];
            ODataPolYWithoutST = [ODataPolYWithoutST; ODataPolY_NoZeros((ii*(25+1)+1):end,:
)];
        end
    end

end;
if (isfield(ofdmParam.SimParam,'equalization') && ofdmParam.SimParam.equalization == 0)
    ODataPolX_Est = ODataPolXWithoutST;
    ODataPolY_Est = ODataPolYWithoutST;
else
    %Estimació de canal
    for ii=1:floor(ofdmParam.simbolosOFDM/25)+1
        STMatrixPolX(ii,:) = STMatrixPolX(ii,:)./ST.';
        STMatrixPolY(ii,:) = STMatrixPolY(ii,:)./ST.';
    end
    %Ecuallització per blocs de 25 símbols
    for ii=1:ofdmParam.simbolosOFDM-1
        ODataPolXWithoutST(ii,:) = ODataPolXWithoutST(ii,:)./STMatrixPolX(floor(ii/25)+1,:);
        ODataPolYWithoutST(ii,:) = ODataPolYWithoutST(ii,:)./STMatrixPolY(floor(ii/25)+1,:);
    end
    ODataPolX_Est = ODataPolXWithoutST;
    ODataPolY_Est = ODataPolYWithoutST;
end;

if (isfield(ofdmParam,'NormFactor'))
    ODataPolX_Est = ODataPolX_Est/ofdmParam.NormFactor;
    ODataPolY_Est = ODataPolY_Est/ofdmParam.NormFactor;
end
rx_data.ODataPolX_Est = ODataPolX_Est;
rx_data.ODataPolY_Est = ODataPolY_Est;

%=====
% DEMAPPING
%=====
rx_bindata=[];

for si=1:Frames-1-floor(ofdmParam.simbolosOFDM/25)
    for (ii=1: ofdmParam.subcarriers)
        auxC2=deMapC2(slicingB(ODataPolX_Est(si,ii),ODataPolY_Est(si,ii)));
        rx_bindata=[rx_bindata auxC2(3)];
        rx_bindata=[rx_bindata auxC2(2)];
        rx_bindata=[rx_bindata auxC2(1)];
    end
end

rx_data.bindata = rx_bindata;

```

```
function res = slicingB(Xcomplex,Ycomplex)
% Retorna el punt de la constel·lació mes proper del punt rebut
mapDist=[-2,0,0,0,9;
          2,0,0,0,9;
          0,-2,0,0,9;
          0,2,0,0,9;
          0,0,-2,0,9;
          0,0,2,0,9;
          0,0,0,-2,9;
          0,0,0,2,9;
          ];

for jj=1:length(mapDist)
    mapDist(jj,5)=sqrt(((mapDist(jj,1)-real(Xcomplex))^2)+((mapDist(jj,2)-
imag(Xcomplex))^2)+((mapDist(jj,3)-real(Ycomplex))^2)+((mapDist(jj,4)-imag(Ycomplex))^2));
end;
[M,I]=min(mapDist(:,5));
res=mapDist(I,[1,2,3,4]);
end
function res=deMapC2(vector)
% Demapeja el vector C2 rebut
A=[-2,0,0,0];B=[0,-2,0,0];C=[0,0,-2,0];D=[0,0,0,-2];
E=[2,0,0,0];F=[0,2,0,0];G=[0,0,2,0];H=[0,0,0,2];
I=[0,0,0];J=[0,0,1];K=[0,1,0];L=[1,0,0];
M=[1,1,1];N=[1,1,0];O=[1,0,1];P=[0,1,1];

if(isequal(vector,A)) res=L;
end;
if(isequal(vector,B)) res=O;
end;
if(isequal(vector,C)) res=N;
end;
if(isequal(vector,D)) res=M;
end;
if(isequal(vector,E)) res=P;
end;
if(isequal(vector,F)) res=K;
end;
if(isequal(vector,G)) res=J;
end;
if(isequal(vector,H)) res=I;
end;
end
end
```