

TREBALL FINAL DE MASTER

Disseny i simulació d'una unitat de recuperació de l'estat de polarització per a sistemes OFDM òptics.

Autor: Francisco Gil Mayo.

Tutor: Josep Maria Fàbrega Sánchez

Màster Interuniversitari Enginyeria de telecomunicacions UOC-URL

Índex TFM:

1. Introducció	3
2. Conceptes bàsics	4
2.1. Birefringència i PMD de primer ordre.....	4
2.2. PMD de segon ordre	6
2.3. PMD com a barrera fonamental.....	7
2.4. Per què és millor CO-OFDM amb diversitat per combatre PDM?.....	8
2.5. OOFDM: Optical OFDM.....	10
2.6. Sistemes MIMO-OFDM.....	16
2.7. Esquema CO-OFDM amb SITO (1x2 MIMO OFDM).....	17
2.8. Esquema CO-OFDM amb TITO (2x2 MIMO CO-OFDM).....	19
2.9. Fibra òptica.....	20
2.10. SSPROP.....	23
3. Implementació codi MATLAB	27
3.1. CO-OFDM amb SITO (1x2 MIMO OFDM).....	27
3.2. CO-OFDM amb TITO (2x2 MIMO OFDM).....	36
4. Simulació i resultats	43
4.1. Sense fibra.....	43
4.2. CO-OFDM amb SITO (1x2 MIMO OFDM).....	44
4.3. CO-OFDM amb TITO (2x2 MIMO OFDM).....	47
4.4. SITO vs TITO.....	49
4.5. Taules comparatives.....	51
5. Conclusions i possibles millores	51
6. Codi MATLAB	52
7. Glossari	61
8. Bibliografia	63

Introducció:

Als darrers anys, les xarxes de telecomunicacions han de créixer i evolucionar de manera molt ràpida per poder satisfer i adaptar-se als canvis i reptes que apareixen en un entorn tan variable. La necessitat de major capacitat i velocitat de transmissió a les xarxes està motivat principalment per l'augment del nombre d'usuaris, la transmissió de grans volums de dades i l'aparició de noves aplicacions.

Paral·lelament, els sistemes de comunicació basats en fibra òptica també han sofert una important evolució tecnològica que afecta tant a la transmissió com a les funcionalitats de commutació, monitoratge i encaminament. Actualment, aquests avanços ens permeten disposar de solucions amb la capacitat de transmetre dades d'alta velocitat a llargues distàncies, conegudes com "long-haul", millorant notablement el rendiment de les xarxes, sense oblidar la vessant econòmica i la reducció de costos.

Com ja sabem, les xarxes troncales que transporten el trànsit agregat de múltiples xarxes per a altes velocitats i grans distàncies, empen normalment fibres monomode. Una forma d'augmentar la seva capacitat és incrementant la seva velocitat de transmissió, encara que per aconseguir-ho s'han de superar altres reptes que hi apareixen en aquest tipus de sistemes. Dos d'aquests reptes són la PMD i la dispersió cromàtica, i d'aquestes dos la PMD és la més difícil de corregir per la seva naturalesa estocàstica.

La dispersió cromàtica (CD) es produeix quan la font no és monocromàtica, és a dir, a l'emetre més d'una longitud d'ona els components espectrals del senyal es propaguen a diferents velocitats. En canvi, la dispersió de polarització (PMD) s'origina per la falta de simetria circular, fibra torçada o estrès de la mateixa.

Últimament, la dispersió de polarització ha despertat un gran interès entre els investigadors, ja que si per a sistemes modulats en intensitat operant a baixes velocitats, per exemple 2.5Gbps, la PMD no era massa greu i no tenia cap efecte negatiu, per als mateixos sistemes amb altes velocitats, 40Gbps o més, la dispersió de polarització esdevé molt significativa i es converteix en un greu problema a solucionar per a continuar desenvolupant i augmentant la capacitat de transmissió dels sistemes òptics.

Per sort, el renaixement dels sistemes coherents i l'arribada de l'OFDM a les comunicacions òptiques han fet que la PMD de primer ordre pugui ser estimada de manera eficaç i compensada. Les proves fetes amb sistemes CO-OFDM han demostrat que són esquemes molt més robustos enfront la PMD que altres configuracions que presenten menys tolerància a aquesta dispersió com els basats en detecció directa. A més, aquest disseny pot aportar una solució factible per la mitigació de la PMD de primer ordre ja existent en els enllaços de fibra instal·lada.

Tot i això, és interessant dir que l'aplicació d'aquestes tecnologies no impliquen el final de la investigació sobre la PMD. De fet, la dispersió de polarització ha ressorgit com una característica d'importància crítica del canal òptic que necessita ser reexaminat i els diferents estudis s'adrecen a reduir les PMD d'ordres superiors.

L'objectiu d'aquest treball és desenvolupar i analitzar totes les qüestions esmentades anteriorment a partir del disseny i simulació via MATLAB d'un receptor CO-OFDM que permeti estimar la dispersió de polarització de l'enllaç òptic a estudi. Per a aconseguir-ho, aplicarem diversitat al sistema de recepció i col·locarem un receptor OFDM per a cada mode de polarització.

Una variant interessant al sistema anterior serà introduir diversitat també al transmissor, el que ens permetrà fer ús de la multiplexació en polarització (PDM). Així podríem enviar les dades en els dos modes de polarització reduint a la meitat la velocitat de transmissió del sistema. Aquesta reducció de dades presenta certs avantatges com que la dispersió de la fibra ara seria menor i que tant el transmissor com el receptor tindrien una complexitat inferior en comparació amb un sistema que no faci servir PDM.

La memòria es divideix en tres capítols, el primer d'ells, "Conceptes bàsics", tracta d'explicar els principis i el funcionament de les diferents tecnologies i sistemes en els que es basa tot el projecte. De totes elles podem destacar les raons de l'aparició de les diferents dispersions que afecten a la fibra, el motiu pel qual la PMD s'ha convertit en un obstacle molt important per desenvolupar nous sistemes d'altas velocitats, la sinergia entre detecció coherent i OFDM, etc..

En el segon capítol, "Implementació codi MATLAB" s'expliquen com s'han realitzat les diferents parts del codi per dissenyar el transmissor i el receptor CO-OFDM i la simulació de la fibra òptica amb SSPROP. Es detallen els dos sistemes, SITO (1x2 MIMO OFDM) i TITO (2x2 MIMO OFDM).

Per últim, en el tercer capítol, "Simulacions i resultats", es presenten les diferents respostes que s'han obtingut per les dues configuracions. Entre elles, tenim la simulació sense fibra, com afecta a la constel·lació QPSK la fibra òptica i com és la seva compensació, comportament per a diferents longituds, etc.

2. Conceptes bàsics:

2.1. Birefringència i PMD de primer ordre:

Com ja sabem les fibres monomode treballen només en el mode fonamental, particularment en el mode HE₁₁, però, en realitat, això no és del tot cert ja que aquest "únic" mode es pot separar en dos modes degenerats polaritzats ortogonalment. La degeneració del modes significa que tots dos tenen la mateixa constant de propagació β .

En condicions ideals, on la fibra té una simetria circular perfecta i no hi ha cap d'elles torçada o estressada, la constant de propagació β es manté idèntica a tots dos modes, però a la pràctica, a causa de les imperfeccions de la fibra, del procés de fabricació o de la instal·lació posterior, cada mode té una constant de propagació diferent. Aquest fenomen es coneix com a birefringència i es caracteritza per:

$$B_m = |n_x - n_y| = \Delta n$$

on n_x i n_y són, respectivament, els índex de refracció de les polaritzacions en els eixos X i Y. En el treball hem fet servir una valor de birefringència de 10^{-7} .

Com a conseqüència de la birefringència, al transmetre un senyal per la fibra hi haurà un acoblament entre els dos modes que es repeteix de forma periòdica i que té una duració de:

$$L_B = \frac{2\pi}{\Delta\beta} = \frac{\lambda}{\Delta n}$$

Si calculem aquest valor per la nostra longitud d'ona de 1550nm, tenim un $L_B = 15,5m$.

Com que cada mode viatja a una velocitat de grup diferents ($v_g = c/n$), tindrem un eix on el mode es transmetrà més ràpid al tenir un índex modal més petit (eix ràpid) i un altre eix on el mode arribarà al receptor més tard (eix lent).

Ho podem veure millor en la següent imatge, el pols que entra a la fibra es divideix en dos components polaritzats ortogonalment, un mode va per l'eix X (eix lent) i l'altre mode per l'eix Y (eix ràpid). Cada mode, anomenat PSP, arriba al final de la fibra en un instant diferent provocant un retard entre els dos modes, aquesta diferència rep el nom de DGD i és la PMD de primer ordre.

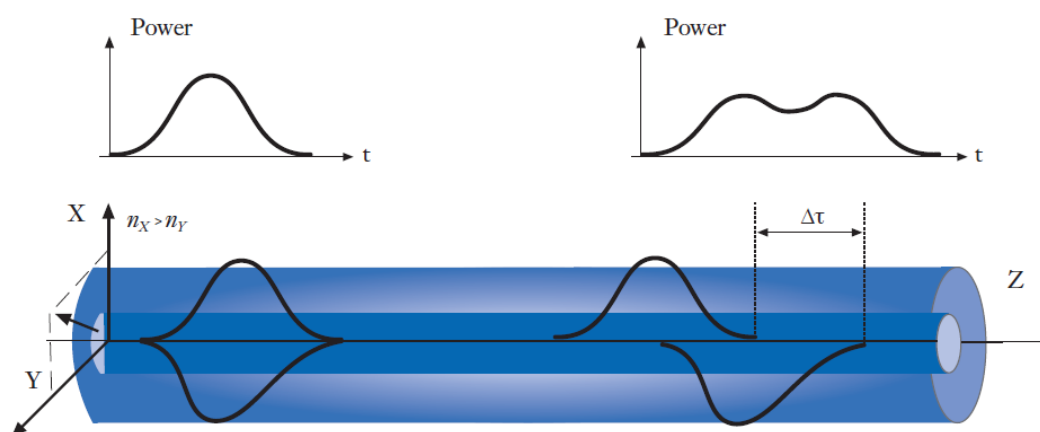


Figura 1. Impacte de la birefringència en un pols òptic.

A la figura 1 podem observar com ha canviat el pols a recepció, veiem que tenim un eixamplament temporal del pols i una disminució de la seva amplitud, aquest augment de l'ample de pols pot provocar ISI i que no es pugui recuperar la informació transmesa com cal.

A les fibres estàndards, la birefringència no és constant durant tota la fibra ja que aquesta es pot veure afectada durant la seva llargària per les imperfeccions de les condicions reals, esmentades abans, per tant, tindrà un comportament totalment aleatori. En conseqüència, la llum transmesa amb una polarització determinada al començament de la fibra anirà canviant la seva polarització al llarg de la fibra. Aquest canvi en la polarització és inofensiu per a llum d'ona contínua (CW) ja que la majoria de fotodetectors no detecten els canvis de polarització de la llum, però es torna en un greu problema quan treballem amb

sistemes de comunicació per a grans distàncies i a grans velocitats, que és el cas que ens ocupa.

L'extensió del pols transmès pot ser estimada a partir del retard ΔT que hi ha entre els dos PSP. Per una fibra de longitud L i amb birefringència B_m tenim:

$$\Delta T = \left| \frac{L}{v_{gx}} - \frac{L}{v_{gy}} \right| = L |\beta_{1x} - \beta_{1y}| = L \Delta \beta_1$$

on $\Delta \beta_1 = k_0 \frac{dB_m}{dw}$ està relacionat amb la birefringència de la fibra. L'equació anterior no pot ser utilitzada per estimar directament la PMD de primer ordre degut als canvis aleatoris de la birefringència de la fibra. De fet, la PMD es caracteritza pel valor RMS del retard ΔT obtingut després de fer el promig sobre les diferents pertorbacions aleatòries, la variància de ΔT serà:

$$\sigma_T^2 = \langle (\Delta T)^2 \rangle$$

Fent una aproximació per a longituds superiors als 100m. podem fer servir l'equació:

$$\sigma_T = D_p \sqrt{L}$$

on D_p es el paràmetre PMD, el valor utilitzat en el nostre treball és $D_p=0.04$ ps/ $\sqrt{\text{km}}$. L'ample del pols induït es relativament petit comparat amb els efectes de la dispersió cromàtica però, veient la seva dependència amb la longitud, es tornarà un factor limitant per a altes velocitats i per a llargues distàncies. Per norma, la PMD es torna perjudicial per a sistemes òptics que transmeten amb velocitats superiors a 10Gb/s en enllaços molt llargs (per sobre dels 500km).

2.2. PMD de segon ordre:

El vector PMD, $\vec{\Omega}(w) = \Delta\tau(w)\hat{q}(w)$, està format per la PMD de primer ordre $\Delta\tau(w)$ i el vector unitari $\hat{q}(w)$ que indica la direcció del eix ràpid vist anteriorment, com podem observar tots dos components estan en funció de la freqüència. La PMD de segon ordre es defineix, per tant, com la derivada del vector PMD respecte la freqüència:

$$PMD^{2 \text{ ordre}} = \frac{d\Omega(w)}{dw} = \frac{d\Delta\tau(w)}{dw} \hat{q}(w) + \Delta\tau(w) \frac{d\hat{q}(w)}{dw}$$

Els dos termes de la dreta de la igualtat són sovint tractats per separat com un component paral·lel i un altre perpendicular.

$$\Omega_{w//} = \frac{d\Delta\tau(w)}{dw} \hat{q}(w)$$

$$\Omega_{w\perp} = \Delta\tau(w) \frac{d\hat{q}(w)}{dw}$$

La component paral·lela $\Omega_{w//}$ causa una polarització que depèn de la dispersió cromàtica i la component perpendicular $\Omega_{w\perp}$ representa una freqüència que depèn de la rotació dels PSP. En la figura 2 veiem gràficament com totes dues components afecten a la

PMD de primer ordre, el vector inicial sofreix una rotació i una amplificació que depèn de la freqüència i que no és fàcil d'equalitzar.

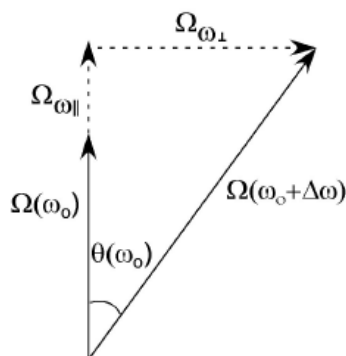


Figura 2. Vector PMD i els seus components derivats $\Omega_{\omega_{\parallel}}$ i $\Omega_{\omega_{\perp}}$

La majoria dels compensadors poden tractar la PMD de primer ordre i la compensen bastant bé. Tanmateix les penalitzacions de segon ordre es mantenen després de la compensació de primer ordre i, per tant, es converteixen en l'efecte dominant.

2.3. PMD com a barrera fonamental.

Com hem vist abans un senyal òptic que travessa una fibra monomode amb birefringència es divideix en dos polsos que arriben al receptor amb diferent retard i que provoca una distorsió del pols i ISI. La primera evidència de la penalització que provocava la PMD en el senyal va ser descrita per Poole on demostrava que la PMD estava relacionada amb el quadrat de la velocitat de transmissió.

Els investigadors han realitzat altres tipus d'aproximacions per a les penalitzacions de primer i segon ordre que permeten observar que a mesura que augmentem la velocitat de transmissió aquestes penalitzacions s'incrementen. Per a primer ordre tenim:

$$\Delta P_1 = A\Omega^2 T^{-2}$$

i pel segon ordre:

$$\Delta P_2 = B\Omega'^2 T^{-4}$$

on T és el període de bit del senyal, A i B són unes constants que depenen del format de modulació i Ω és el vector PMD. Es veu clarament la forta relació que tenen amb T, són inversament proporcionals i, al no ser una relació lineal, les penalitzacions augmenten exponencialment amb un increment petit del "bit rate". A la figura 3 es presenta com es comporta la penalització de la PMD de primer ordre per a diferents valors de velocitat de transmissió.

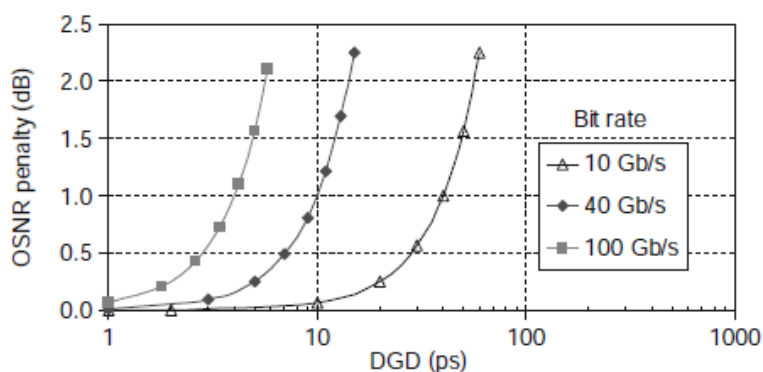


Figura 3. Penalitzacions OSNR segons DGD utilitzat.

Per a futurs sistemes de transmissió que treballin a 100Gbps o a 400Gps, per una correcta transmissió, la DGD ha d'estar limitada a 10ps i 4ps respectivament per a que les penalitzacions no superin els 1db.

Per tant, podem afirmar que la PMD s'ha convertit en la barrera fonamental per desenvolupar sistemes de comunicacions òptiques per a altes velocitats i a grans distàncies.

2.4. Per què és millor CO-OFDM amb diversitat per combatre PDM?

La sinergia entre la detecció coherent i OFDM és doble. Per una banda, el sistema coherent aporta a OFDM una linealitat molt necessària en la conversió optoelectrònica up i down, i per l'altra, la modulació OFDM contribueix amb una major eficiència i facilitat de càlcul de canal i estimació de fase.

De la modulació OFDM parlarem més àmpliament més endavant, però de forma resumida podem dir que és una modulació multiportadora molt utilitzada per combatre l'esvaïment multicamí. El seu principi és basa en transmetre les dades per diferents subportadores múltiples que siguin ortogonals entre elles per poder recuperar més tard la informació enviada. En el domini temporal s'afegeix un interval de guàrdia Δg per poder eliminar perfectament l'ISI provocada per la distorsió cromàtica i la PMD.

A la figura 4 veiem com funciona un detector CO-OFDM amb diversitat de polarització. A l'entrada de l'enllaç de fibra, cada subportadora (S_1, S_2, \dots) té la seva polarització alineada amb la resta de subportadores. Després de transmetre la informació, a la sortida, cada subportadora ha evolucionat de forma diferent depenent de la PMD de la fibra.

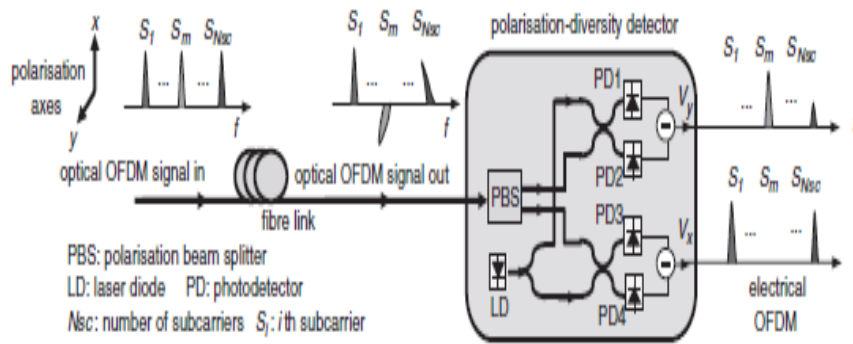


Figura 4. Receptor CO-OFDM robust a PMD.

Al receptor, el bloc PBS separa el flux de dades en dos components ortogonals de polarització i els connecta separatament a dos fotodetectors balancejats. Finalment, els senyals de sortida dels fotodetectors, V_x i V_y , són processats i tractats pel receptor OFDM per recuperar els símbols de cada subportadora.

Matemàticament, l'equació del canal es pot representar per:

$$\vec{C}'_{ki} = e^{j\phi_i} e^{j\phi_D(f_k)} U(f_k) \vec{C}_{ki} + \vec{n}_{ki}$$

on podem representar cada símbol d'informació rebut en forma d'un vector de Stokes:

$$\vec{C}'_{ki} = \begin{pmatrix} c_x^{ki} \\ c_y^{ki} \end{pmatrix}$$

Cada component d'aquest vector és la component de polarització en l'eix X i eix Y respectivament. A l'equació del canal, ϕ_i és l'error de fase comú OFDM pròpia del soroll de fase dels làsers i oscil·ladors locals, ϕ_D és la dispersió de fase introduïda per la dispersió cromàtica, U és la matriu de Jones i \vec{n}_{ki} és soroll. Hem de tenir en compte que hem suposat des de l'inici que no hi hauria soroll de fase, $\phi_i = 0$, per tant, l'equació de canal es simplificaria una mica:

$$\vec{C}'_{ki} = e^{j\phi_D(f_k)} U(f_k) \vec{C}_{ki} + \vec{n}_{ki}$$

En definitiva, observem que l'impacte de la PMD de primer ordre, per a cada subportadora, és només una simple rotació del vector de Jones \vec{C}'_{ki} , i que es pot compensar fàcilment i no provoca un greu deteriorament del senyal.

A més, la sensibilitat a la PMD està només determinada per l'espai entre subportadores i no depèn del nombre d'aquestes (NSC), per tant, un augment en la velocitat de transmissió del sistema amb el mateix espai de canal i un major nombre de portadores serà igual de robusta enfront la PMD. Per contra, en un sistema amb una sola portadora el deteriorament augmenta de forma dràstica amb l'augment de velocitat de transmissió. En els sistemes basats en detecció directa, l'enviament d'una portadora òptica principal amb la resta de subportadores OFDM, provoca forts esvaïments, i per tant, té una pitjor tolerància a la PMD que la detecció coherent.

2.5. OOFDM: Optical OFDM.

L'aplicació de la modulació OFDM a les comunicacions òptiques va succeir sorprenentment molt tard amb relació altres tecnologies. El principal interès que té l'OFDM per al món de les comunicacions òptiques és la seva robustesa contra la dispersió del canal òptica, però no va ser fins l'any 2001 que es va proposar, per primer cop, el seu ús per a combatre-la.

En els darrers anys, el fort interès que ha suscitat l'OFDM es deu en gran part a les diferents propostes que s'han presentat per donar solucions als sistemes de comunicacions en fibra òptica per a llargues distàncies.

Entre els diferents avantatges que té destacaríem:

- El format multiprotadora enfront altres tipus de modulació de portadora única.
- La reducció de la taxa de símbols implica una menor interferència intersimbòlica (ISI) i una millor protecció davant el retard del senyal i els desajustos de sincronització.
- Es realitza una millor i més senzilla equalització del canal degut a la separació de la informació en portadores ortogonals, el que facilita l'equalització en freqüència.
- Es una bona alternativa de futur gracies a la seva gran flexibilitat com a modulació multinivell. No requereix canvis de hardware, només s'ha de modificar els algorismes dins dels DSP (Digital Signal Processor).

Però, malauradament també presenta desavantatges:

- El sistema presenta una alta PAPR (Peak-to-Average Power Ratio), el que significa que els símbols OFDM transmesos tenen uns marges considerables entre la potència mínima i la màxima, presentant uns pics de potència gens negligibles. En aquesta situació, els amplificadors de potència del transmissor han de tenir una elevada linealitat, el que encareix el seu preu. A més, aquests pics de potència poden causar certa distorsió de senyal pels efectes no lineals de la fibra.
- La taxa de transferència de bits es veu afectada per una sèrie d'overheads que són necessaris per a la construcció dels sistema, com són el prefix cíclic, la quantitat de TS, etc.
- La modulació OFDM presenta una sensibilitat especial al "Frequency Offset" ja que es basa en l'eficiència espectral que ofereix l'ortogonalitat de les subportadores i si aquesta no es manté, tindrem problemes amb la interferència entre portadores (ICI).

2.5.1 Principis bàsics

Mitjançant la modulació OFDM, cada símbol es transmet ocupant un temps major en una amplada de banda que és menor respecte a altres modulacions d'una sola portadora. En aquests darrers casos, els símbols es transmeten en sèrie i cadascun ocupa una amplada de banda, en canvi, en les modulacions OFDM, el conjunt total de símbols es divideix en grups de N símbols, cada grup es transmet en paral·lel i ocupa una freqüència portadora diferent. Per a la transmissió d'un símbol en particular, es disposa d'un temps N vegades major que en el cas

de portadora única, mentre que l'amplada de banda disponible per a cada símbol queda dividida per N.

En el nostre cas, amb una codificació QPSK i una velocitat de transmissió de 10Gbps tenim un $r=5\text{Gbauds}$ (2bits/símbol) i, amb $N=64$ subportadores i $T=0,2\text{ns}$, l'ocupació freqüencial de cada símbol serà:

$$r_p = \frac{r}{N} = 78,125\text{Mbauds/portadora}$$

i l'ocupació temporal:

$$T_p = NT = 12,8\text{ns}$$

A la figura 5 representem aquesta seqüència de símbols sencera mitjançant un diagrama d'ocupació temps-freqüència, on cada rectangle, blanc o gris, representa l'ocupació d'un símbol determinat. Clarament, s'observa com repartim la l'ocupació freqüencial entre les portadores però, a canvi, augmentem la durada de símbol.

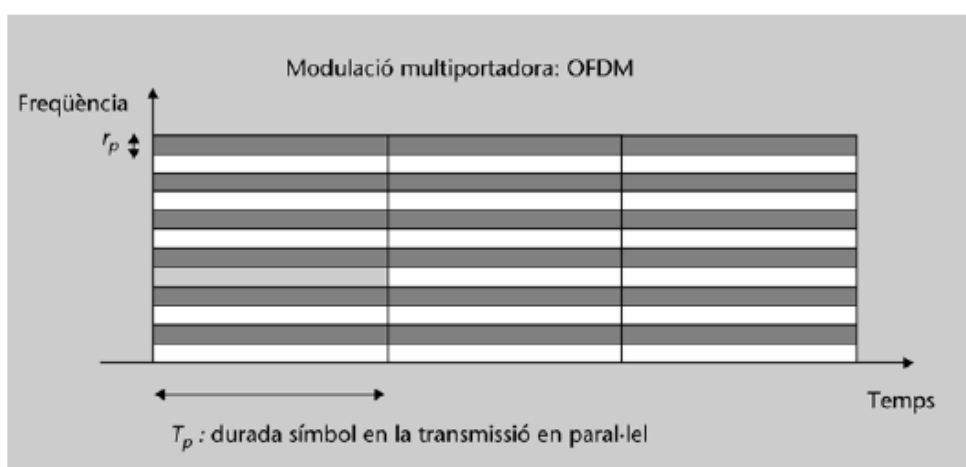


Figura 5. Ocupació temps-freqüència modulació OFDM.

A continuació, es pot comparar la resposta en el domini temporal respecte la resposta en el domini freqüencial. A l'esquerra, cada grup de N símbols, com veiem abans, ocupa una freqüència diferent dels altres grups, a la dreta cada subportadora es representa mitjançant una funció sinc, ja que s'assumeix que els polsos temporals són rectangulars, una cosa habitual en OFDM.

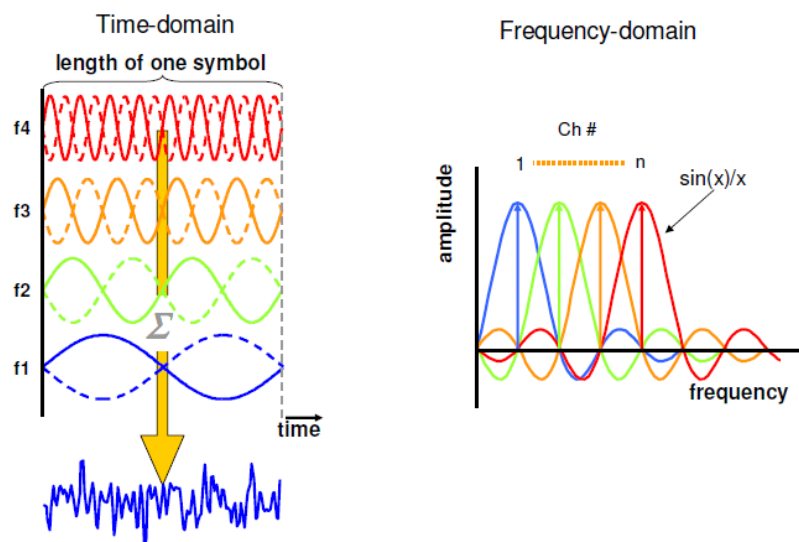


Figura 6. Domini temporal vs. domini freqüencial de modulació OFDM.

2.5.2 Condició d'ortogonalitat:

Les funcions que transporten els símbols de les diferents portadores corresponen a exponencials complexes de freqüències f_k .

$$y_k(T_p) = \sum_{k'=1}^N a[k'] \frac{1}{T_p} \int_0^{T_p} e^{j2\pi(f_{k'} - f_k)t} dt$$

Aquestes funcions han de ser ortogonals entre elles perquè en el receptor es puguin separar els símbols que corresponen a cada una de les portadores sense interferir-se entre elles. Per aconseguir la condició d'ortogonalitat hem de triar les freqüències de manera que cadascuna tingui un nombre enter de períodes en la T_p (durada del símbol OFDM). És a dir, es trien funcions amb valor de freqüències:

$$f_k = \frac{k}{T_p}; k = \frac{-N}{2}; \dots; \frac{N}{2} - 1$$

La separació freqüencial entre dues freqüències consecutives serà :

$$\Delta f = \frac{1}{T_p}$$

En el nostre cas amb $N=64$ i $T_p=12,8\text{ns}$ tenim:

$$\Delta f = \frac{1}{T_p} = 78,125\text{Mhz}$$

2.5.3. Efecte del canal de comunicacions en OFDM:

Una de les característiques principals de l'OFDM consisteix en la introducció d'un temps de guàrdia entre símbol i símbol OFDM durant el qual es transmet una repetició parcial del símbol OFDM següent. Amb aquesta estratègia es combat l'efecte ISI que podria provocar

un canal selectiu en freqüències alhora que preserva l'ortogonalitat entre les diferents portadores.

Per aconseguir-ho, la primera part del símbol, d'una durada donada T_G , s'estén cíclicament i es transmet de nou al final del símbol, el tram repetit es diu prefix cíclic. D'aquesta manera és cert que s'augmenta la duració de temps, provocant una reducció de la taxa efectiva d'informació, però ens assegurem mantenir la forma d'ona de cada símbol i la informació modulada en les subportadores.

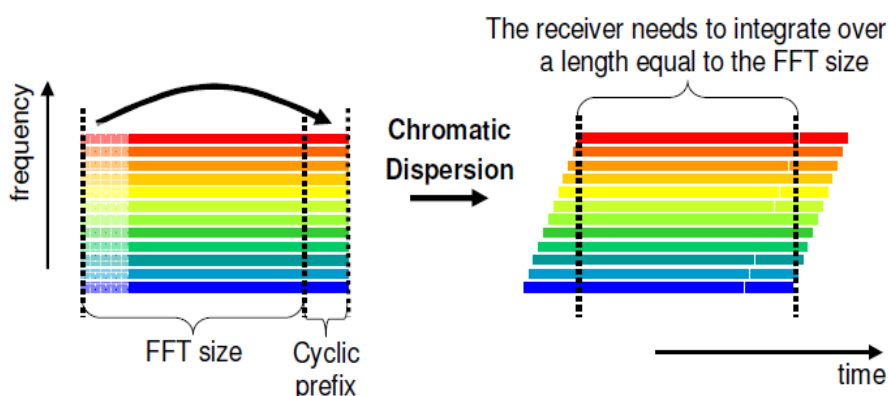


Figura 7. Introducció de l'interval de guàrdia.

En el nostre cas amb $N=64$ i $r=5\text{Gsimbols/s}$ i temps de símbol $T_s=0,2\text{ns}$ tenim:

- Temps adjudicat al símbol OFDM:

$$T_{OFDM} = NT_s = 12,8\text{ns}$$

- Hem suposat un prefix cíclic de 25% del símbol útil, per tant:

$$T_G = \frac{T_p}{4} = \frac{T_{OFDM}}{5} = 2,56\text{ns}$$

- Temps adjudicat a la transmissió de $N=64$ símbols en paral·lel:

$$T_p = T_{OFDM} - T_G = 10,24\text{ns}$$

- Amplada de banda resultant per a la modulació OFDM:

$$B_s = \frac{N}{T_p} = 6.25\text{Ghz}$$

La introducció del temps de guarda amb la transmissió del prefix cíclic evita la interferència intersimbòlica (ISI) i la interferència entre portadores (ICI), sempre que es compleixi la condició que la durada del temps de guarda sigui superior o igual que la durada de la resposta impulsional del canal. Aquesta condició de disseny s'ajusta amb el paràmetre de "Delay spread" del canal en qüestió:

$$\tau_m < T_G$$

En aquestes condicions, l'únic efecte rellevant sobre els símbols és que cada un d'aquests queda multiplicat pel valor de la funció de transferència del canal, justament a la freqüència portadora a la que ha estat transmès.

Per últim, la figura 8 mostra com seria la modulació OFDM vista en dues dimensions, tant en freqüència com en temps i que permet observar de forma clara el que hem comentat fins ara. En el domini temporal, s'afegeix el CP entre els diferents símbols de informació i, com en freqüència, s'agrupen en conjunts de N subportadores.

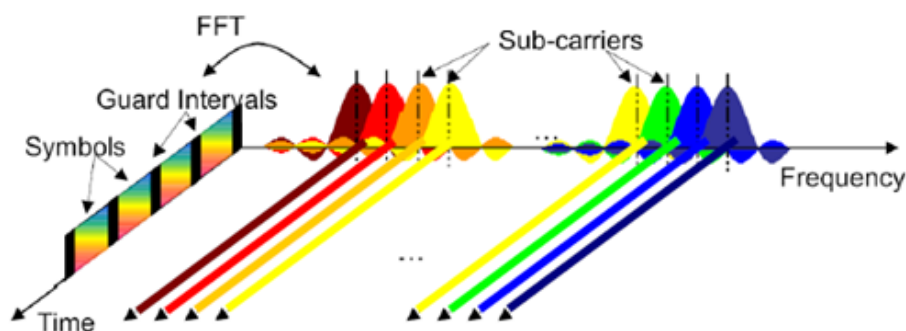


Figura 8. OFDM en dos dimensions:temporal i freqüencial.

2.5.4. PAPR

Com hem comentat anteriorment, un dels principals desavantatges de la modulació OFDM és que presenta un elevat PAPR, que redueix l'eficiència del sistema. Existeixen diverses tècniques per reduir el PAPR: clipping, tècniques de codificació, probabilitat, etc. Nosaltres farem servir el clipping per la seva senzillesa.

El PAPR pot ser calculat com:

$$PAPR = \frac{\max(x^2(t))}{\text{mean}(x^2(t))}$$

on $x(t)$ representa l'amplitud del senyal complex.

Si el nostre senyal fos, per exemple, el de la figura 9, eliminaríem tots els valors que superin k vegades la desviació estàndard de les dades.

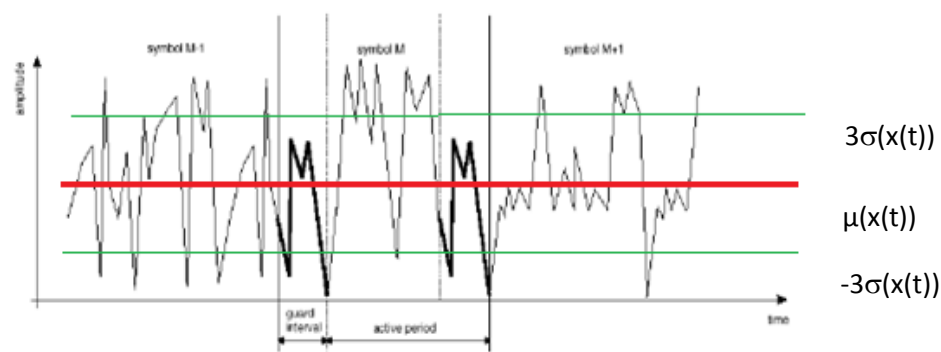


Figura 9. Senyal OFDM en temps.

En el nostre treball hem utilitzat una $K=3$, en aquesta situació no eliminem quasi cap valor de les dades ja que representa un 99,7% de les dades d'una distribució normal.

2.5.5 Esquema de transmissió i recepció OOFDM:

En la transmissió, els bits a enviar s'introdueixen en el sistema en sèrie i el primer que es troben és un bloc S/P que els transforma de sèrie a paral·lel en grups de N subportadores, després passen per un bloc de mapeig per a codificar-los en símbols, en el nostre cas hem triat la modulació QPSK, però podria ser un altre: BPSK, m-QAM, etc.

A continuació, tenim el bloc IFFT, habitualment es treballa amb valors de N compresos entre 64 i 8182, aquest algoritme ens facilita, des d'un punt de vista de cost computacional, realitzar la transformada del domini freqüencial al temporal i, a més, garanteix l'ortogonalitat de les portadores que havien imposat abans. A continuació s'afegeix el prefix cíclic (CP) per evitar la interferència intersimbòlica (ISI).

Tornem a transformar les dades en sèrie per a poder fer la conversió digital analògica i passem les dades per un filtre passabaix que recupera la rèplica del senyal en banda base. Les components I i Q del senyal codificat passen per un modulador òptic IQ que està excitat per un làser. Aquest modulador transforma el senyal elèctric en senyal òptic equivalent i envia la informació per la fibra òptica.

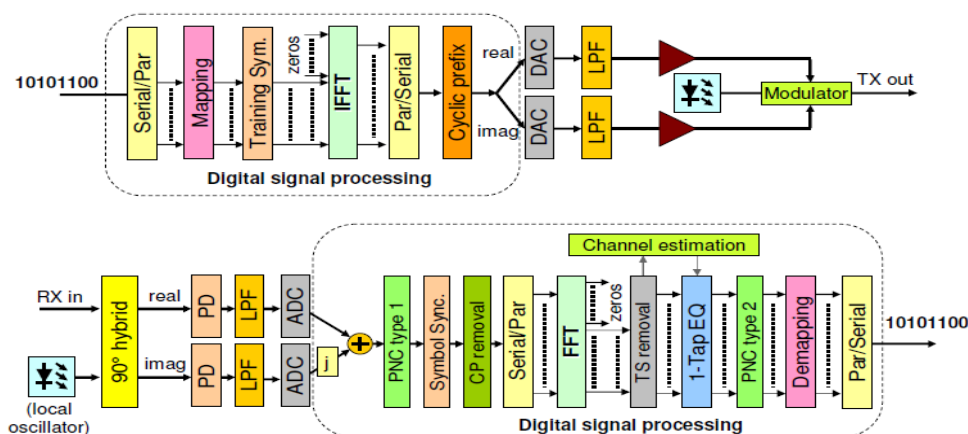


Figura 10 Esquema bàsic per blocs transmissió i recepció OOFDM.

En recepció, realitzem els passos de forma inversa als fets a la transmissió. Primer, trobem un híbrid òptic de 90° que combina els feixos de llum del senyal rebut amb els d'un làser d'un oscil·lador local per detectar els components en fase i quadratura del senyal òptic.

A continuació, la combinació de tots dos senyals passa pels fotodetectors balancejats per obtenir el senyal elèctric i, tot seguit, fem la conversió analògica digital. Després, traiem el prefix cíclic i convertim el flux de dades de sèrie en paral·lel. A continuació, realitzem la FFT per passar del domini temporal al freqüencial i equalitzem el canal. Per últim, demodulem els símbols rebuts i tornem a passar les dades per un bloc P/S per obtenir els bits originals.

2.6. Sistemes MIMO-OFDM.

En general, l'arquitectura de sistemes CO-OFDM amb MIMO es pot classificar en quatre categories depenent del nombre de transmissors i receptors utilitzats. A la imatge inferior es poden veure aquestes quatre configuracions:

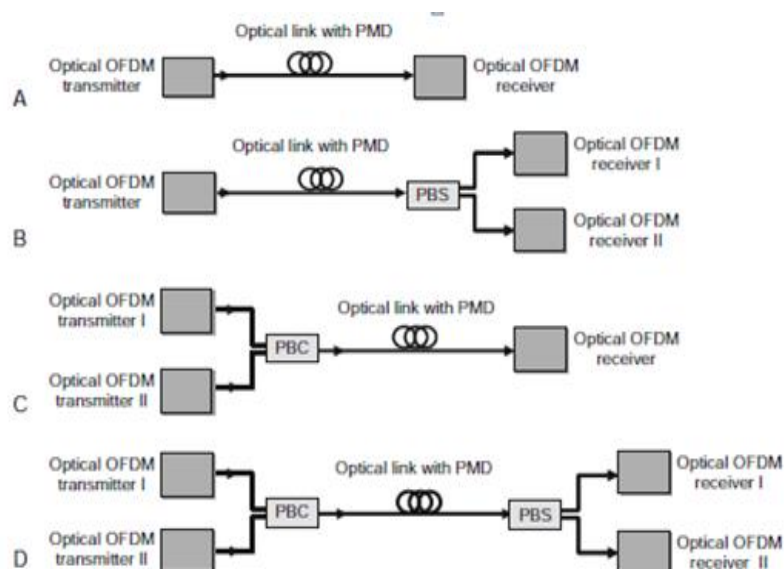


Figura 11. Diferents configuracions MIMO-CO-OFDM:(a) SISO,(b) SITO,(c) TISO,(d) TITO.

- **SISO (Single-Input Single-Output)**, correspon al model A de la figura 11, on hi ha només un transmissor i un receptor. És necessari col·locar un controlador de polarització abans del receptor per a alinear el senyal de polarització amb la polarització de l'oscil·lador local. Fins i tot, si hi trobem una gran PMD a la transmissió, pot passar que aquest controlador de polarització no funcioni correctament per la falta d'una polarització de subportadora uniforme amb la que es pugui alinear el làser local del receptor. Per tant, aquesta configuració presentarà distorsions al senyal degut a la PMD i necessitarà un controlador extern.
- **SITO (Single-Input Two-Output)**, correspon al model B, on hi ha un sol transmissor i dos receptors, un per a cada polarització. El receptor es coneix com receptor amb diversitat de polarització i no necessita un control de polarització com en el cas anterior. Veiem que entre el receptor i l'enllaç de fibra hi ha un dispositiu PBS que separa el flux de dades en dues components amb estats de polarització ortogonals. L'impacte de la PMD a la transmissió CO-OFDM és simplement la rotació de polarització de la subportadora, que es pot compensar de manera senzilla mitjançant l'estimació del canal i la reconstrucció de les constel·lacions. Per tant, aquesta configuració és més robusta enfront la PMD quan fem servir un receptor amb diversitat de polarització. A més, la introducció de PMD a l'enllaç millorarà els marges del sistema contra els esvaïments induïts per pèrdues dependents de la polarització, com succeeix amb les millores introduïdes per "delay spread" als sistemes inalàmbrics.

- TISO (Two-Input Single-Output)**, pertany al gràfic del model C i disposem de dos transmissors, un per cada polarització i un sol receptor. El transmissor es conegut com transmissor amb diversitat de polarització i entre ell i l'enllaç de fibra hi ha un dispositiu PBC que uneix els senyals dels dos transmissor en un sol flux. Configurant correctament la informació a la transmissió OFDM, es pot realitzar la comunicació CO-OFDM sense necessitat d'un controlador al receptor. Una possible configuració és fer que al transmissor es transmeti dues vegades consecutives el mateix símbol però amb polarització ortogonals. S'observa que el transmissor amb diversitat de polarització pot ser també resistent a la PMD sense necessitat de control, per tant, tenim un comportament semblant als esquema TISO. Encara que, com la mateixa informació és transmesa i repetida dues vegades consecutives, les eficiències elèctriques i òptiques es redueixen a la meitat i els requeriments d'OSNR es doblen comparant-los amb l'esquema SITO.
- TITO (Two-Input Two-Output)**, correspon al model D. Ara disposem de transmissors i receptors amb diversitat de polarització. També disposem d'un PBC a l'entrada de l'enllaç de fibra que uneix els dos fluxos de dades i d'un PBS al final de l'enllaç que els torna a separar. És l'esquema adient per realitzar modulació de polarització o multiplexació de polarització. La capacitat del sistema és el doble comparada amb l'esquema SITO. Com l'impacte de la PMD és simplement la rotació de polarització de la subportadora, es pot compensar amb estimació de canal i reconstrucció de constel·lació. A més, el doble de capacitat del canal no es veu afectat per la PMD.

En resum, s'observa que totes les arquitectures anteriors són robustes enfront la PMD excepte el primer esquema, SISO.

Per la seva banda, el model TISO no és tan eficient com els altres dos restants, per tant, les configuracions a estudi seran la SITO (*Single-Input Two-Output*) i la TITO (*Two-Input Two-Output*) que són les que millors prestacions ofereixen.

2.7. Esquema CO-OFDM amb SITO (1x2 MIMO OFDM):

A continuació representem l'esquema CO-OFDM amb arquitectura SITO, on disposem d'un transmissor i un receptor amb diversitat de polarització:

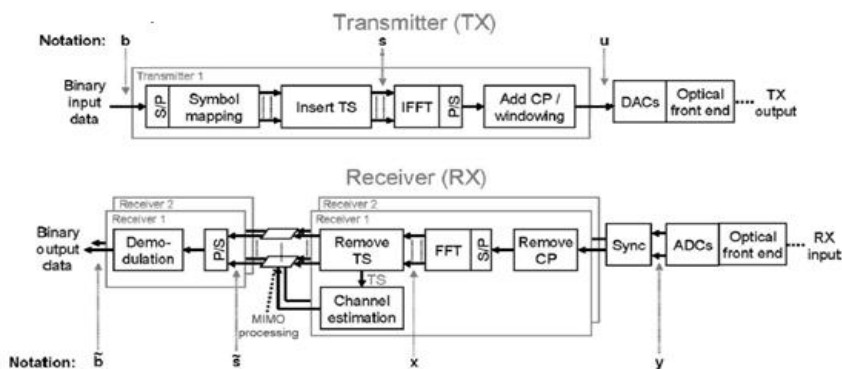


Figura 12. Esquema SITO amb recuperació de PMD.

Com hem comentat en l'apartat d' OOFDM, inicialment les dades a transmetre es transformen de sèrie a paral·lel amb un bloc S/P i, a continuació, es realitza un mapeig per convertir els bits a símbols, en el nostre cas fem servir la modulació QPSK (2bits/símbol). Després, s'introdueixen, periòdicament, els TS (Training Symbols), que ens ajudaran al sincronisme i a l'estimació del canal, creant el senyal complex s i seguidament, amb el bloc IFFT, modulem per obtenir les subportadores ortogonals.

Després afegim CP (Ciclic Prefix) per a mitigar el deteriorament produït per les distorsions i passem les dades de paral·lel a sèrie. Per últim, amb el DAC convertim el senyal de digital a analògic i l'enviem per l'enllaç de fibra òptica.

Al receptor, el senyal analògic que ens arriba per l'enllaç òptic es digitalitza fent servir ADC's on obtenim el vector y . A continuació, s'aplica sincronisme de símbol, els dos receptors operen igual així que només hem de recuperar el temps de símbol una sola vegada. Seguidament, transformem el senyal de sèrie a paral·lel i passem al domini freqüencial amb la FFT, on tenim el vector x . Després, es treuen els TS del payload i els utilitzem per poder estimar el canal. A continuació, es fa servir el processament MIMO per a cada subportadora per compensar la rotació de polarització en el símbols rebuts. Finalment, aquests símbols passen pel bloc S/P per a passar de paral·lel a sèrie i són demapejats per passar de símbols a bits, i com a resultat, obtenim el vector \tilde{b} .

Per reduir la complexitat computacional del detector MIMO aprofitem el coneixement del canal al receptor. Assumint un sistema perfectament sincronitzat, el senyal rebut abans del processament MIMO serà:

$$\vec{x}(k) = \vec{H}(k)\vec{s}(k) + \vec{n}(k)$$

on

$$\vec{H}(k) = \begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix}$$

representa la matriu de Jones i el vector $\vec{n}(k)$ representa el soroll en el domini freqüencial per a cada subportadora k .

El detector MIMO més senzill fa servir l'algoritme ZF (Zero Forcing), bàsicament el que fa és multiplicar els senyals rebuts \vec{x} amb la pseudoinversa de la matriu del canal estimat per trobar una estimació del vector transmès, és a dir:

$$\vec{s}(k) = \tilde{H}^+(k)\vec{x}(k)$$

on $\tilde{H}^+(k)$ és la pseudoinversa de la matriu del canal i es calcula com: $\tilde{H}^+ = (H^T H)^{-1} H^T$

Tot i que amb l'algoritme ZF és fa una bona estimació, tindrem un cert error degut al soroll afegit pels amplificadors òptics i altres fonts de soroll, una bona solució seria utilitzar un estimador MMSE que, al ser un estimador adaptatiu, serà més robust davant de possibles distorsions del senyal i donarà una millor compensació del canal davant el soroll afegit al canal.

2.8. Esquema CO-OFDM amb TITO (2x2 MIMO CO-OFDM):

Aquesta configuració és l'adient per PDM (Polarization Division Multiplexing) on disposem de d'un transmissor i un receptor, tots dos amb diversitat de polarització. La multiplexació de polarització és un mètode molt efectiu per doblar la eficiència espectral dels sistemes de transmissió. La figura 13 representa un sistema PDM amb MIMO que utilitza el sistema CO-OFDM.

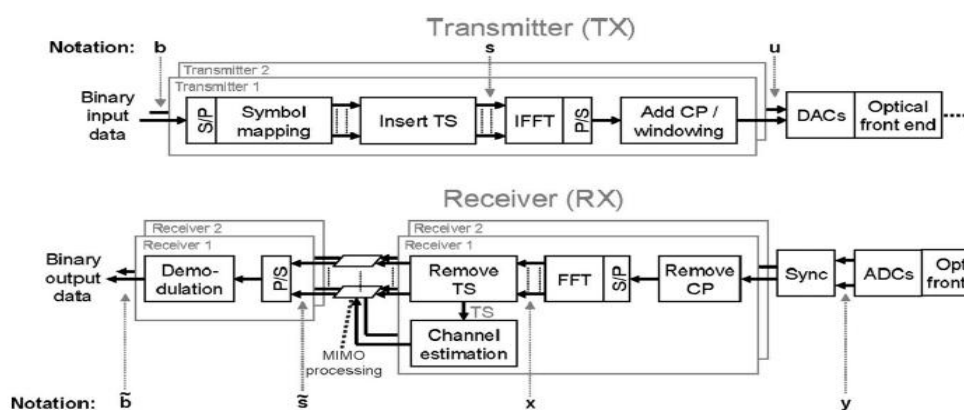


Figura 13. Esquema TITO amb recuperació de PMD.

La següent imatge mostra el diagrama de blocs d'un transmissor receptor PDM-OFDM i on disposem d'un PBC que uneix els fluxos dels dos transmissors i un PBS, a l'entrada del receptor que divideix el flux de dades de l'enllaç òptic en dos fluxos diferents.

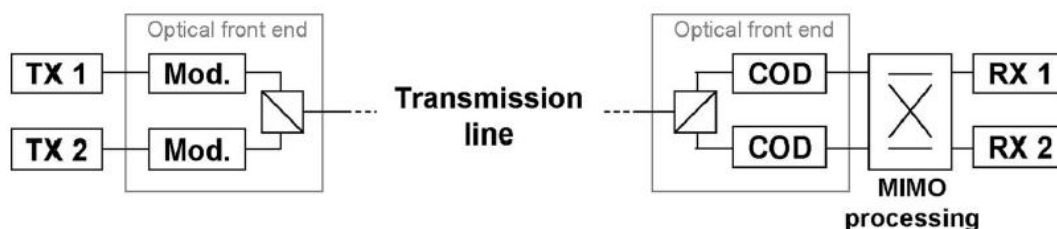


Figura 14. Diagrama de blocs TITO.

El funcionament del sistema és molt semblant al sistema 1x2 MIMO CO-OFDM vist abans. Les principals diferències estan:

- El transmissor OFDM consisteix de dos transmissors convencionals OFDM on s'introdueixen les dades.
- Disposem d'un bloc PBC a la sortida del transmissor que uneix els dos fluxos de dades, matemàticament les dades inicials serien un vector de dimensió 2 mentre que abans era un escalar. Per tant, les dades binàries es poden agrupar en un vector

$$\vec{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \text{ on cada índex representa un transmissor}$$

- La matriu de Jones H ara és de dimensió 2x2, ja que tenim dos transmissors i dos receptors:

$$\vec{H}(k) = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix}$$

L'estimació del canal es realitza mitjançant els TS com a l'esquema SITO. Però en aquest cas, és necessari que els TS de cada transmissor siguin ortogonals entre ells per a que es puguin ser detectats unívocament al receptor com es pot veure a la figura 15.

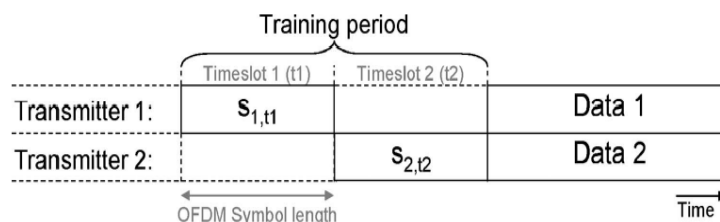


Figura 15. Multiplexació en temps dels TS per l'estimació de canal MIMO.

2.9. Fibra òptica:

2.9.1 Valors inicials de la fibra

Els valors utilitzats per fer l'estudi s'han basat en els que tenen les fibres utilitzades habitualment en aquest tipus d'aplicacions de llargues distàncies. A la web es poden trobar diferents fitxes tècniques dels fabricants ([Corning](#), [Prysmian](#), etc.) amb valors molt similars.

Tipus de fibra	Monomode
Finestra	3ª finestra
Longitud d'ona	1550nm
Atenuació	0.2db/m
Dispersió cromàtica	$16,5 \cdot 10^{-6} \text{ s/m}^2$
PMD	0.04 ps/ $\sqrt{\text{km}}$
Slope	0.057 s/m^3
Index no linealitat	$2.4 \cdot 10^{-20} \text{ m}^2/\text{w}$
Àrea efectiva	$80 \cdot 10^{-12} \text{ m}^2$
Birefringència	10^{-7}

Figura 16. Taula amb els diferents valors de la fibra.

Altres especificacions que s'han tingut en compte en el disseny són:

- No hi ha pèrdues d'inserció, PDL.
- No hi ha no linealitats.
- No hi ha soroll de fase dels làsers.

2.9.2. Atenuació.

L'atenuació de la fibra pot ser descrita en general amb la relació:

$$\frac{dP}{dz} = -\alpha P$$

on α és el coeficient d'atenuació de potència per unitat de longitud. Si la P_{in} és la potència injectada a la fibra, la potència que arriba al receptor després de passar per la fibra de longitud L serà:

$$P_{out} = P_{in} e^{-\alpha L}$$

2.9.3. Dispersió cromàtica:

En els sistemes que utilitzen fibra monomode, com és el nostre cas, la dispersió multimode no està present ja que només es transmet un mode a través de la fibra, no obstant això, la transmissió també es degrada per culpa de la dispersió cromàtica.

Aquesta dispersió es deu al fet que la font òptica no és monocromàtica, és a dir, emet més d'una longitud d'ona i cada freqüència que forma el senyal viatja a una velocitat diferent al llarg de la fibra, degradant el senyal que rebem. Quan els polsos veïns es transmeten tindrem ISI i la velocitat de transmissió del senyal i la distància a la que podem arribar es veuran molt limitades.

Cada component espectral arribarà al receptor amb un retard τ_g conegut com retard de grup:

$$\tau_g = \frac{L}{v_g} = L \frac{d\beta}{d\omega} = \frac{L}{c} \frac{d\beta}{dk} = -L \frac{\lambda^2}{2\pi c} \frac{d\beta}{d\lambda}$$

on β és la constant de propagació introduïda prèviament com :

$$v_g = \left(\frac{d\beta}{d\omega} \right)^{-1}$$

i v_g és la velocitat de grup amb la qual l'energia del pols òptic viatja per la fibra.

La constant de propagació β és una funció de la freqüència òptica i els component espectral del senyal es propaguen, per tant, a diferents velocitats provocant l'eixamplament temporal del pols que hem dit abans. Podem analitzar la constant de propagació β entorn a ω_0 fent una aproximació en sèrie de Taylor:

$$\beta(\omega) = n(\omega) \frac{\omega}{c} = \beta_0 + \beta_1(\omega - \omega_0) + \frac{1}{2} \beta_2(\omega - \omega_0)^2 + \dots$$

on podem resumir amb:

$$\beta_m = \left(\frac{d^m \beta}{d\omega^m} \right)_{\omega=\omega_0}$$

i $m=0,1,2,..$

Com a resultat de la diferència en retards de temps, el pols òptic es dispersarà després de viatjar una determinada distància i l'eixamplament del pols es caracteritza:

$$\Delta\tau_g = \frac{d\tau_g}{dw} \Delta w = \frac{d\tau_g}{d\lambda} \Delta\lambda$$

on Δw representa l'ample de banda de la freqüència de la font i $\Delta\lambda$ és l'ample de les longituds d'ona de la font. Substituint les equacions anteriors tenim:

$$\Delta\tau_g = LD\Delta\lambda$$

Per altra banda, els paràmetres β_1 i β_2 es relacionen amb l'índex de refracció a partir de les següents relacions:

$$\beta_1 = \frac{1}{v_g} = \frac{n_g}{c} = \frac{1}{c} \left(n + w \frac{dn}{dw} \right)$$

$$\beta_2 = \frac{1}{c} \left(2 \frac{dn}{dw} + w \frac{d^2n}{dw^2} \right)$$

on n_g és l'índex de refracció de grup. El paràmetre β_2 representa la dispersió cromàtica, també coneguda com a dispersió de velocitat de grup (GVD), i és la responsable, en primer ordre, de l'allargament del pols.

Normalment a la literatura s'utilitza el paràmetre de dispersió D on:

$$D = \frac{d\beta_1}{d\lambda} = -\frac{2\pi c}{\lambda^2} \beta_2 \approx \frac{\lambda}{c} \frac{d^2n}{d\lambda^2}$$

El paràmetre D representa la dispersió cromàtica total en una fibra monomode convencional, aquesta dispersió es podria descompondre en dos components: la dispersió provocada pel material i la dispersió causada per la guia d'ona.

La dispersió cromàtica se sol mesurar amb el paràmetre D en [ps/nm/km] que està relacionat amb la dispersió cromàtica β_2 mitjançant l'equació anterior.

També hem de tenir en compte la dispersió residual deguda als ordres de dispersió superiors que encara existeixen. Els efectes de la dispersió d'ordres elevats estan regits pel paràmetre de dispersió slope S :

$$S = \frac{dD}{d\lambda}$$

Aquest valor S també es conegut com paràmetre de dispersió diferencial o de dispersió de segon ordre i està relacionat amb el paràmetre de segon ordre:

$$S = \frac{dD}{d\lambda} = \frac{d}{d\lambda} \left(-\frac{2\pi c}{\lambda^2} \beta_2 \right) = \frac{4\pi c}{\lambda^3} \beta_2 + \left(\frac{2\pi c}{\lambda^2} \right)^2 \beta_3$$

2.9.4. Dispersió de mode de polarització

Ja l'hem comentada al [primer punt](#) d'aquest capítol.

2.10. SSPROP:

El software [SSPROP](#) és un simulador que resol l'equació no lineal d'Schrödinger per a la propagació de senyals òptics en fibres dispersives i amb no linealitats. Ha estat creat per la [Universitat de Maryland \(USA\)](#).

SSPROP és un codi obert realitzat en MATLAB i que pot modelar diferents efectes de la fibra, tal com la birefringència, la dispersió de polarització, el retard de grup, la pèrdua de polarització PDL, etc.

També disposa d'una versió compilada en C que pot ser cridada directament des de MATLAB i que millora l'eficiència i la velocitat de la versió inicial, un aspecte important ja que el programa ha de calcular un munt de transformades de Fourier per simular el comportament de la fibra.

Existeixen dues versions del software : escalar i vectorial. A nosaltres ens interessa aquesta última ja que és la que permet treballar amb fibres amb birefringència. La sintaxis de l'ordre per cridar a SSPROP des del codi MATLAB és:

```
"u1=sspropv(uox,uoy,dt,dz,nz,alphaa,alphab,betapa,betapb,gamma,psp,method,maxiter, tol);"
```

On cada entrada representa :

- **uox,uoy**- Són els components de l'eix X i eix Y del vector d'entrada del senyal a transmetre. Tenen com a longitud el nombre de símbols d'informació.
- **dt**- L' increment de temps entre punts adjacents, nosaltres hem fet servir la inversa de la freqüència.
- **dz**- És la dimensió de cada pas per fer la propagació.
- **nz**- És el nombre passos utilitzats sent la distància total $L=nz \cdot dz$.
- **alphaa,alphab**- Són els coeficients en lineal de l'atenuació per a cada eix. Si volem modelar la PDL, hauríem de posar valors diferents en cadascun. Com hem vist abans, no és el cas i posarem el mateix valor a totes dues variables.
- **betapa, betapb**- Especifiquen la dispersió en cada eix de la fibra, cada variable serà un vector format pels diferents coeficients de β : $[\beta_0 \beta_1 \beta_2 \beta_3]$. La birefringència s'introdueix fent que els valors de β_0 siguin diferents per a cada variable. El DGD o PMD es realitza igual però fent diferents els valors de β_1 . Els valors de β_2 i β_3 es calculen a partir de les fórmules de D i S respectivament, vistes en l'apartat de [dispersió cromàtica](#).
- **gamma**- És el coeficient no lineal de la fibra que està relacionat amb l'àrea efectiva i l'índex de refracció no lineal n_2 .
- **psp**- Especifica l'orientació que té la birefringència respecte als eixos X i Y. Es defineix per $[\Psi, \chi]$, segons els valors de Ψ i χ disposarem d'una polarització lineal, circular o el·líptica.
- **method**- Ens indica quin mètode s'utilitza per realitzar els càlculs de la divisió per parts. És pot realitzar mitjançant el mètode el·líptic que és l'adient quan les fibres tenen per a la birefringència un "beat length" més curt que la longitud no lineal L_{NL} ($L_B < L_{NL}$) o mitjançant el mètode circular quan és al contrari ($L_B > L_{NL}$). El nostre cas és el primer, el mètode el·líptic.

- **maxiter**- El nombre màxim d'iteracions a fer per pas, per defecte és 4.
- **tol**- És la tolerància de la convergència, controla que el nivell de la solució convergeixi, per defecte és 10^{-5} .

2.10.1 Valors de la dispersió:

- Càlcul de β_0 :

Al llibre "Nonlinear Fiber Optics" d' Agrawal, G.P., pàgina 207, tenim l'equació:

$$\Delta\beta = \beta_{0x} - \beta_{0y} = \frac{2\pi}{\lambda} B_m$$

Per introduir-lo a SSPROP hem de tenir en compte:

$$\text{betapa}(1) - \text{betapb}(1) = \Delta\beta_0$$

i

$$\text{betapa}(1) = -\text{betapb}(1)$$

llavors

$$\Delta\beta_0 = 2\text{betapa}(1)$$

$$\Delta\beta_0 = \frac{2\pi}{\lambda} B_m = 2\text{betapa}(1)$$

$$\text{betapa}(1) = \frac{\pi}{\lambda} B_m$$

- Càlcul de β_1 :

Com :

$$\Delta\beta_1 = \frac{G}{c}$$

on G és la birefringència de grup.

I per introduir-lo a SSPROP sabem que:

$$\text{betapa}(2) - \text{betapb}(2) = \Delta\beta_1$$

i

$$\text{betapa}(2) = -\text{betapb}(2)$$

llavors

$$\Delta\beta_1 = 2\text{betapa}(2)$$

$$\beta_2 = \frac{G}{2c}$$

- Càlcul de β_2 :

Es calcula a partir de l'equació:

$$D = \frac{d\beta_1}{d\lambda} = -\frac{2\pi c}{\lambda^2} \beta_2$$

per tant:

$$\beta_2 = -\frac{D\lambda^2}{2\pi c}$$

- Càlcul de β_3 :

Es calcula a partir de l'equació:

$$S = \frac{dD}{d\lambda} = \frac{d}{d\lambda} \left(-\frac{2\pi c}{\lambda^2} \beta_2 \right) = \frac{4\pi c}{\lambda^3} \beta_2 + \left(\frac{2\pi c}{\lambda^2} \right)^2 \beta_3$$

per tant:

$$\beta_3 = \frac{s\lambda^4}{4\pi^2 c^2} - \frac{\lambda}{\pi c} \beta_2$$

2.10.2 Valors de dz:

Cada valor de dz es calcula a partir de:

$$dz = 0.05 \left(\frac{\pi}{180} \right) L_{NL}$$

on L_{NL} és:

$$L_{NL} = \frac{1}{\gamma P_0}$$

2.10.3 Valors de gamma:

Com hem comentat abans l'entrada gamma indica el coeficient no lineal de la fibra, es pot calcular amb:

$$\gamma = \frac{2\pi n_2}{A_{eff} \lambda} = 0,0012$$

2.10.4. Valors de psp:

Normalment els eixos de la birefringència de la fibra no coincideixen amb els eixos cartesianes X i Y, ja que poden estar girats un angle Ψ qualsevol en relació a l'eix d'abscisses. Altre factor a tenir en compte és que els dos eixos principals de polarització de fibra no sempre corresponen a polaritzacions lineals, ja que podem tenir polaritzacions circulars o el·líptiques.

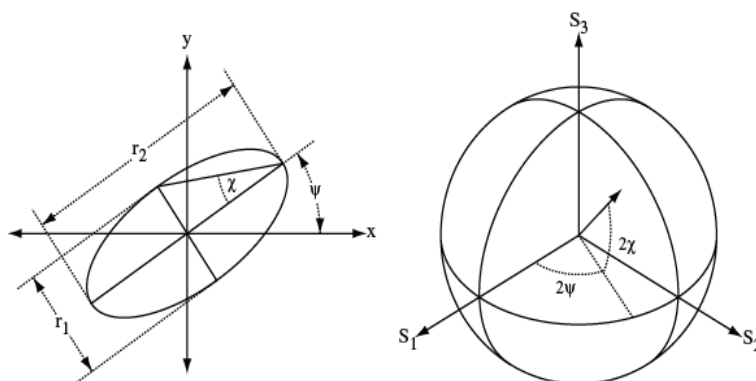


Figura 17. Polarització en 2D i en esfera de Poincaré.

Per treballar amb polaritzacions lineals només hem de mantenir χ a zero i modificar l'angle de Ψ , per exemple una polarització lineal a 45° serà:

$$psp = \left[\frac{\pi}{4}, 0 \right]$$

Una polarització circular, ha de mantenir χ a 45° , per tant, una polarització circular seria:

$$psp = \left[\frac{\pi}{4}, \frac{\pi}{4} \right]$$

Per última una polarització el·líptica la tindrem quan χ sigui diferent de 0 i 45°

$$psp = \left[\frac{\pi}{4}, \frac{\pi}{3} \right]$$

2.10.5. Valors de method:

Com hem dit abans, el mètode triat és l'el·líptic perquè el "beat length" de la birefringència és més curt que la longitud no lineal L_{NL}

$$L_B = \frac{2\pi}{\Delta\beta} = \frac{\lambda}{\Delta n} = 15,5m$$

L_{NL} es pot calcular amb el coeficient de no linealitat γ i la potència de pic P_0

$$L_{NL} = \frac{1}{\gamma P_0} = 134,41km$$

Per tant, és evident que:

$$L_B < L_{NL}$$

3. Implementació codi MATLAB.

3.1. CO-OFDM amb SITO (1x2 MIMO OFDM):

3.1.1 Transmissor:

- **Generació de dades:**

En primer lloc hem de generar les dades i els TS ("Training Symbol") que ens permetran, més tard, estimar la resposta del canal en el receptor. Hem decidit treballar amb QPSK (2 bits/símbol) i assignar 128K símbols a la informació i 1 símbol a TS.

Per tant, en bits tindrem 256kbits de dades i dos bits per cada TS. En el codi, introduïm l'ordre "RandStream" per a que la generació pseudo-aleatòria sigui sempre la mateixa i poder treballar amb els mateixos valors de dades i TS en tot moment.

Codi:

```
s = RandStream('mt19937ar','Seed',1);
RandStream.setGlobalStream(s);

% 1) Generar les dades de "payload"

data=randi([0 1],1,Nsymbol*bps);

% 2) Generar les dades de TS
TS=randi([0 1],1,Ncarriers*bps);
TS1=(repmat(TS,1,NTS));
```

A la variable "data" tenim les dades d'informació, un array de dimensió 1x256k, on els 256k els obtenim de multiplicar el nombre de símbols (Nsymbol=128K) pel nombre de bits de la modulació QPSK (bps=2bits/símbol)

A la variable "TS" tenim un array de dimensió 1x128, en aquest cas els 128 provenen de multiplicar el nombre de subportadores (Ncarriers=64) pel nombre de bits de la modulació QPSK (bps=2bits/símbol). Així, tenim un conjunt total de 64TS per a les 64 subportadores de la modulació OFM.

Per últim, amb la sentència "repmat" guardem a "TS1" un vector que és l'array "TS" repetit tantes vegades com nombre de "Training symbol" volem que hi hagi per subportadora. Amb aquesta acció podrem introduir de forma periòdica la quantitat de TS que vulguem entre els símbols d'informació.

- **Mapejar dades i TS:**

Transformem tant les dades com els TS a QPSK,. El MATLAB disposa d'ordres específiques per realitzar-ho. El modulador QPSK agafa els bits en sèrie que li arriben de dos en

dos i genera un símbol. Guardem els símbols de dades a la variable "símbol" i els de TS a la variable "símbolTS".

Codi:

```
% 3) Mapejar les dades resultants amb QPSK
hModulator = comm.QPSKModulator('BitInput',true);
símbol = step(hModulator, data');
símbolTS = step(hModulator, TS1');
```

Una codificació possible seria:

Bits de dades	Símbol QPSK
0 0	-1-j
0 1	1-j
1 0	-1+j
1 1	1+j

Figura 18. Modulació QPSK

- **Paral·lelitzar i introduir els TS de forma periòdica**

Fins aquí tenim només dos vectors amb les dades i els TS mapejats en QPSK, el que necessitem ara és paral·lelitzar les dades en grups de 64 subportadores i introduir de forma periòdica el nombre de TS que hem definit per subportadora. Ens hauria de quedar una cosa semblant a la figura següent.

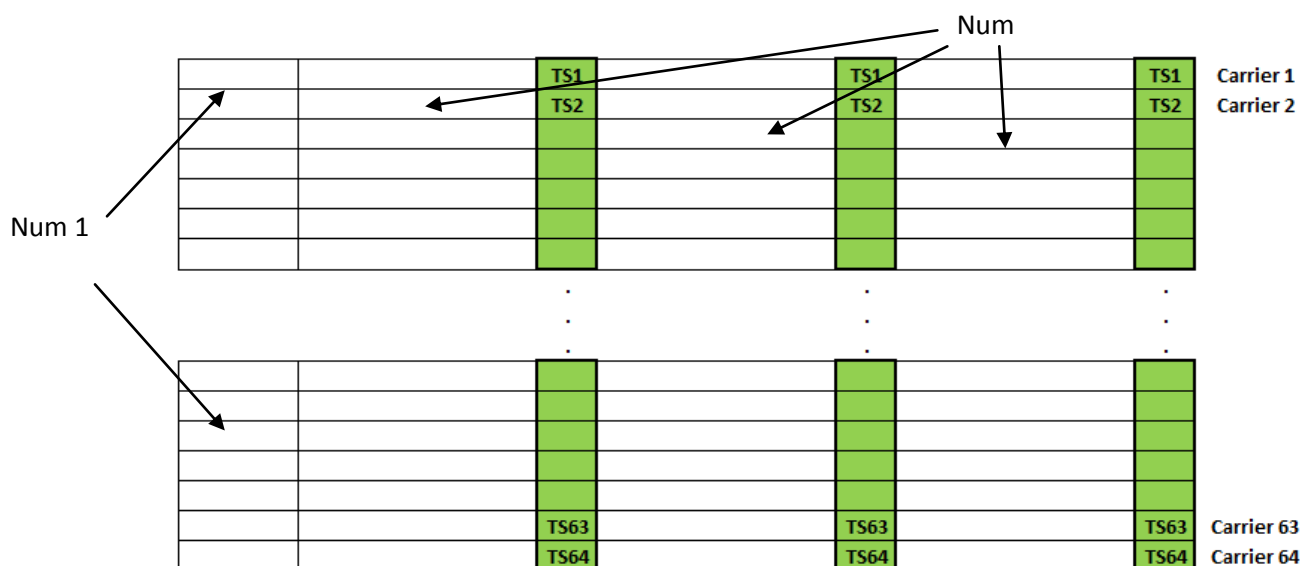


Figura 19. Estructura dades paral·lelitzades abans d'entrar al bloc IFFT.

Per aconseguir-ho, definim la divisió $\frac{N_{símbol}}{N_{carriers} * N_{TS}}$ que ens indica quants símbols d'informació hi ha entre dos TS seguits per subportadora (variable "Num"). Si la divisió no és exacta, guardem el mòdul de la divisió en una nova variable (Num1) que ens indicarà quants símbols hem d'afegir a la cua de la matriu. Amb l'ordre "reshape" modifiquem les dimensions

tant del vector "simbol" com de "simbolTS", unim les dues matrius resultants (c i d) en una nova (c1) i a aquesta la transformem en una matriu de Ncarriers=64 columnes per aconseguir el format de la figura 19. Si Num1 és diferent de zero afegirem aquests símbols a la cua de la matriu i guardem la matriu resultant a la variable "cdatar".

Codi:

```
% 4) Paralelitzar i introduir TS de forma periòdica
Num=fix(Nsymbol/(Ncarriers*NTS)); % Quocient divisió
Num1=mod(Nsymbol,(Ncarriers*NTS)); % Mòdul divisió
c=reshape(simbol(1:end-Num1),Ncarriers*Num,[]);
d=reshape(simbolTS,Ncarriers,[]);
c1=[d;c];
cdatar=transpose(reshape(c1,Ncarriers,[]));
if Num1~=0 % Afegim símbols "lliures"
    b1=transpose(reshape(simbol(end-Num1+1:end),Ncarriers,[]));
    cdatar=[cdatar;b1];
end
```

- **Algoritme IFFT, afegir CP:**

En primer lloc introduïm els símbols al bloc IFFT. MATLAB disposa de l'ordre "ifft" per realitzar el canvi del domini freqüencial al temporal i, a continuació, afegim el prefix cíclic CP.

Hem assignat un CP de 0.25. Per entendre més fàcilment quines accions realitza el codi hem dibuixat la següent figura, en primer lloc podem veure com tenim agrupats els símbols de dades i TS en files de 64 subportadores, per tant un període OFDM estarà format per 64 símbols de cada columna.

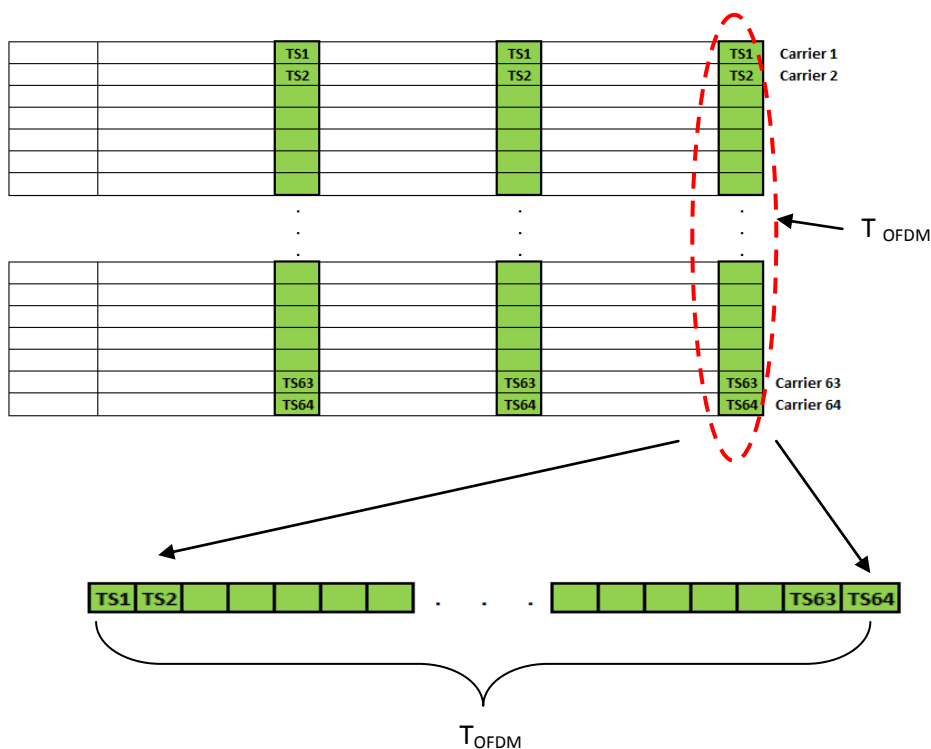


Figura 20. Símbols OFDM i T_{OFDM}

Si volem afegir un prefix cíclic de 0.25, haurem d'afegir a la matriu $0.25 \cdot N_{\text{carriers}}$ al T_{OFDM} , que en aquest cas són 16 valors ($0.25 \cdot 64$)

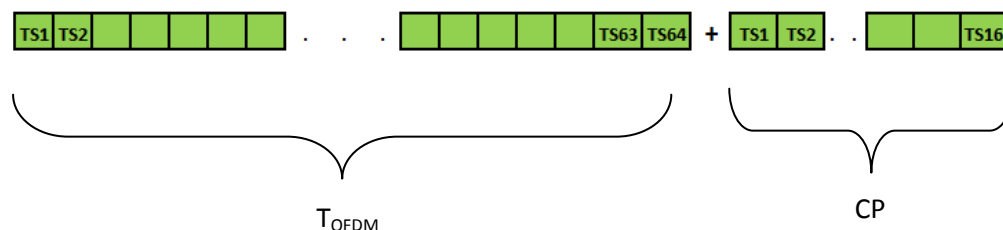


Figura 21. $T_{\text{OFDM}} + \text{CP}$

Codi:

```
% 6) Fer la transformada inversa
fftdata=ifft(cdatar,Ncarriers,2);

% 7) Afegir el "cyclic prefix"
fftdata_cp=[fftdata fftdata(:,1:round(CP*Ncarriers))];

% 8) Serialitzar
Cx=reshape(transpose(fftdata_cp), [], 1);
```

Per últim, el que hem fet és modificar la dimensió de la matriu "fftdata_cp" i serialitzar tota la informació en la variable "Cx" per tenir un array amb tot els símbols en sèrie.

- **Clipping:**

En aquest pas el que fem és implementar el clipping per reduir el PAPR, per aconseguir-ho, calculem la desviació estàndard de la variable "Cx" i eliminem els valors superiors a k vegades la desviació per dalt i per baix. En aquest cas, hem utilitzat un valor de $k=3$.

Codi:

```
% 9) Clipping
deviation=std(Cx);

%Per dalt
index=Cx>=k*deviation;
Cx_c=Cx;
Cx_c(index)=k*deviation;
clear index;

%Per baix
index=find(Cx<=-k*deviation);
Cx_c(index)=-k*deviation;
```

- **Sobremostrejar i filtrar:**

Com que el retard, degut a la PMD de primer ordre, pot estar per sota del temps de símbol QPSK, és necessari sobremostrejar el senyal per a poder tenir retards que estiguin per sota d'aquest temps de símbol.

Amb l'ordre "upsample" augmentarem el nombre de mostres per símbol, en aquest cas ho farem per un factor 8. A continuació, filtrarem el senyal per no ocupar tot l'espectre, per fer-lo treballarem amb un filtre FIR de cosinus alçat (raised cosine) que permet utilitzar l'ample de banda disponible d'una forma més eficient. Aquests filtres depenen d'un paràmetre anomenat de roll-off que varia entre 0 i 1.

Codi:

```
M=8; %Factor oversampling

%Valors filtre
N=34; %Ordre del filtre
fc=BitRate/2; %frequència de tall
r=0.1; %roll-off
fm=M*2*fc; %frequència de mostratge

h1 = firrcos(N,fc,r,fm,'rolloff','sqrt');

% 10)Upsampling i filtratge
Cx_cm=upsample(Cx_c,M); % Augmentem a 8 mostres/símbol
Cx_cm_aux=[Cx_cm; zeros(N/2,1)]; % Afegir zeros al final
Cx_cm_auxf=filter(h1,1,Cx_cm_aux); % Filtrar
Cx_cmf=Cx_cm_auxf(N/2+1:end);
```

Un aspecte a tenir en compte és el retard que els filtres FIR afegixen al senyal filtrat, aquest retard és sempre de $N/2$, on N és l'ordre del filtre. Per eliminar aquest inconvenient, afegirem al final del senyal, abans de filtrar, tants zeros com retard tenim i, després de filtrar, eliminarem els primers $N/2$ coeficients del nou senyal. Finalment, guardarem el vector a "Cx_cmf".

És interessant observar que segons els valors del filtre FIR que s'utilitzin, la resposta del sistema pot ser molt diferent. Hem de cercar aquells paràmetres que ofereixin un resultat més eficient.

- **Divisor de polarització:**

Com en aquest treball no implementem la conversió optoelectrònica, les amplituds complexes del senyal corresponen a les del camp òptic i, per tant, tindrem el senyal òptic descompost sobre l'eix X i l'eix Y a l'entrada de la fibra. Això es pot fer amb una multiplicació del nostre senyal òptic amb un vector de Jones[1, 1];

Codi:

```
%) 11)Divisor de polarització:
p=[1 1];
j2=Cx_cmf*p;
```

3.1.2 Fibra òptica:

Com hem vist en el capítol 1, a l'apartat on explicàvem el programa [SSPROP](#), simularem el comportament de la fibra amb aquest software. En primer lloc, donarem els valors necessaris a totes les entrades que necessita l'script i que hem definit anteriorment.

Entre els valors més destacats trobaríem la definició dels vectors de dispersió (betap1 i betap2), l'alineació del vector de Jones de la fibra amb el senyal òptic (psp) i la dimensió dels passos per simular l'equació d'Schrödinger.

Codi:

```
% Càlcul dels coeficients de dispersió
beta2=-((Dp*(lambda0^2))/(2*pi*c));
beta3=((S*lambda0^4)/(4*(pi^2)*(c^2)))-beta2*lambda0/(pi*c);

%Vector de Jones fibra
psp=[0 0]; % Alineació amb el vector de Jones del senyal

n2=2.4e-20; %[m^2/W] index de no-linialitat
Aeff=80e-12; %[m^2] area efectiva
gamma= 2*pi*n2/(Aeff*lambda0); % Coeficient de no linialitat

fiber_length=100e3; % [m] Longitud de la fibra
max_deltaz=1e3;% [m] Màxima longitud de pas per al Split-Step
max_phase_change=.05; % [degrees] Màxim canvi de fase

beta0=pi*birref/lambda0; %
beta1=birref/(2*c);
betap1=[beta0, beta1, beta2, beta3]; % array de dispersió
betap2=[-beta0, -beta1,beta2, beta3]; % array de dispersió

% Definim l'increment i el número de passos

deltaz=max_phase_change*(pi/180)/(gamma*max(abs(j2(:,1)).^2+abs(j2(:,2)).^2));

if deltax>max_deltax
    deltax=max_deltax;
end

step_num=fiber_length/deltax;
```

A continuació, cridem a l'script "sspropv", si volem posar longitud de fibres superiors a 150km podem fer servir el següent bucle "for", es repetirà com a màxim $i=(\text{Length_fiber}/150)$ vegades:

Codi:

```
% Entrem a la fibra
j1ss=j2;
G=31.66;

for i=1:3
[j21a, j22b]= sspropv(j1ss(:,1), j1ss(:,2), 1/fm,deltax,step_num,
alpha, alpha, betap1, betap2, gamma, psp, 'elliptical', 100, 1e-5);
j1ss=G*[j21a, j22b];
end;

j2ss=[j21a j22b].';
```


3.1.3 Mòdul de potència:

El nostre objectiu es conèixer la resposta del sistema per a diferents valors de SNR, hem definit un interval d'estudi d'aquest paràmetre entre 0 i 13db. Per a cada valor de SNR, l'ordre de MATLAB "awgn" introduirà en el senyal un soroll blanc que representarà el soroll que introdueix el canal de fibra òptica.

Per veure la resposta de tots els valors de SNR farem servir un bucle "for" que ens calculi les diferents BER que obtindrem per a cadascun d'ells. Més tard representarem gràficament la seva relació i la compararem amb la resposta ideal.

Cal tenir en compte que estem transmetent un senyal d'una determinada potència per dos components de polarització ortogonals, això significa que tenim una redundància d'aquest senyal en els dos eixos.

Per tant, al passar pel mòdul "awgn", hem de proporcionar només la meitat de la SNR especificada a cada component, per aconseguir-ho restem a la SNR donada uns 3dB que, en lineal, representen la meitat de potència total.

És interessant observar que per aplicar correctament el soroll en tots dos components de "j2ss", és preferible treballar amb "awgn" de forma matricial i no component a component, ja que provocaria errors.

Codi:

```
for ksnr = 1:1:length(SNR),  
  
    j3_mf=awgn(j2ss, SNR(ksnr)-3, 'measured'); % Els 3 dB son pel fet  
de transmetre en nomes 1 comp. de la pol;  
    ...  
    ...  
    ...  
    [Nerr, BER(ksnr)]=biterr(data, rec');  
  
end
```

3.1.4 Receptor:

- **Filtratge i delmació:**

Una vegada el senyal ha arribat al receptor, hem de fer l'inversa del sobremostreig i filtratge que havíem realitzat abans d'enviar el senyal per la fibra. Per tant, utilitzarem el mateix filtre FIR i els mateixos paràmetres que havíem aplicat abans i, després, farem una delmació utilitzant l'ordre "downsample" de MATLAB. El factor de la delmació és 8.

Com en el cas anterior, hem de tenir en compte el retard que el filtre afegeix i eliminar-lo amb el [mètode explicat anteriorment](#). Per últim les dades rebudes es guardaren a la matriu "j3".

Codi:

```
% Filtratge i downsampling
j3_mf_aux=[j3_mf zeros(2, N/2)]; % afegir zeros al final
j3_mf_auxf=filter(h1,1,j3_mf_aux); % filtrar
j3_m=j3_mf_auxf(:,N/2+1:end);

j3=downsample(j3_m.',M); %Delmació 1/8
```

- **Paral·lelitzar , treure CP i calcular la FFT**

Tenir un receptor amb diversitat és com disposar, per aquest sistema, de dos receptors que tracten de forma independent cada component de polarització. En el nostre cas cada component o PSP està emmagatzemada en cada una de les files de la matriu "j3".

Primer de tot, volem tornar a paral·lelitzar els valors en sèrie de cada component per a poder enviar el senyal al bloc "fft", això ho aconseguirem amb l'ordre "reshape" i traient el CP que havíem afegit al transmissor.

Per últim, passarem del domini temporal al freqüencial amb "fft" i guardarem els valors obtinguts de cada component a les matrius "fftdatarx1" i "fftdatarx2".

Codi:

```
% 1) paral·lelitzar
rxdata_cp1=transpose(reshape(j3(:,1), round((1+CP)*Ncarriers), []));
rxdata_cp2=transpose(reshape(j3(:,2), round((1+CP)*Ncarriers), []));

% 2) Treure el cyclic prefix
rxdata1=rxdata_cp1(:, 1:Ncarriers);
rxdata2=rxdata_cp2(:, 1:Ncarriers);

% 3) calcular la FFT
fftdatarx1=fft(rxdata1,Ncarriers, 2);
fftdatarx2=fft(rxdata2,Ncarriers, 2);
```

- **Equalitzar**

Aquest és el punt més important del receptor, on estimem la resposta del canal i equalitzem el senyal rebut. Primer de tot el que fem es recuperar els valors inicials, abans de transmetre, dels TS i els guardem a la variable "TSr". Això és pot fer perquè, a priori, el receptor coneix els valors de TS que s'enviaran.

A continuació, traiem de les matrius "fftdatarx1" i "fftdatarx2", on estan les dades rebudes, els valors dels TS que s'han rebut i que han sofert la distorsió i el soroll del canal. Els valors de TS del component de l'eix X es guarden a "TSdata1" i els de l'eix Y es guarden a "TSdata2".

El pas següent és , per a cada subportadora, fer la mitja dels valors de TS rebuts i guardar-los a les matrius "Hs1" i "Hs2".

Per fer l'estimació del canal, calculem, també per a cada subportadora, els diferents coeficients de la matriu z, en aquest cas, com el sistema està format per un transmissor i dos receptors, les dimensions de la matriu són 2x1.

$$z = \begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix}$$

on

$$h_{11} = \frac{Hs1(k)}{TSr(k)}$$

i

$$h_{21} = \frac{Hs2(k)}{TSr(k)}$$

Després, per aplicar l'algoritme ZF, fem la pseudoinversa de la matriu z. MATLAB realitza aquesta operació amb una sola ordre "pinv" i guardem el seu resultat a la matriu b.

Per últim, obtindrem el senyal estimat al multiplicar aquesta matriu "b" per una altra formada per dues columnes on cadascuna representa cada component del senyal rebut. Hem de recordar que tot aquests càlculs es fan per cada subportadora, és a dir, per les 64.

Codi:

```
% 4) Equalitzar
TSr=cdata(1,:);
r1=fftdatarx1(1:end-Num1/Ncarriers,:);
r2=fftdatarx2(1:end-Num1/Ncarriers,:);
TSdata1=reshape(r1.',[],NTS);
TSdata2=reshape(r2.',[],NTS);
Hs1=mean(TSdata1(1:Ncarriers,:),2); % Fem la mitja dels símbols
d'entrenament rebuts
Hs2=mean(TSdata2(1:Ncarriers,:),2); % Fem la mitja dels símbols
d'entrenament rebuts

for k=1:Ncarriers %Calculem la pseudomatriu per a cada subportadora
    h11=Hs1(k)/TSr(k);
    h21=Hs2(k)/TSr(k);
    z=[h11; h21];
    b=pinv(z);
    t(:,k)=b*[fftdatarx1(:,k) fftdatarx2(:,k)].';%Estimació de coef.
end
```

- **Demapejar QPSK**

Una vegada tenim el senyal estimat, només ens queda transformar els símbols QPSK en bits i serialitzar les dades obtingudes .

Codi:

```
% 11) demapejar QPSK

r2=reshape(transpose(t(1:end-Num1/Ncarriers,:),[]),[],NTS);
rdata=reshape(r2(Ncarriers+1:end,:),[],1); %Treiem TS
if Num1~=0
    r4=t(end-(Num1/Ncarriers)+1:end,:);
    r5=reshape(r4.',[],1);
    rdata=[rdata;r5];
end;
hDemod = comm.QPSKDemodulator('BitOutput',true);
rec = step(hDemod, rdata);
```

• **Gràfics:**

Per últim, comparem els valors obtinguts de la BER per a cada SNR amb els valors ideals de la BER per a QPSK. La BER teòrica de QPSK és:

$$BER = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \left[2 - Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \right]$$

Al gràfic imposem que l'eix Y sigui logarítmic i posem les dues corbes a la mateixa gràfica per veure les diferències entre la corba ideal i la simulada.

Codi:

```
% 13) Fer la gràfica de BER vs SNR
ebno=SNR-10*log10(bps); % Calcul de la Eb/No d'acord amb doc. Matlab
AWGN
ebnolin=10.^(ebno/10); % Passa Eb/No a linial
BERteorica=(1/bps)*erfc(sqrt(ebnolin)).*(1-.25*erfc(sqrt(ebnolin))); %
Formula teorica (Proakis/Salehi 5a ed; cap. 4, pags. 193/195)

semilogy(SNR,BER,SNR,BERteorica);

xlabel('SNR (dB)');
ylabel('BER');
grid on;
legend('Simulacions ZF','Teoria');
title('Comparació BER teòrica vs BER');
```

3.2. CO-OFDM amb TITO (2x2 MIMO OFDM):

El codi per realitzar el PDM és molt semblant al fet servir en el sistema SITO 1x2 MIMO OFDM. Les diferències principals són:

- Disposem de dos transmissors per a cada senyal en comptes d'un.
- Multiplexació en temps dels TS de cada senyal.
- L'ordre de la matriu d'estimació és 2x2.
- La potència no es reparteix entre les dues components com en el cas anterior.

3.2.1 Transmissor:

- **Generació, mapeig, IFFT, CP, serialitzar i clipping.**

Repetim els mateixos procediments que en el cas anterior, simplement que ara ho fem dues vegades ja que tenim dos senyals diferents de dos transmissors. La principal diferència al transmissor és com tractem els TS.

Codi:

```
% 1) Generar les dades de "payload"
data(:,1)=randi([0 1],1,Nsymbol*bps);
data(:,2)=randi([0 1],1,Nsymbol*bps);

% 2) Generació de TS
TS=randi([0 1],1,Ncarriers*bps);
TS1= repmat(TS,1,NTS);

% 3) Mapejar les dades resultants amb la constel·lació que toca
hModulator = comm.QPSKModulator('BitInput',true);
simbol1 = step(hModulator, data(:,1));
simbol2 = step(hModulator, data(:,2));
simbolTS1 = step(hModulator, TS1);
simbolTS2=simbolTS1;

% 4) Paral·lelitzar i introduir TS de forma periòdica
Num=fix(Nsymbol/(Ncarriers*NTS)); %Quocient divisió
Num1=mod(Nsymbol,(Ncarriers*NTS)); %Mòdul divisió
c1=reshape(simbol1(1:end-Num1),Ncarriers*Num,[]);
c2=reshape(simbol2(1:end-Num1),Ncarriers*Num,[]);

d1=reshape(simbolTS1,Ncarriers,[]);
d2=reshape(simbolTS2,Ncarriers,[]);

c11=[d1;c1];
c22=[d2;c2];
cdatar1=transpose(reshape(c11,Ncarriers,[]));
cdatar2=transpose(reshape(c22,Ncarriers,[]));

if Num1~=0 %Afegim símbols "lliures"
    b1=transpose(reshape(simbol1(end-Num1+1:end), Ncarriers, []));
    b2=transpose(reshape(simbol2(end-Num1+1:end), Ncarriers, []));
    cdatar1=[cdatar1;b1];
    cdatar2=[cdatar2;b2];
end;

% 6) IFFT

fftdata1=ifft(cdatar1,Ncarriers, 2);
fftdata2=ifft(cdatar2,Ncarriers, 2);

% 7) Afegir CP

fftdata_cp1=[fftdata1 fftdata1(:,1:round(CP*Ncarriers))];
fftdata_cp2=[fftdata2 fftdata2(:,1:round(CP*Ncarriers))];

% 8) Serialitzar

Cx1=reshape(fftdata_cp1.', [], 1);
```

```
Cx2=reshape(fftdata_cp2.', [], 1);
```

```
% 9) Clipping
```

```
% Per Tx1
deviation1=std(Cx1);
%Per dalt
index1= Cx1>=k*deviation1;
Cx_c1=Cx1;
Cx_c1(index1)=k*deviation1;
clear index1;
%Per baix
index1=find(Cx1<=-k*deviation1);
Cx_c1(index1)=-k*deviation1;

% Per Tx2
deviation2=std(Cx2);
%Per dalt
index2= Cx2>=k*deviation2;
Cx_c2=Cx2;
Cx_c2(index2)=k*deviation2;
clear index2;
%Per baix
index2=find(Cx2<=-k*deviation2);
Cx_c2(index2)=-k*deviation2;
```

- **Multiplexació TS's:**

En aquest punt, tenim la informació de cada senyal en dos arrays diferents (Cx_c1 i Cx_c2) que corresponen als component X i Y respectivament. Com es mostra a la figura 22 és necessari introduir zeros al senyal del component Y quan es transmeten els TS del senyal X i, al revés, quan s'envien els TS del senyal Y posar zeros al senyal X. Això és necessari per, més tard, poder recuperar i fer una bona estimació del canal en el receptor.

Com sabem quan s'envien els TS del senyal Y, només hem d'afegir zeros a la matriu del senyal X en la posició adient i fem el mateix pel cas invers. Els dos senyals es guarden a la matriu "j2".

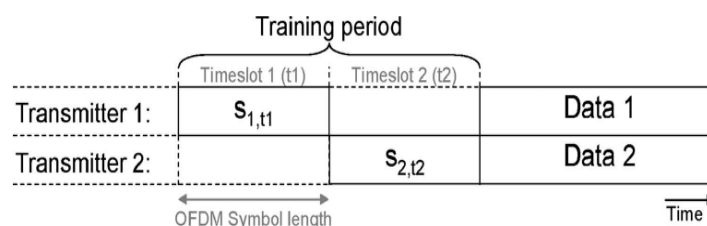


Figura 22. Multiplexació en temps dels TS per a PDM.

Codi:

```
% 10) Intercalar
```

```
% [TS1][ 0 ][Payload1]
% -----
```

```
% [ 0 ][TS2][Payload2]

Num2=Ncarriers+round(CP*Ncarriers);

%Tx1
a1=Cx_c1(1:end-(Num2*Num1/Ncarriers));
a2=reshape(a1,[],NTS);
a3=[a2(1:Num2,:);zeros(Num2,NTS);a2(Num2+1:end,:)];
a4=reshape(a3,[],1);
j2(:,1)=[a4;Cx_c1(end-(Num2*Num1/Ncarriers)+1:end)];

%Tx2
a12=Cx_c2(1:end-(Num2*Num1/Ncarriers));
a22=reshape(a12,[],NTS);
a32=[zeros(Num2,NTS);a22];
a42=reshape(a32,[],1);
j2(:,2)=[a42;Cx_c2(end-(Num2*Num1/Ncarriers)+1:end)];
```

- **Oversampling i filtratge**

Com en el cas anterior, tornem a sobremostrejar i filtrar amb els mateixos paràmetres. Les dades dels dos senyals es guarden a "j2_cmf".

Codi:

```
% 11)Upsampling i filtratge
j2_cm=upsample(j2,M); % Augmentem a 8 mostres/simbol
j2_cm_aux=[j2_cm; zeros(N/2,2)]; % afegir zeros al final
j2_cm_auxf=filter(h1,1,j2_cm_aux); % filtrar
j2_cmf=j2_cm_auxf(N/2+1:end,:);
```

Abans, en el SITO, teníem el mateix senyal que viatjava tant per l'eix X com l'eix Y, ara, al TITO, tenim dos senyals diferents.

3.2.2 Fibra òptica:

És idèntic al [codi introduït](#) pel SITO 1x2 MIMO OFDM. El senyal òptic abans d'entrar al receptor està a la matriu "j2ss".

3.2.3 Mòdul potència:

Com comentàvem la principal diferència amb el codi del sistema SITO és que ara la potència no es reparteix entre els dos components, per tant, no hem de restar cap valor com fèiem abans. També afegim dos "biterr" per calcular les BER dels dos senyals.

Codi:

```
for ksnr = 1:1:length(SNR),
    j4=awgn(j2ss, SNR(ksnr), 'measured');
    ...
    ...

[Nerr, BER1(ksnr)]=biterr(data(:,1), rec(:,1));
[Nerr, BER2(ksnr)]=biterr(data(:,2), rec(:,2));
```

end

Recordar que és millor treballar amb "awgn" matricialment i no per files o columnes.

3.2.4 Receptor:

- **Filtratge, delmació, paral·lelitzar, treure CP, FFT.**

Aquests passos són idèntics als realitzats amb el cas del pel SITO 1x2 MIMO OFDM ja que ara sí coincideix el nombre de receptors en tots dos sistemes.

Codi:

```
% Filtratge i downsampling
j4_mf_aux=[j4 zeros(2, N/2)]; % afegir zeros al final
j4_mf_auxf=filter(h1,1,j4_mf_aux); % filtrar
j4_m=j4_mf_auxf(:,N/2+1:end);

j5=downsample(j4_m.',M); %Delmació 1/8

% 1) Paral·lelitzar
%Dividim el senyal rebut en dos components Ex i Ey

rxdata_cp1=transpose(reshape(j5(:,1), round((1+CP)*Ncarriers),
[]));%dades principals S1
rxdata_cp2=transpose(reshape(j5(:,2), round((1+CP)*Ncarriers),
[]));%dades principals S2

% 2) Treure el cyclic prefix

rxdata1=rxdata_cp1(:, 1:Ncarriers);
rxdata2=rxdata_cp2(:, 1:Ncarriers);

% 3) calcular la FFT

fftdatarx1=fft(rxdata1,Ncarriers, 2);
fftdatarx2=fft(rxdata2,Ncarriers, 2);
```

- **Equalitzar**

Ara com tenim dos transmissors i dos receptors, la matriu z tindrà una dimensió de 2x2.

Per calcular els coeficients de la matriu, recuperarem els valors originals dels TS de totes dues senyals (TSr1 i TSr2) i, després, seleccionarem de les dades rebudes tant els TS enviats a través de la fibra com els efectes que han tingut en el senyal contrari (TSdata1 i TSdata2). Veiem que on teníem zeros abans, a l'enviar el TS de l'altre senyal, tenim dades complexes.

A continuació calculem les mitjanes dels valors anteriors per calcular els coeficients de la matriu z, on:

- Hs11, és la mitjana dels TS del senyal 1 per subportadora.
- Hs21, és la mitjana dels efectes dels TS del senyal 2 en el senyal 1 per subportadora.

- Hs12, és la mitjana dels efectes dels TS del senyal 1 en el senyal 2 per subportadora.
- Hs22, és la mitjana dels TS del senyal 2 per subportadora.

Ara z és:

$$z = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix}$$

on

$$h_{11} = \frac{Hs11(k)}{TSr(k)}$$

$$h_{21} = \frac{Hs21(k)}{TSr(k)}$$

$$h_{12} = \frac{Hs12(k)}{TSr(k)}$$

$$h_{22} = \frac{Hs22(k)}{TSr(k)}$$

Per aplicar l'algoritme ZF, fem la pseudoinversa de la matriu z amb l'ordre "pinv" i guardem el seu resultat a la matriu b com vam fer abans

Per últim, obtindrem els senyal estimats al multiplicar aquesta matriu "b" (2x2) per una altra formada per dues columnes on cadascuna representa cada component del senyal rebut i el serialitzem.

Hem de recordar que tot aquests càlculs es fan per cada subportadora, és a dir, per les 64.

Codi:

% 4) Equalitzar

```
TSr1=cdata1(1,:);
TSr2=cdata2(1,:);
r1=fftdatarx1(1:end-Num1/Ncarriers,:);
r2=fftdatarx2(1:end-Num1/Ncarriers,:);
TSdata1=reshape(r1.',[],NTS);
TSdata2=reshape(r2.',[],NTS);

Hs11=mean(TSdata1(1:Ncarriers,:),2); % Mitja TS1
Hs21=mean(TSdata1(Ncarriers+1:2*Ncarriers,:),2); % Mitja efectes
                                                TS2 en senyal X
Hs12=mean(TSdata2(1:Ncarriers,:),2); % Mitja efectes TS1 en senyal Y
Hs22=mean(TSdata2(Ncarriers+1:2*Ncarriers,:),2); % Mitja TS2

for k=1:Ncarriers %Calculem la pseudomatriu per subportadora
    h11=Hs11(k)/TSr1(k);h12=Hs21(k)/TSr2(k);
    h21=Hs12(k)/TSr1(k);h22=Hs22(k)/TSr2(k);
    z=[h11 h12; h21 h22];
    b=pinv(z);
```

```
%Estimació de coeficients
t(:,:,k)=b*[fftdatarx1(:,k) fftdatarx2(:,k)].';
%treiem símbols "sobrants"
t1(:,:,k)=t(:,1:end-Num1/Ncarriers,k);
%posem els TS a les 4 primeres files(2Tx*2bit/1TS)
t11(:,:,k)=reshape(t1(:,:,k),[],NTS);
t12(:,:,k)=t11(Txs*bps+1:end,:k); %eliminem els TS
t13(:,k)=reshape(t12(:,:,k),[],1);
%en serie els símbol sobrants
t15(:,k)=reshape(t(:,end-Num1/Ncarriers+1:end,k),[],1);
t16(:,k)=[t13(:,k);t15(:,k)];%els afegim
end;

t18=reshape(transpose(t16),Txs*Ncarriers,[]);

rdata(:,1)=reshape(t18(1:Ncarriers,:),[],1);
rdata(:,2)=reshape(t18(Ncarriers+1:end,:),[],1);
```

- **Demapejar QPSK**

Una vegada tenim els dos senyals estimat a la matriu "rdata", només ens queda transformar els símbols QPSK en bits i guardar-los en la matriu "rec".

Codi:

```
% 11) demapejar QPSK
hDemod = comm.QPSKDemodulator('BitOutput',true);
rec(:,1) = step(hDemod, rdata(:,1));
rec(:,2) = step(hDemod, rdata(:,2));
```

- **Gràfics:**

Per últim, com van fer també abans, comparem els valors obtinguts de la BER per a cada SNR amb els valors ideals de la BER per a QPSK dels dos senyals. La BER teòrica de QPSK és:

$$BER = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \left[2 - Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\right]$$

Al gràfic imposem que l'eix Y sigui logarítmic i posem les dues corbes dels dos senyals a la mateixa gràfica que la corba ideal per a comparar-les.

Codi:

```
% 13) Fer la gràfica de BER vs SNR
ebno=SNR-10*log10(bps); % Calcul de la Eb/No d'acord amb doc. Matlab
AWGN
ebnolin=10.^(ebno/10); % Passa Eb/No a linial
BERteorica=(1/bps)*erfc(sqrt(ebnolin)).*(1-.25*erfc(sqrt(ebnolin))); %
Formula teorica (Proakis/Salehi 5a ed; cap. 4, pags. 193/195)

% 13) Fer la gràfica de BER vs SNR
semilogy(SNR, BER1,SNR,BER2,SNR,BERteorica);
grid on;
```

```
xlabel('SNR (dB)');  
ylabel('BER');  
legend('Simulacions MIMO TX1','Simulacions MIMO TX2','Teoria');
```

4. Simulació i resultats

4.1. Sense fibra

Amb aquesta configuració podem analitzar com es comporta el sistema OFDM per si sol i observem que té un comportament molt similar a la corba de la BER teòrica, encara que si modifiquem el nombre de TS per subportadora millorarem la resposta fins a arribar a superposar quasi del tot la corba teòrica amb la de simulació.

Per a NTS=5, obtenim una corba per sobre de la corba teòrica amb una SNR=10,3db per una BER de 10^{-3} . En canvi per a la corba teòrica, tenim una SNR de 9.8db.

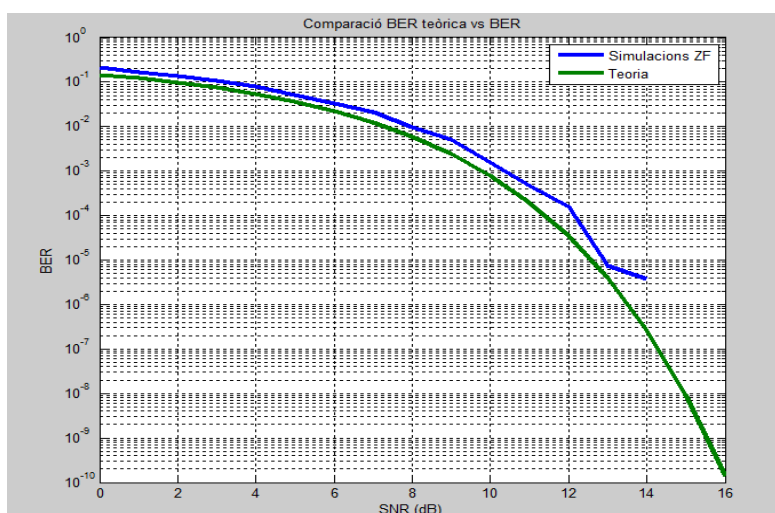


Figura 23. BER sense fibra amb NTS=5.

Si augmentem la quantitat de TS per subportadora (NTS), per exemple NTS= 50, obtenim una resposta quasi idèntica a la corba teòrica.

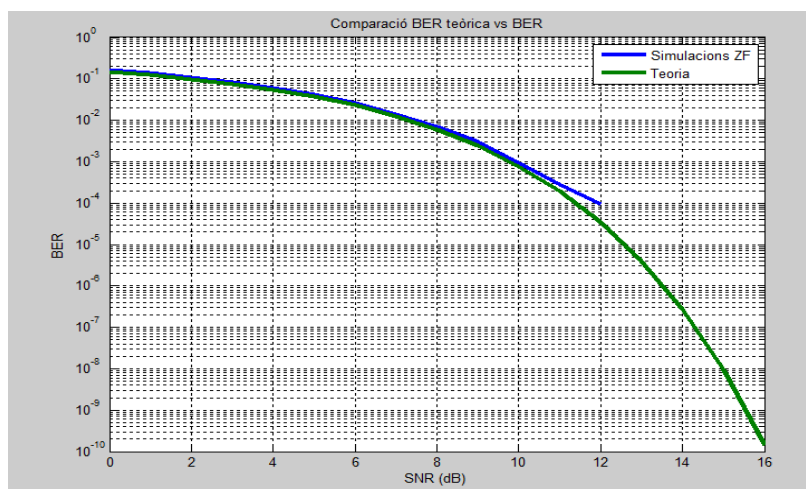


Figura 24. BER sense fibra amb NTS=50

4.2. CO-OFDM amb SITO (1x2 MIMO OFDM).

Treballarem amb valors de NTS=5, ja que donen valors acceptables i així reduïrem l'overhead necessari per a transmetre la informació. El bit rate del sistema serà 10Gbps i la quantitat d'informació transmesa seran 128k símbols. A més, considerarem que els eixos de la birefringència de la fibra coincideixen amb els dels eixos X i Y, és a dir, PSP=[0,0].

- **Efecte en el símbols QPSK**

En primer lloc, volíem comprovar els efectes que tindria en les constel·lacions QPSK les diferents distorsions que presenta la fibra i com la diversitat del receptor permet estimar el senyal rebut. Per mostrar-ho agafarem un exemple senzill amb una fibra de L=20km.

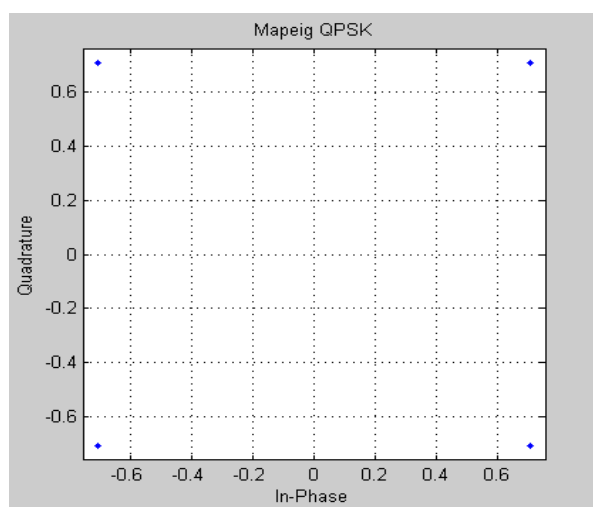


Figura 25. Constel·lació QPSK a transmetre.

A la figura 25 veiem representats els quatre símbols QPSK de les dades, mentre que a la figura 26 es representen com arriben els símbols al receptor, es pot observar com sofreixen distorsions tant en amplitud com en fase.

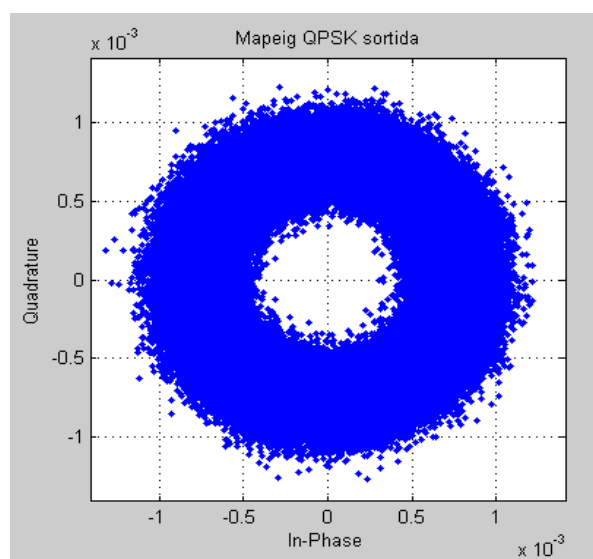


Figura 26. Constel·lació QPSK rebuda

En canvi, a la figura 27 es veu com l'etapa equalitzadora del receptor ha pogut compensar les distorsions que havia sofert el senyal.

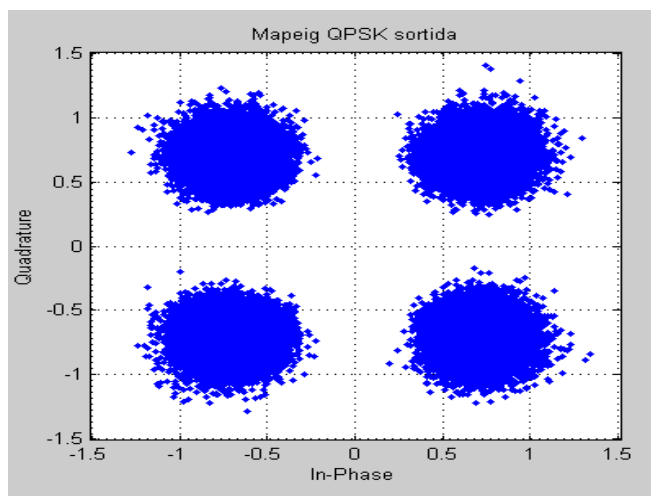


Figura 27. Constel·lació QPSK compensada

- **Efectes amb l'augment de la distància.**

Per una longitud inicial de $L=20\text{km}$ la corba de la BER està molt a prop dels valors ideals. Tenim una SNR de 10,15db per la BER típica, a continuació presentem la gràfica:

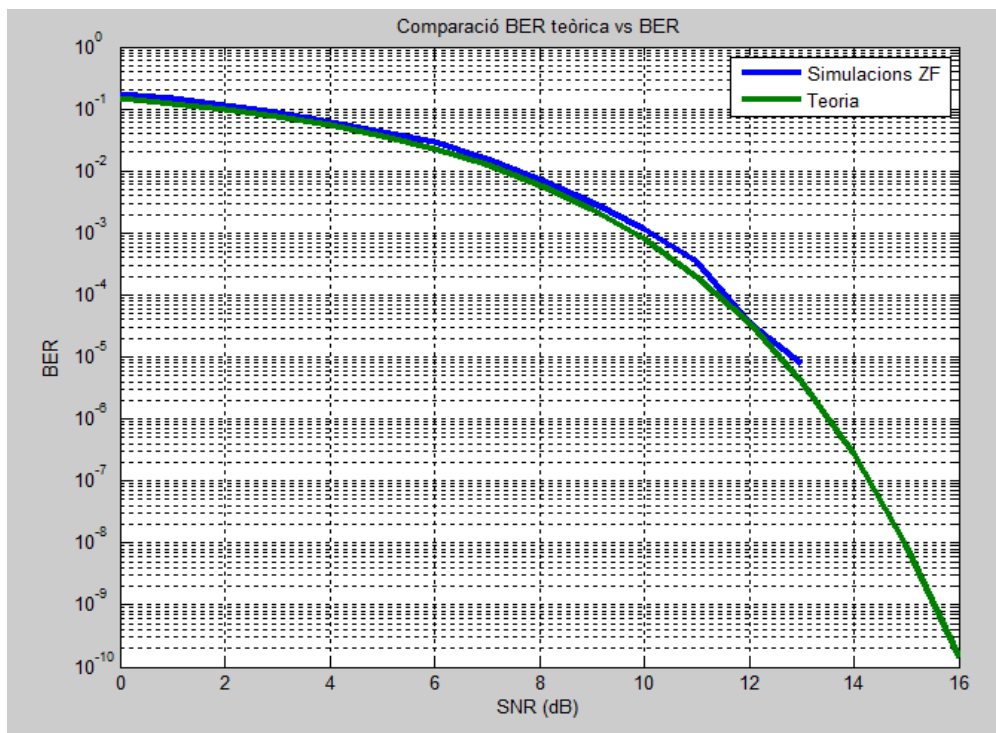


Figura 28. BER SITO per a $L=20\text{km}$

Si augmentem la distància veiem que la BER es comporta cada vegada una mica pitjor, per a $L=100\text{km}$ tenim una SNR=10,45db per a la BER típica.

El comportament del sistema per aquesta distància és:

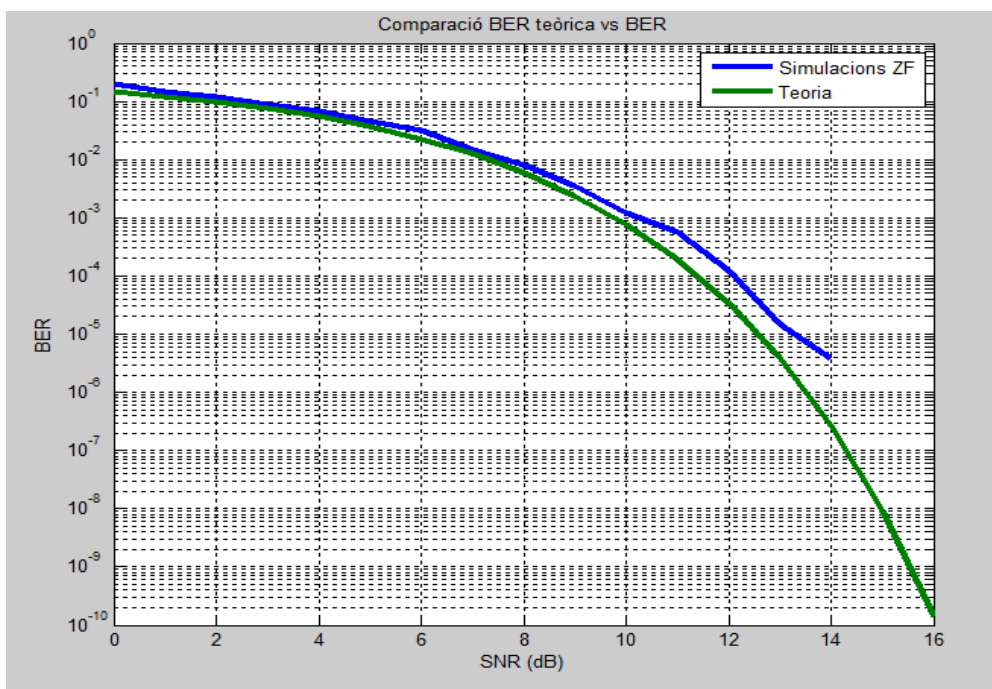


Figura 29. BER SITO per a L=100km

Si incrementem la longitud a L=150km obtenim un empitjorament superior, SNR=11,4db.

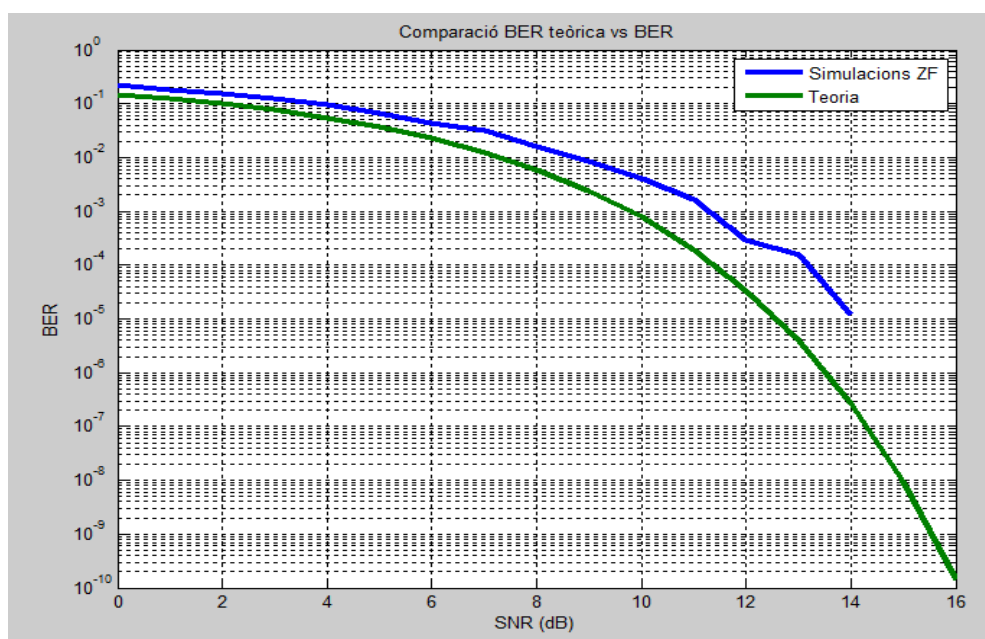


Figura 30. BER SITO per a L=150km

Per últim, mirarem quins valors obtenim per a una distància de 450Km fets amb 3 trams de fibra de 150km i un guany entre fibres de G=31,66 que equival a 15db.

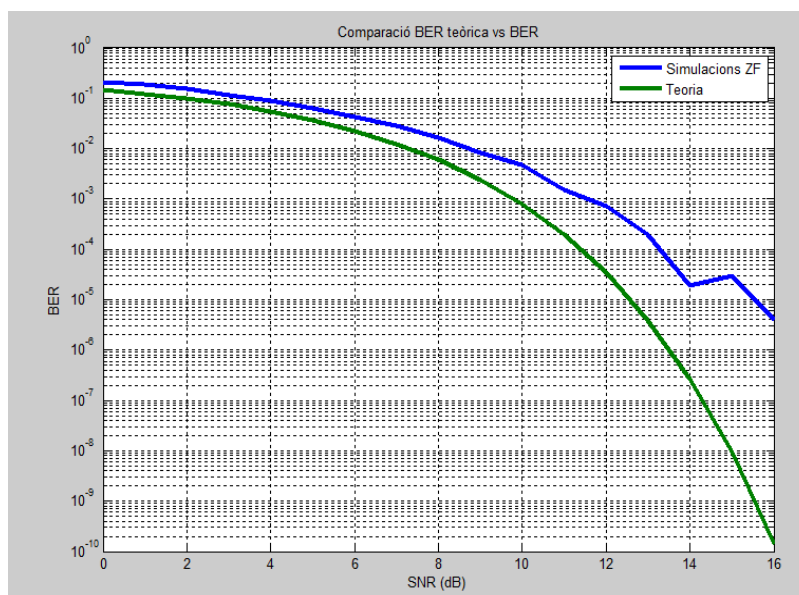


Figura 31. BER SITO per a L=450km

La SNR és de 11,5db per la BER típica. Com havíem suposat, l'augment de la distància ens incrementa els efectes de les distorsions a la fibra i haurem d'augmentar la potència del sistema per assolir uns comportaments adients.

4.3. CO-OFDM amb TITO (2x2 MIMO OFDM)

- Efectes amb l'augment de la distància.

Ara, farem servir aquesta configuració per aplicar PDM i enviar dos senyals diferents, és a dir, tenim l'avantatge de poder enviar la mateixa informació que abans però amb la meitat de velocitat, el que farà que la distorsió provocada per la PMD de primer ordre sigui menor i obtindrem una millor resposta que en el cas anterior. El bit rate continua sent de 10Gbps però els símbols transmesos seran 256K, el doble que en els casos anteriors. Observem que ara tenim dues corbes amb un comportament quasi idèntic i, com abans, cada vegada que augmentem la distància el sistema tindrà un comportament pitjor. Per una distància de L=20km, els valors obtinguts pels dos senyals són molt semblants, la SNR és de 10,6db.

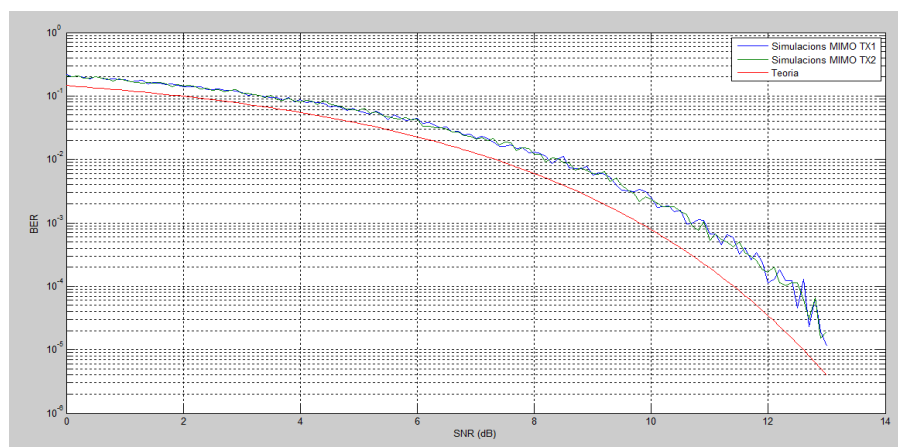


Figura 32. BER TITO per a L=20km

Per a $L=50\text{km}$ obtenim una $\text{SNR}=11,1\text{db}$ per la BER típica.

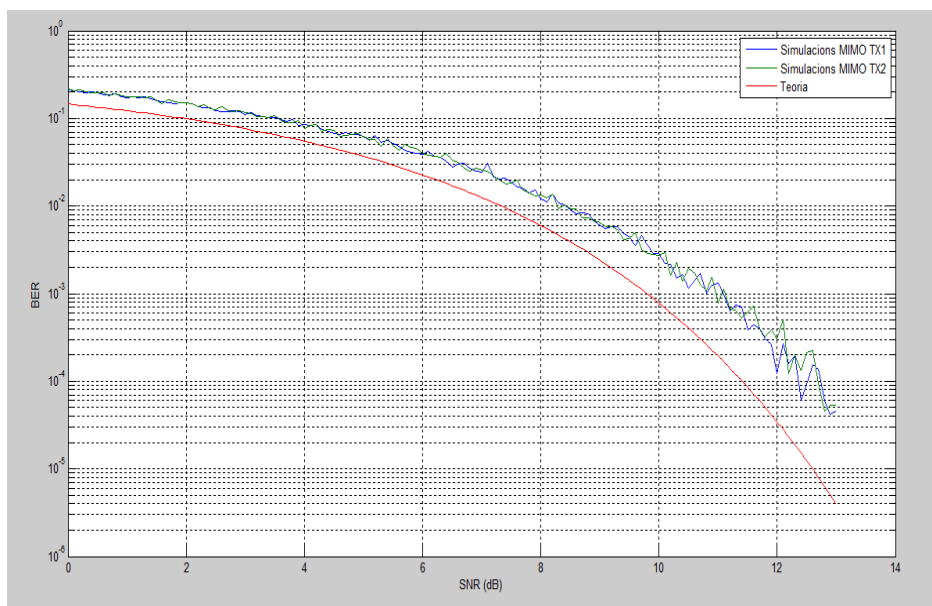


Figura 33. BER TITO per a $L=50\text{km}$

Per una $L=150\text{km}$ obtenim una $\text{SNR}=12.2\text{db}$.

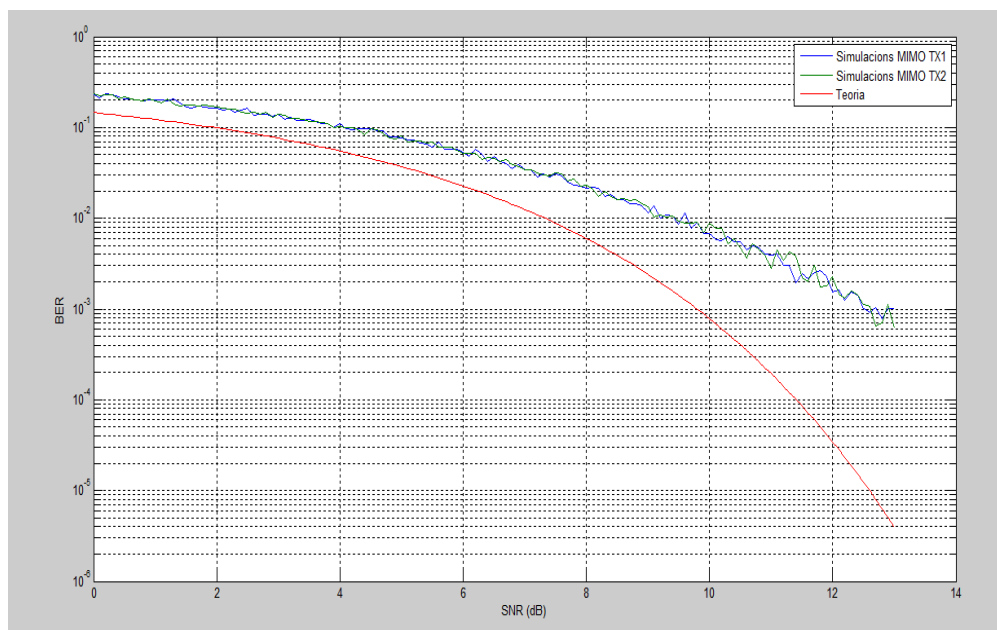


Figura 34. BER TITO per a $L=150\text{km}$

Per últim, estudiem la corba per una distància de 450km fets amb 3 trams de fibra de 150km com en el cas anterior:

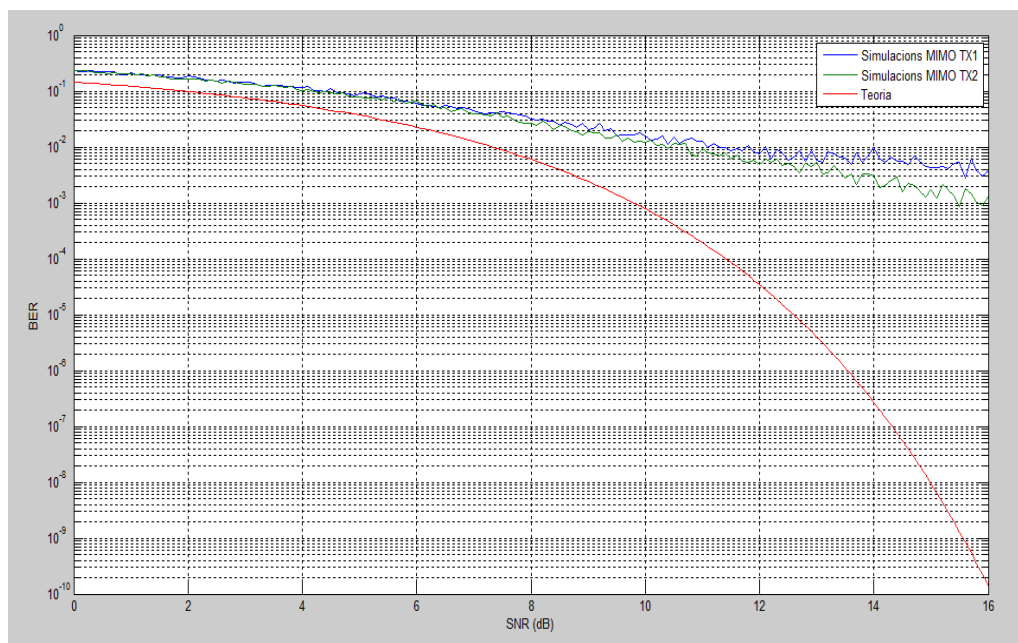


Figura 35. BER TITO per a $L=450\text{km}$

Obtenim una SNR aproximadament de 16db per la BER típica.

4.4. SITO vs TITO.

En aquest apartat volem fer una comparació entre el comportament que ofereix un sistema SITO i un altre TITO amb les mateixes característiques. El TITO serà el model que hem estudiat abans: bit rate de 10Gbps, 256K símbols d'informació, 5 TS per subportadora i les diferents longituds observades.

L'arquitectura SITO equivalent per enviar la mateixa informació en el mateix temps ha de doblar la seva velocitat de transmissió, per tant, tindrem un bit rate de 20Gbps, 256K símbols d'informació i 5 TS per subportadora.

Primer, observem la corba resultant per a una $L=20\text{km}$:

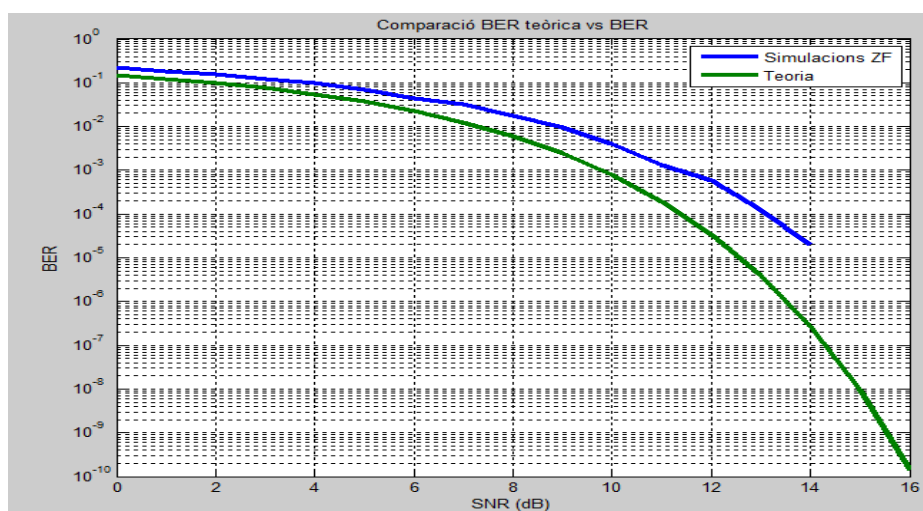


Figura 36. BER SITO per a $L=20\text{km}$

Obtenim una SNR d'11,35db per la BER típica, un valor molt més elevat que per al TITO equivalent que tenia una SNR de 10,60db.

Si augmentem la distància fins a L=50Km:

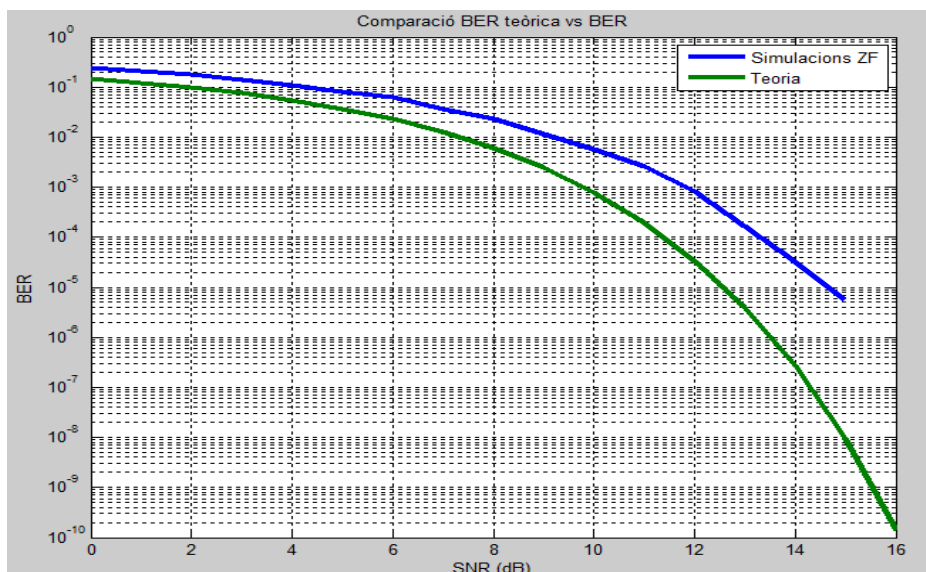


Figura 37. BER SITO per a L=50km

El valor de SNR augmenta fins a 12db, continua sent superior que pel cas del TITO que només tenia un SNR d'11,1db. Si provem per a una distància de la fibra de L=150km tenim la següent gràfica:

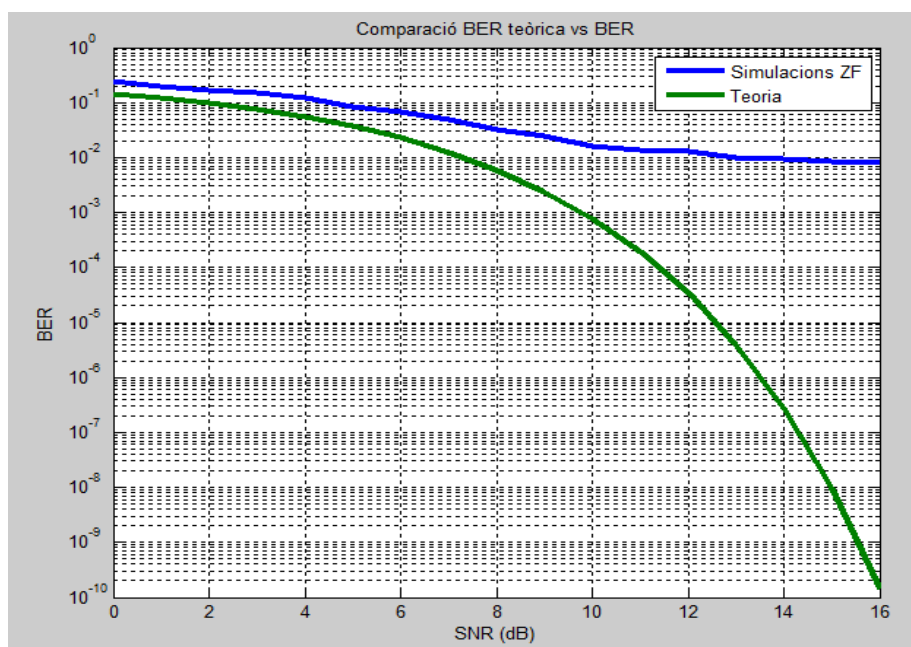


Figura 38. BER SITO per a L=150km

En aquest cas, els valors de SNR són superiors als 16db mentre que per la versió que utilitza PDM tenim una SNR de 12,20db i, fins i tot, podem obtenir bons resultats fins a longituds de 450km.

4.5. Taules comparatives:

A continuació presentem una taula amb tots els valors de la SNR per la BER típica dels casos que hem estudiat anteriorment:

SITO 10Gbps 128k 5 TS		SITO 20Gbps 256k 5 TS		TITO 10Gbps 256k 5 TS	
Distància	SNR	Distància	SNR	Distància	SNR
20km	10.15db	20km	11.35db	20km	10.60db
100km	10.45db	50km	12db	50km	11.10db
150km	11.40db	150km	>16db	150km	12.20db
450km	11.50db	450km	>16db	450km	16db

Figura 39. Taula comparativa SNR

En una primera observació, com ja havíem comentat abans, en tots tres casos a l'augmentar els kilòmetres necessitem una SNR major per mantenir la BER de 10^{-3} , per tant, tenim una major distorsió a l'augmentar la distància

Si comparem els dos sistemes SITO, veiem que tenim un pitjor comportament on tenim el bit rate de 20Gbps, per tant, un augment de la velocitat de transmissió comporta una pitjor estimació del canal i un augment dels errors en la transmissió.

Per últim, si comparem el SITO de 20Gbps i el TITO de 10Gbps veiem que utilitzar PDM és beneficiós per la transmissió ja que tenim uns valors de SNR més petits i es mantenen durant més temps amb la distància, pel SITO a 150km ja tenim valors de SNR superiors a 16db i en canvi al TITO no arribem a aquest valor fins als 450km. Aquest avantatge dels sistemes TITO enfront al d'un sol transmissor es basa en el fet que la PMD no afecta tant al sistema que treballa amb una velocitat de transmissió inferior, en aquest cas 10Gbps.

5. Conclusions i possibles millores

Una vegada finalitzat el treball i analitzant les tasques i els estudis fets podem destacar entre les principals conclusions:

- S'ha analitzat la sinergia entre la detecció coherent i la modulació OFDM comprovant la seva capacitat per compensar la PMD de primer ordre.
- S'ha comprovat com la PMD és una barrera per a aplicacions "long-haul" i com pot afectar al sistema de comunicació a l'augmentar tant el "bit rate" com la distància. Fins i tot, fent inviable una bona transmissió de les dades per a certes configuracions.
- La necessitat d'investigar noves tècniques per a mitigar les penalitzacions que produeixen les PMD d'ordres superiors. La forta demanda de majors capacitats i velocitats dels sistemes de comunicació per part de la societat fan que prenguin cada vegada més protagonisme.
- Hem estudiat quines estratègies es realitzen per a construir tant sistemes amb SITO (1x2 MIMO OFDM) com TITO (2x2 MIMO OFDM). En aquest aspecte, el treball ha estat molt heterogeni, ja que s'han tractat una bona varietat de temes diferents:

funcionament modulació OFDM, detecció coherent, paràmetres i funcionament de la fibra òptica, coneixement de MATLAB i SSPROP, polarització de la llum, etc.

- És un treball que permet una bona aproximació inicial a aquests sistemes de comunicació òptica i que pot ser la base per a futures ampliacions o estudis més exhaustius.

Durant el desenvolupament del treball s'anaven obrint noves vies d'estudi i possibilitats per a millorar alguns aspectes del projecte que poden ser interessants en futurs estudis o treballs:

- Analitzar altres estimadors diferents a ZF com poden ser LSE o MMSE per comprovar quina resposta ofereixen. Per exemple, MMSE té un millor comportament que ZF ja que per fer l'estimació del canal sí té en compte el soroll que s'introdueix en ell, en canvi ZF no ho fa. MMSE millora a ZF en 1db per a QPSK.
- Canviar el mapeig de QPSK per un m-QAM, que permetrà augmentar la capacitat i velocitat del sistema amb un petit canvi en el codi.
- Fer un estudi més ampli de la resposta dels sistemes a longituds superiors a 500km.
- Implementar el bloc de conversió optoelectrònica.
- Afegir condicions que ens apropin més a un sistema real: no linealitats, PDL, soroll de fase, etc.

6. Codi MATLAB

CO-OFDM amb SITO (1x2 MIMO OFDM)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% CO-OFDM amb SITO (1x2 MIMO OFDM)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Informació:256kbps
%   QPSK 2bits/simbol
%   64 subportadores
%   5 TS per subportadora repartits
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; close all; % Borra variables i tanca figures
Ncarriers=64;
BitRate=10e9;          % Velocitat de transmissió
bps=2;
NTS=5;                 % TS per subportadora
CP=0.25;k=3;
Nsymbol=128*1024;
SNR=0:1:16;           % Canviat de pas de 0.1 a 1, triga menys

M=8;                   %Factor oversampling

%Valors filtre
N=34;                  %Ordre del filtre
fc=BitRate/2;         %frecuència de tall
r=0.1;                %roll-off
fm=M*2*fc;            %frecuència de mostratge
```

```
h1 = firrcos(N,fc,r,fm,'rolloff','sqrt');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Transmissor
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s = RandStream('mt19937ar','Seed',1); % Fem que "s" sigui un generador
de numeros aleatoris amb llavor 1
RandStream.setGlobalStream(s); % Fixem el generador intern de numeros
aleatoris de Matlab a "s"
% 1) Generar les dades de "payload"
data=randi([0 1],1,Nsymbol*bps);

% 2) Generar les dades de TS
TS=randi([0 1],1,Ncarriers*bps);
TS1=(repmat(TS,1,NTS));

% 3) Mapejar les dades resultants amb QPSK
hModulator = comm.QPSKModulator('BitInput',true);
simbol = step(hModulator, data);
simbolTS = step(hModulator, TS1);

% 4) Paral·lelitzar i introduir TS de forma periòdica
Num=fix(Nsymbol/(Ncarriers*NTS)); % Quocient divisió
Num1=mod(Nsymbol,(Ncarriers*NTS)); % Mòdul divisió
c=reshape(simbol(1:end-Num1),Ncarriers*Num,[]);
d=reshape(simbolTS,Ncarriers,[]);
c1=[d;c];
cdatar=transpose(reshape(c1,Ncarriers,[]));
if Num1~=0 % Afegim símbols "lliures"
    b1=transpose(reshape(simbol(end-Num1+1:end),Ncarriers,[]));
    cdatar=[cdatar;b1];
end;

% 6) Fer la transformada inversa
fftdata=ifft(cdatar,Ncarriers,2);

% 7) Afegir el "cyclic prefix"
fftdata_cp=[fftdata fftdata(:,1:round(CP*Ncarriers))];

% 8) Serialitzar
Cx=reshape(transpose(fftdata_cp),[],1);

% 9) Clipping
deviation=std(Cx);

%Per dalt
index= Cx>=k*deviation;
Cx_c=Cx;
Cx_c(index)=k*deviation;
clear index;

%Per baix
index=find(Cx<=-k*deviation);
Cx_c(index)=-k*deviation;

% 10) Upsampling i filtratge
Cx_cm=upsample(Cx_c,M); % Augmentem a 8 mostres/símbol
```

```
Cx_cm_aux=[Cx_cm; zeros(N/2,1)]; % Afegir zeros al final
Cx_cm_auxf=filter(h1,1,Cx_cm_aux); % Filtrar
Cx_cmf=Cx_cm_auxf(N/2+1:end);

%) 11)Divisor de polarització:
p=[1 1];
j2=Cx_cmf*p;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Efecte canal fibra òptica
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Valors típics fibra òptica monomode
c = 3e8; %[m/sec] velocitat de la llum
lambda0 = 1550e-9; %[m] longitud d'ona central
alphadB = 0.2; %[dB/m] atenuació de la fibra
alpha = (alphadB*1e-3)/(10*log10(exp(1))); % pas a linial
Dp = 16.5e-6; %[s/m^2] dispersió
S = 0.057e3; %[s/m^3] dispersion slope
Dpmd=0.04e-12; %[s/m^0.5] dispersió PMD
birref=1e-7; %Birrefringència

% Càlcul dels coeficients de dispersió
beta2=-((Dp*(lambda0^2))/(2*pi*c));
beta3=((S*lambda0^4)/(4*(pi^2)*(c^2)))-beta2*lambda0/(pi*c);

%Vector de Jones fibra
psp=[0 0]; % Alineació amb el vector de Jones del senyal

n2=2.4e-20; %[m^2/W] index de no-linialitat
Aeff=80e-12; %[m^2] area efectiva
gamma= 2*pi*n2/(Aeff*lambda0); % Coeficient de no linialitat de la
fibra

fiber_length=100e3; % [m] Longitud de la fibra
max_deltaz=1e3;% [m] Màxima longitud de pas per al Split-Step
max_phase_change=.05; % [degrees] Màxim canvi de fase

beta0=pi*birref/lambda0;
beta1=Dpmd/(2*c);

betap1=[beta0, beta1, beta2, beta3]; % array de dispersió
betap2=[-beta0, -beta1,beta2, beta3]; % array de dispersió

% Definim l'increment i el número de passos

deltaz=max_phase_change*(pi/180)/(gamma*max(abs(j2(:,1)).^2+abs(j2(:,2)
)).^2));
%deltaz=500;
if deltaz>max_deltaz
    deltaz=max_deltaz;
end

step_num=fiber_length/deltaz;

% Entrem a la fibra
```

```
j1ss=j2;

[j21a, j22b]= sspropv(j1ss(:,1), j1ss(:,2), 1/fm,deltaz,step_num,
alpha, alpha, betap1, betap2, gamma, psp, 'elliptical', 100, 1e-5);

j2ss=G*[j21a j22b].';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Carrega soroll
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for ksnr = 1:1:length(SNR),

    j3_mf=awgn(j2ss, SNR(ksnr)-3, 'measured'); % Els 3 dB son pel fet
de transmetre en nomes 1 comp. de la pol;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Receptor
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Filtratge i downsampling
j3_mf_aux=[j3_mf zeros(2, N/2)]; % afegir zeros al final
j3_mf_auxf=filter(h1,1,j3_mf_aux); % filtrar
j3_m=j3_mf_auxf(:,N/2+1:end);

j3=downsample(j3_m.',M); %Delmació 1/8

% 1) paral·lelitzar

rxdata_cp1=transpose(reshape(j3(:,1), round((1+CP)*Ncarriers), []));
rxdata_cp2=transpose(reshape(j3(:,2), round((1+CP)*Ncarriers), []));

% 2) Treure el cyclic prefix

rxdata1=rxdata_cp1(:, 1:Ncarriers);
rxdata2=rxdata_cp2(:, 1:Ncarriers);

% 3) calcular la FFT
fftdatarx1=fft(rxdata1,Ncarriers, 2);
fftdatarx2=fft(rxdata2,Ncarriers, 2);

% 4) Equalitzar
TSr=cdatar(1,:);
r1=fftdatarx1(1:end-Num1/Ncarriers,:);
r2=fftdatarx2(1:end-Num1/Ncarriers,:);
TSdata1=reshape(r1.', [],NTS);
TSdata2=reshape(r2.', [],NTS);
Hs1=mean(TSdata1(1:Ncarriers,:),2); % Fem la mitja dels símbols
d'entrenament rebuts
Hs2=mean(TSdata2(1:Ncarriers,:),2); % Fem la mitja dels símbols
d'entrenament rebuts

for k=1:Ncarriers %Calculem la pseudomatriu per a cada subportadora
    h11=Hs1(k)/TSr(k);
    h21=Hs2(k)/TSr(k);
    z=[h11; h21];
```

```
b=pinv(z);
t(:,k)=b*[fftdatarx1(:,k) fftdatarx2(:,k)].';%Estimació de
                                                coeficients

end

% 11) demapejar QPSK

r2=reshape(transpose(t(1:end-Num1/Ncarriers,:)), [],NTS);
rdata=reshape(r2(Ncarriers+1:end,:), [],1); %Treiem TS
if Num1~=0
    r4=t(end-(Num1/Ncarriers)+1:end,:);
    r5=reshape(r4.', [],1);
    rdata=[rdata;r5];
end;

hDemod = comm.QPSKDemodulator('BitOutput',true);
rec = step(hDemod, rdata);

[Nerr, BER(ksnr)]=biterr(data, rec');

recmmse = step(hDemod, rdatammse);
[Nerrmmse, BERmmse(ksnr)]=biterr(data, recmmse');

end

% 13) Fer la gràfica de BER vs SNR
ebno=SNR-10*log10(bps); % Calcul de la Eb/No d'acord amb doc. Matlab
AWGN
ebnolin=10.^(ebno/10); % Passa Eb/No a linial
BERteorica=(1/bps)*erfc(sqrt(ebnolin)).*(1-.25*erfc(sqrt(ebnolin))); %
Formula teorica (Proakis/Salehi 5a ed; cap. 4, pags. 193/195)

semilogy(SNR,BER,SNR,BERteorica,SNR,BERmmse);

xlabel('SNR (dB)');
ylabel('BER');
grid on;
legend('Simulacions ZF','Teoria','Simulacions MMSE');
title('Comparació BER teòrica vs BER');
```

CO-OFDM amb TITO (2x2 MIMO OFDM)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CO-OFDM amb TITO (2x2 MIMO OFDM)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Informació:256kbps
%   QPSK 2bits/simbol
%   64 subportadores
%   5 TS per subportadora repartits
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; close all; % Borra variables i tanca figures
Ncarriers=64;
bps=2; % Bits per simbol QPSK
BitRate=10e9; % Velocitat de transmissió

Txs=2; % Nombre de transmissors
NTS=5; % TS per subportadora
```



```
CP=0.25;k=3;
Nsymbol=128*1024;
SNR=0:.1:13;

M=8; % Factor oversampling
%Valors filtre
N=32; % Ordre del filtre
fc=BitRate/2; % freqüència de tall
r=0.15; % roll-off
fm=M*2*fc; % freqüència de mostratge

h1 = firrcos(N,fc,r,fm,'rolloff','sqrt');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Transmissor
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s = RandStream('mt19937ar','Seed',1); % Fem que "s" sigui un generador
de numeros aleatoris amb llavor 1
RandStream.setGlobalStream(s); % Fixem el generador intern de numeros
aleatoris de Matlab a "s

% 1) Generar les dades de "payload"
data(:,1)=randi([0 1],1,Nsymbol*bps);
data(:,2)=randi([0 1],1,Nsymbol*bps);

% 2) Generació de TS
TS=randi([0 1],1,Ncarriers*bps);
TS1= repmat(TS,1,NTS);

% 3) Mapejar les dades resultants amb la constel·lació que toca
hModulator = comm.QPSKModulator('BitInput',true);
simbol1 = step(hModulator, data(:,1));
simbol2 = step(hModulator, data(:,2));
simbolTS1 = step(hModulator, TS1);
simbolTS2=simbolTS1;

% 4) Paral·lelitzar i introduir TS de forma periòdica
Num=fix(Nsymbol/(Ncarriers*NTS)); % Quocient divisió
Num1=mod(Nsymbol,(Ncarriers*NTS)); % Mòdul divisió
c1=reshape(simbol1(1:end-Num1),Ncarriers*Num,[]);%Agafem els
simbols justos
c2=reshape(simbol2(1:end-Num1),Ncarriers*Num,[]);
d1=reshape(simbolTS1,Ncarriers,[]);
d2=reshape(simbolTS2,Ncarriers,[]);
c11=[d1;c1];
c22=[d2;c2];
cdatar1=transpose(reshape(c11,Ncarriers,[]));
cdatar2=transpose(reshape(c22,Ncarriers,[]));

if Num1~=0 %Afegim símbols "lliures"
b1=transpose(reshape(simbol1(end-Num1+1:end),Ncarriers,[]));
b2=transpose(reshape(simbol2(end-Num1+1:end),Ncarriers,[]));
cdatar1=[cdatar1;b1];
cdatar2=[cdatar2;b2];
end;

% 6) IFFT
```

```
fftdata1=ifft(cdatar1,Ncarriers, 2);
fftdata2=ifft(cdatar2,Ncarriers, 2);

% 7) Afegir CP
fftdata_cp1=[fftdata1 fftdata1(:,1:round(CP*Ncarriers))];
fftdata_cp2=[fftdata2 fftdata2(:,1:round(CP*Ncarriers))];

% 8) Serialitzar
Cx1=reshape(fftdata_cp1.', [], 1);
Cx2=reshape(fftdata_cp2.', [], 1);

% 9) Clipping

% Per Tx1
deviation1=std(Cx1);
%Per dalt
index1= Cx1>=k*deviation1;
Cx_c1=Cx1;
Cx_c1(index1)=k*deviation1;
clear index1;
%Per baix
index1=find(Cx1<=-k*deviation1);
Cx_c1(index1)=-k*deviation1;

% Per Tx2
deviation2=std(Cx2);
%Per dalt
index2= Cx2>=k*deviation2;
Cx_c2=Cx2;
Cx_c2(index2)=k*deviation2;
clear index2;
%Per baix
index2=find(Cx2<=-k*deviation2);
Cx_c2(index2)=-k*deviation2;

% 10) Intercalar

% [TS1] [ 0 ] [Payload1]
% -----
% [ 0 ] [TS2] [Payload2]

Num2=Ncarriers+round(CP*Ncarriers);

%Tx1
a1=Cx_c1(1:end-(Num2*Num1/Ncarriers));
a2=reshape(a1,[],NTS);
a3=[a2(1:Num2,:);zeros(Num2,NTS);a2(Num2+1:end,:)];
a4=reshape(a3,[],1);
j2(:,1)=[a4;Cx_c1(end-(Num2*Num1/Ncarriers)+1:end)];

%Tx2
a12=Cx_c2(1:end-(Num2*Num1/Ncarriers));
a22=reshape(a12,[],NTS);
a32=[zeros(Num2,NTS);a22];
a42=reshape(a32,[],1);
j2(:,2)=[a42;Cx_c2(end-(Num2*Num1/Ncarriers)+1:end)];
```

```
% 11)Upsampling i filtratge
j2_cm=upsample(j2,M); % Augmentem a 8 mostres/simbol
j2_cm_aux=[j2_cm; zeros(N/2,2)]; % afegir zeros al final
j2_cm_auxf=filter(h1,1,j2_cm_aux); % filtrar
j2_cmf=j2_cm_auxf(N/2+1:end,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Efecte canal fibra òptica
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Valors típics fibra òptica monomode

c = 3e8; % [m/sec] velocitat de la llum
lambda0 = 1550e-9; % [m] longitud d'ona central
alphadB = 0.2; % [dB/m] atenuació de la fibra
alpha = (alphadB*1e-3)/(10*log10(exp(1))); % pas a linial
Dp = 16.5e-6; % [s/m^2] dispersió
S = 0.057e3; % [s/m^3] dispersion slope
Dpmd=0.04e-12; % [s/m^0.5] dispersió PMD
birref=1e-7; % Birrefringència

% Càlcul dels coeficients de dispersió
beta0=birref/2;
beta1=birref/(2*c);
beta2=-((Dp*(lambda0^2))/(2*pi*c));
beta3=((S*lambda0^4)/(4*(pi^2)*(c^2)))-beta2*lambda0/(pi*c);

%Vector de Jones
psp=[0 0];

n2=2.4e-20; % [m^2/W] index de no-linialitat
Aeff=80e-12; % [m^2] area efectiva
gamma= 2*pi*n2/(Aeff*lambda0); % Coeficient de no linialitat de la
                                fibra

fiber_length=10e3; % [m] Longitud de la fibra
max_deltaz=1e3; % [m] Màxima longitud de pas per al Split-Step
max_phase_change=.05; % [degrees] Màxim canvi de fase

betap1=[beta0, beta1, beta2, beta3]; % array de dispersió
betap2=[-beta0, -beta1,beta2, beta3]; % array de dispersió

% Definim l'increment i el número de passos

deltaz=max_phase_change*(pi/180)/(gamma*max(abs(j2_cmf(:,1)).^2+abs(j2
_cmf(:,2)).^2));

if deltaz>max_deltaz
    deltaz=max_deltaz;
end

step_num=fiber_length/deltaz;

% Entrem a la fibra

[j21a, j22b]= sspropv(j2_cmf(:,1), j2_cmf(:,2), 1/fm,deltaz,step_num,
alpha, alpha, betap1, betap2, gamma, psp, 'elliptical', 100, 1e-5);
```

```
j2ss=[j21a j22b].';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Carrega soroll
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for ksnr = 1:1:length(SNR),
    j4=awgn(j2ss, SNR(ksnr), 'measured');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Receptor
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Filtratge i downsampling
j4_mf_aux=[j4 zeros(2, N/2)]; % afegir zeros al final
j4_mf_auxf=filter(h1,1,j4_mf_aux); % filtrar
j4_m=j4_mf_auxf(:,N/2+1:end);

j5=downsample(j4_m.',M); % Delmació 1/8

% 1) Paral·lelitzar
%Dividim el senyal rebut en dos components Ex i Ey

rxdata_cp1=transpose(reshape(j5(:,1), round((1+CP)*Ncarriers),
[]));%dades principals S1
rxdata_cp2=transpose(reshape(j5(:,2), round((1+CP)*Ncarriers),
[]));%dades principals S2

% 2) Treure el cyclic prefix

rxdata1=rxdata_cp1(:, 1:Ncarriers);
rxdata2=rxdata_cp2(:, 1:Ncarriers);

% 3) calcular la FFT

fftdatarx1=fft(rxdata1,Ncarriers, 2);
fftdatarx2=fft(rxdata2,Ncarriers, 2);

% 4) Equalitzar

TSr1=cdata1(1,:);
TSr2=cdata2(1,:);
r1=fftdatarx1(1:end-Num1/Ncarriers,:);
r2=fftdatarx2(1:end-Num1/Ncarriers,:);
TSdata1=reshape(r1.', [],NTS);
TSdata2=reshape(r2.', [],NTS);

Hs11=mean(TSdata1(1:Ncarriers,:),2); % Fem la mitja dels símbols
d'entrenament rebuts
Hs21=mean(TSdata1(Ncarriers+1:2*Ncarriers,:),2); % Fem la mitja
d'entrenament rebuts
Hs12=mean(TSdata2(1:Ncarriers,:),2); % Fem la mitja dels símbols
d'entrenament rebuts
Hs22=mean(TSdata2(Ncarriers+1:2*Ncarriers,:),2); % Fem la mitja
d'entrenament rebuts
```

```
for k=1:Ncarriers %Calculem la pseudomatriu per subportadora
h11=Hs11(k)/TSr1(k);h12=Hs21(k)/TSr2(k);
h21=Hs12(k)/TSr1(k);h22=Hs22(k)/TSr2(k);
z=[h11 h12; h21 h22];
b=pinv(z);
%Estimació de coeficients
t(:, :,k)=b*[fftdatarx1(:,k) fftdatarx2(:,k)].';
%treiem símbols "sobrants"
t1(:, :,k)=t(:, 1:end-Num1/Ncarriers,k);
%posem els TS a les 4 primeres files(2Tx*2bit/1TS)
t11(:, :,k)=reshape(t1(:, :,k), [],NTS);
%eliminem els TS
t12(:, :,k)=t11(Txs*bps+1:end, :,k);
t13(:,k)=reshape(t12(:, :,k), [],1);
t15(:,k)=reshape(t(:, end-Num1/Ncarriers+1:end,k), [],1);% en
% serie els símbols sobrants
t16(:,k)=[t13(:,k);t15(:,k)];%els afegim
end;

t18=reshape(transpose(t16),Txs*Ncarriers, []);

rdata(:,1)=reshape(t18(1:Ncarriers,:), [],1);
rdata(:,2)=reshape(t18(Ncarriers+1:end,:), [],1);

% 11) demapejar QPSK
hDemod = comm.QPSKDemodulator('BitOutput',true);
rec(:,1) = step(hDemod, rdata(:,1));
rec(:,2) = step(hDemod, rdata(:,2));

% 12.1) Mesura de la probabilitat d'error (Bit Error Rate)

[Nerr, BER1(ksnr)]=biterr(data(:,1), rec(:,1));
[Nerr, BER2(ksnr)]=biterr(data(:,2), rec(:,2));
end

% 13) Fer la gràfica de BER vs SNR
ebno=SNR-10*log10(bps); % Calcul de la Eb/No d'acord amb doc. Matlab
AWGN
ebnolin=10.^(ebno/10); % Passa Eb/No a linial
BERteorica=(1/bps)*erfc(sqrt(ebnolin)).*(1-.25*erfc(sqrt(ebnolin))); %
Formula teorica (Proakis/Salehi 5a ed; cap. 4, pags. 193/195)

% 13) Fer la gràfica de BER vs SNR
semilogy(SNR, BER1,SNR,BER2,SNR,BERteorica);
grid on;
xlabel('SNR (dB)');
ylabel('BER');
legend('Simulacions MIMO TX1','Simulacions MIMO TX2', 'Teoria');
```

7. Glossari

ADC- Analogue-to-Digital Converter

BER – Bit Error Rate

BPSK- Binary phase-shift keying

CD- Chromatic Dispersion

CP - Cyclic Prefix

CO-OFDM- Coherent Optical- Orthogonal Frequency Division Multiplexing

DAC - Digital-to-Analogue Converter

DGD- Differencial Group Delay

DSP – Digital Signal Processor

FIR- Finite Impulse Response

FFT – Fast Fourier Transform

GVD- Group Velocity Dispersion

ICI – Inter-Carrier Interference

IFFT - Inverse Fast Fourier Transform

ISI - Inter-Symbol Interference

MIMO- Multiple-input Multiple-output

MMSE- Minimum Mean Square Estimation

MZM - Mach-Zehnder Modulator

OFDM – Orthogonal Frequency Division Multiplexing

PAPR – Peak-to-Average Power Ratio

PBS- Polarization Beam Splitter

PBC- Polarization Beam Combiner

PDM- Polarization-division multiplexing

PDL- Polarization-Dependent Loss

PMD – Polarization Mode Dispersion

PSP- Principal State of Polarization

QAM- Quadrature Amplitude Modulation

QPSK- Quadrature phase-shift keying

RF – Radio Frequency

SITO- Single-Input Two-Output

SNR- Signal Noise Ratio

TITO- Two-Input Two-Output

TS- Training symbol

ZF- Zero Forcing

8. Bibliografia

"Orthogonal Frequency Division Multiplexing for Optical Communications". William Shieh, Ivan Djordjevic.

"Nonlinear Fiber Optics". Agrawal, G.P. (3rd Ed).Academic Press, 2001.

"Fiber-Optic Communications Systems". Agrawal, G P .3 Ed Wiley, 2002.

"Digital Communications", John Proakis , 5th Edition.

"Phenomenological approach to polarisation dispersion in long single-mode fibres" Poole, C.D. Wagner, R.E. Electronics Letters.

"Xarxes de fibra òptica". Salvatore Spadaro. Material docent UOC.

"Multiplexació per divisió en freqüències ortogonals (OFDM). Margarita Cabrera, Francesc Tarrés. Material docent UOC.

"Sistemes òptics de transmissió". Josep Maria Fàbrega. Material docent UOC.

"Polarisation mode dispersion mitigation in coherent optical orthogonal frequency division multiplexed systems". W. Shieh, W. Chen and R.S. Tucker

"Modelagem numérica de uma fibra óptica microestruturada para sensoriamiento distribuído de pressão" Rodrigo Cesar Riberio Miraglia.2010

"V-BLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel"P. W. Wolniansky, G. J. Foschini, G. D. Golden, R. A. Valenzuela.

"An Introduction to the Fundamentals of PMD in Fibers", White Paper Corning.

"SC341 OFDM for Optical Communications".Sanser L. Jansen.Nokia Siemens Network

"Real-Time Multi-Gigabit Receiver for Coherent Optical MIMO-OFDM Signals". Simin Chen, Qi Yang, Yiran Ma, and William Shieh