



**Escola Politècnica Superior  
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TREBALL DE FI DE CARRERA

**TÍTOL DEL TFC** Aplicació de la lògica difusa a un algoritme LMS en un entorn FPGA

**TITULACIÓ:** Enginyeria Tècnica de Telecomunicació, especialitat Sistemes de Telecomunicació

**AUTOR:** David Pertegal Miranda

**DIRECTOR:** Pere Lluís Gilabert Pinal

**DATA:** 12 de novembre de 2008



**Títol:** Aplicació de la lògica difusa a un algoritme LMS en un entorn FPGA

**Autor:** David Pertegal Miranda

**Director:** Pere Lluís Gilabert Pinal

**Data:** 12 de novembre de 2008

## Resum

Aquest treball fi de carrera presenta el disseny i desenvolupament d'un algoritme de lògica difusa que té per objectiu millorar l'adaptació d'un sistema de linealització basat en tècniques de predistorsió (PD) digital.

Aquest sistema predistorsionador, orientat a la linealització d'amplificadors de potència (AP), està format principalment per tres blocs: el bloc d'identificació, el de postdistorsió i el de predistorsió.

Les parts d'identificació i de postdistorsió estan formades per taules de memòria que emmagatzemen els guanys corresponents a funcions no-lineals, i incorporen cadascuna un bloc LMS, que és l'encarregat d'adaptar les funcions no lineals. L'adaptació d'aquestes funcions no lineals es basa en la minimització de l'error quadràtic mig definit com la diferència entre un senyal de referència i el senyal obtingut a partir del model d'identificació o de postdistorsió. L'algoritme triat per dur a terme aquesta minimització és el Least Mean Square (LMS).

El comportament d'aquest error en l'algoritme LMS varia en funció del valor d'un paràmetre, en un principi fix, al qual anomenem  $\mu$ . Segons el valor assignat a  $\mu$  es pot aconseguir una adaptació lenta però acurada dels paràmetres que formen la funció no-lineal o bé, en el cas contrari una adaptació ràpida dels paràmetres però limitada pel que fa al valor final de l'error, és a dir, menys acurada.

Com alternativa a fixar un valor per  $\mu$  es presenta en aquest TFC un algoritme Fuzzy-LMS. Aquest algoritme es basa en l'aplicació de tècniques de lògica difusa per fer variable el paràmetre  $\mu$  en funció de l'evolució de l'error d'adaptació de l'algoritme LMS.

**Title:** Fuzzy logic application to a LMS algorithm in a FPGA environment

**Author:** David Pertegal Miranda

**Director:** Pere Lluís Gilabert Pinal

**Date:** November, 12th 2008

## Overview

This project presents the design and development of a fuzzy logic algorithm that aims to improve the adaptation of a linearizer system based on digital predistortion techniques.

The Predistorter system, targeted to linearize power amplifiers, consists on three blocks: the identification block, the postdistortion block and the predistortion block.

Identification and postdistortion parts are formed by memory tables that store non-linear functions gains that are incorporated in a LMS block and whose aim is adapting the nonlinear functions. The adaptation of these non-linear functions is based on the minimization of the mean squared error defined as the difference between a reference signal and the signal obtained from the identification or postdistortion model. The chosen algorithm to perform this minimization is the Least Mean Squares (LMS).

The behavior of this error in the LMS algorithm varies depending on the fixed value of a parameter called  $\mu$ . Depending on the value assigned to  $\mu$  the adaptation can be achieved by a slow but accurate adjustment of the non-linear function parameters or in the other case a quick adjustment of the parameters but limited by the fact of the final value of the error, less accurate.

As an alternative to set a value for  $\mu$  a Fuzzy-LMS algorithm is presented in this project. This algorithm is based on the use of fuzzy logic techniques and allows the variability of the  $\mu$  parameter using the error adaptation of the LMS algorithm.

## DEDICATÒRIA

Primer de tot agrair al meu tutor Pere Lluís Gilabert la seva dedicació i constància. Sense ell no hauria estat possible la realització d'aquest TFC.

Per altra banda m'agradaria dedicar aquest projecte a totes les persones que m'han donat el seu suport incondicional durant tota la carrera.

En l'àmbit universitari Anna, Eulàlia, Eric, Romi, Barru, Guille, Pepe, Xavi, David, Orod, Juli, Xicu, Santi, Lídia, Elena, Mariona, Ali i els Alex amb els quals he viscut experiències molt agradables dintre i fora de la universitat.

Als meus amics Pablo, Xavi, Ruben, Pedro, Albert i Xana amb qui he crescut des de ben petit a l'escola.

També, en l'àmbit extraescolar, als meus amics Adrià, Gemma, Paloma, Edu, Mari, Rocío, Dámaris, Jona, Enric i Cristina.

I a tota la meua família, destacant a la meua germana Gemma i la meua mare Antonia que m'han de suportar cada dia, i en especial al meu pare Manuel que segurament es sent molt orgullós de mi.

# ÍNDEX

|  |           |
|--|-----------|
| <b>INTRODUCCIÓ .....</b>   | <b>9</b>  |
| <b>CAPÍTOL 1. PROBLEMÀTICA DELS AMPLIFICADORS DE POTÈNCIA .....</b>          | <b>11</b> |
| 1.1. Introducció .....   | 11        |
| 1.2. Linealitat i no linealitat d'un amplificador de potència.....           | 11        |
| 1.3. Efectes de la no linealitat i mesures per a la seva representació ..... | 12        |
| 1.4. Test de dos tons .....  | 13        |
| 1.5. Mesures de no linealitat per a senyals modulats.....                    | 15        |
| 1.6. Linealitat i eficiència .....   | 16        |
| <b>CAPÍTOL 2: TÈCNiques DE LINEALITZACIÓ .....</b>                           | <b>18</b> |
| 2.1. Introducció .....   | 18        |
| 2.2. Linealitzadors Feedback.....  | 18        |
| 2.3. Linealitzador Feedforward.....  | 18        |
| 2.4. Linealitzadors LINC i CALLUM.....                                       | 19        |
| 2.5. Linealització per predistorsió.....                                     | 19        |
| <b>CAPÍTOL 3: L'ALGORITME LMS .....</b>                                      | <b>21</b> |
| 3.1. Introducció .....   | 21        |
| 3.2. L'algoritme LMS.....  | 21        |
| 3.3. Aplicació de l'algoritme LMS al sistema Predistorsionador .....         | 22        |
| <b>CAPÍTOL 4: INTRODUCCIÓ A LA LòGICA DIFUSA.....</b>                        | <b>25</b> |
| 4.1. Introducció .....   | 25        |
| 4.2. Sèries difuses (Fuzzy sets).....  | 25        |
| 4.3. Funcions de pertinença (Membership functions) .....                     | 25        |
| 4.4. Operacions lògiques .....   | 26        |
| 4.5. If-Then rules .....   | 28        |
| <b>CAPÍTOL 5: INTRODUCCIÓ AL SYSTEM GENERATOR.....</b>                       | <b>29</b> |
| 5.1. Introducció .....   | 29        |

|  |   |           |
|--|---|-----------|
| 5.2.   | FPGA .....  | 29        |
| 5.3.   | Introducció al System Generator .....   | 30        |
| 5.4.   | Block Set de Xilinx .....   | 30        |
| 5.4.1  | Memòria RAM d'un port (Single port RAM).....                                      | 30        |
| 5.4.2.   | Memòria RAM de dos ports (Dual port RAM).....                                     | 31        |
| 5.4.3.   | Memòria compartida (Shared Memory).....   | 31        |
| 5.5.   | Suport per a Matlab .....   | 32        |
| 5.6.   | Plataforma d'integració.....  | 33        |
| <b>CAPÍTOL 6: DESCRIPCIÓ DE L'ARQUITECTURA DEL PREDISTORSIONADOR DIGITAL .....</b>         |   | <b>34</b> |
| 6.1.   | Introducció .....   | 34        |
| 6.2.   | Arquitectura del Predistorsionador digital .....                                  | 34        |
| 6.3.   | Comportament del Predistorsionador.....   | 36        |
| <b>CAPÍTOL 7: APLICACIÓ DE LA LòGICA DIFUSA A L'ALGORITME LMS ..</b>                       |   | <b>39</b> |
| 7.1.   | introducció .....   | 39        |
| 7.2.   | Establiment de les regles lingüístiques .....                                     | 39        |
| 7.3.   | Selecció de les funcions de pertinença .....                                      | 40        |
| 7.4.   | Taula de decisió.....   | 42        |
| 7.5.   | Resultats.....  | 43        |
| <b>CAPÍTOL 8: INCORPORACIÓ DE L'ALGORITME FUZZY-LMS AL SISTEMA PREDISTORSIONADOR .....</b> |   | <b>45</b> |
| 8.1.   | Introducció .....   | 45        |
| 8.2.   | Mètode d'incorporació de l'algoritme Fuzzy-LMS al sistema predistorsionador ..... | 45        |
| 8.3.   | Comportament de l'error quadràtic mig de l'LMS.....                               | 46        |
| 8.4.   | Inconvenients a l'utilitzar l'algoritme Fuzzy-LMS .....                           | 48        |
| <b>CONCLUSIONS.....</b>  |   | <b>51</b> |
| <b>BIBLIOGRAFIA .....</b>  |   | <b>52</b> |
| <b>ANNEXOS.....</b>  |   | <b>53</b> |
| A.1.   | Codi principal.....   | 53        |
| A.3.   | Funcions.....   | 58        |
| A.2.1.   | Funció per crear un senyal modulat en 16 QAM.....                                 | 58        |

|             |   |           |
|-------------|---|-----------|
| A.2.2.      | Funció per l'assignació de guanys a la LUT PA.....                                    | 59        |
| A.2.3.      | Funció per recuperar els resultats utilitzant l'algoritme basat en lògica difusa .... | 60        |
| <b>A.3.</b> | <b>Esquema del -Predistorsionador digital utilitzat en System Generator .....</b>     | <b>63</b> |
| A.3.1.      | Amplificador de potència .....  | 63        |
| A.3.2.      | Identificador .....   | 63        |
| A.3.3.      | Postdistorsionador.....   | 64        |
| A.3.4.      | Predistorsionador .....   | 64        |



## INTRODUCCIÓ

Aquest TFC té com a objectiu millorar l'algoritme LMS (Least Mean Square) per aconseguir una òptima adaptació en diverses fases d'un sistema de linealització d'Amplificador de Potència (APs) basat en tècniques de predistorsió digital.

En l'algoritme LMS és necessari destacar la rellevància del valor numèric assignat al paràmetre de pas  $\mu$ . Aquest paràmetre està vinculat a l'adaptació de la resposta de l'algoritme, és a dir, la convergència de l'error del sistema adaptatiu es comportarà de manera diferent depenent del valor de  $\mu$ .

Un valor de  $\mu$  alt proporciona una velocitat de convergència de l'error quadràtic mig elevada, en canvi mai arriba a aconseguir una adaptació acurada. En el cas d'una assignació de valor de  $\mu$  baix es garanteix una bona adaptació però el temps de convergència és major que en el cas anterior.

Per aquest motiu, a l'hora d'implementar un sistema LMS sempre s'entra en el compromís de la tria del paràmetre de pas  $\mu$ .

En aquest TFC es proposa un sistema LMS en el qual el paràmetre de pas és variable. Aquesta variabilitat es fa possible mitjançant un algoritme basat en la lògica difusa, a la combinació d'aquests dos algoritmes l'anomenem Fuzzy-LMS.

Així doncs, el TFC està organitzat de la següent manera:

En el capítol 1 es presenta la problemàtica associada a la distorsió no-lineal que introdueixen els APs en un sistema de transmissió, enumerant i explicant les no linealitats més importants.

En el capítol 2 s'esmenten les tècniques principals per a contrarestar les no linealitats descrites en el primer capítol, aprofundint concretament en el mètode de predistorsió digital, l'utilitzat en el nostre sistema de linealització.

El capítol 3 presenta una introducció a l'algoritme LMS utilitzat en el sistema predistorsionador.

El capítol 4 introdueix el concepte de lògica difusa explicant les diferents passes a seguir a l'hora de establir una regla basada en aquest tipus de lògica.

En el capítol 5 es presenta l'entorn de disseny i simulació del System Generator, utilitzat per al desenvolupament del predistorsionador digital. En aquest capítol es defineixen els blocs més importants que constitueixen l'arquitectura del predistorsionador, així com també es defineix el concepte processament en paral·lel que permet el dispositiu (*Field-Programmable Gate Array*) FPGA.

El capítol 6 detalla l'arquitectura, el funcionament i el comportament del predistorsionador digital implementat mitjançant el System Generator.

Al capítol 7 s'explica la implementació de l'algoritme Fuzzy-LMS, on es proporcionen diversos resultats de simulacions que permeten comparar les diferents prestacions obtingudes al variar els paràmetres de l'algoritme.

El capítol 8 descriu el procés d'incorporació de l'algoritme Fuzzy-LMS al sistema de predistorsionador i amplificador explicant les dificultats que s'han trobat en el procés. Finalment es comenten els resultats obtinguts.

# CAPÍTOL 1. PROBLEMÀTICA DELS AMPLIFICADORS DE POTÈNCIA

## 1.1. Introducció

En aquest apartat es pretén fer una introducció de la problemàtica que introdueixen els APs en un sistema de comunicacions. Es presentarà el comportament no lineal d'aquests i els motius de la seva existència. També es definiran els paràmetres més significatius per a poder avaluar aquestes no linealitats. I per últim es definiran conceptes com l'eficiència i s'enumeraran les diferents classes d'amplificadors segons el mode d'operació, o la seva linealitat i eficiència.

## 1.2. Linealitat i no linealitat d'un amplificador de potència

Un AP lineal es defineix com aquell el qual la seva sortida és un múltiple escalar de la seva entrada [1]. Definint el voltatge de sortida com a  $V_{out}(t)$ , el voltatge d'entrada com a  $V_{in}(t)$  i el seu guany com a  $g$ , aquesta afirmació es pot expressar de la següent manera:

$$V_{out}(t) = g \cdot V_{in}(t) \quad (1.1)$$

En la realitat els APs no són dispositius ideals pel que presenten distorsió no lineal, com per exemple la distorsió harmònica, la distorsió per productes de intermodulació i la distorsió dintre de banda.

La distorsió harmònica es pot definir com l'aparició de components freqüencials addicionals a múltiples de la freqüència portadora de la senyal d'entrada.

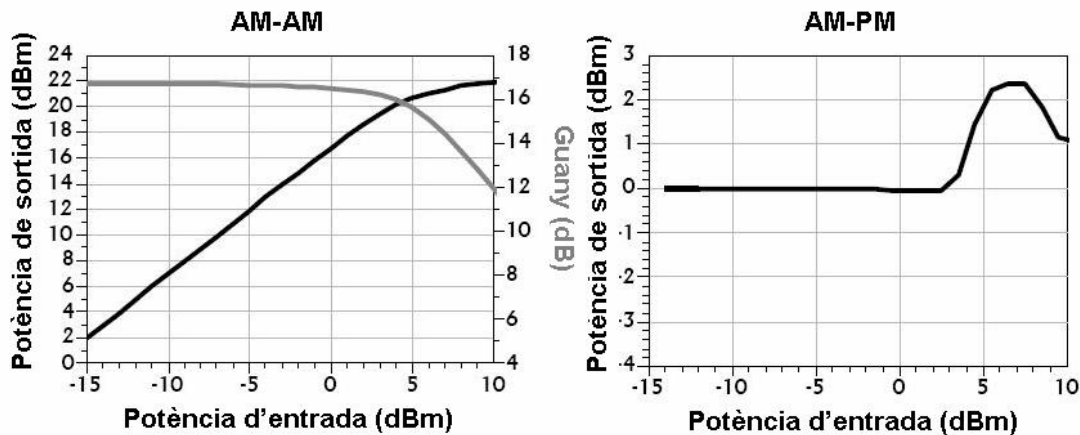
La distorsió per productes d'intermodulació consisteix en la formació de components freqüencials a la vora de la banda del senyal desitjat. Aquests productes d'intermodulació no es poden eliminar utilitzant el filtratge.

La distorsió de banda es genera a partir de combinacions no lineals que generen components freqüencials que cauen dins l'ample de banda del senyal de treball.

### 1.3. Efectes de la no linealitat i mesures per a la seva representació

Considerant senyals modulades que contenen informació a les components de fase i quadratura, hauríem de tenir en compte la necessitat d'obtenir informació també de la distorsió soferta a ambdues d'aquests components.

Per a la representació d'aquesta distorsió es solen utilitzar les anomenades corbes AM-AM i AM-PM. La corba AM-AM representa la conversió de l'amplitud del senyal d'entrada en funció de l'amplitud del senyal de sortida. Aquesta representació mostra la relació de no linealitat entre el senyal d'entrada i el de sortida. La corba AM-PM representa la conversió de l'amplitud del senyal d'entrada en funció de la diferència de fases entre l'entrada i la sortida (Figura 1.1)[1][2].

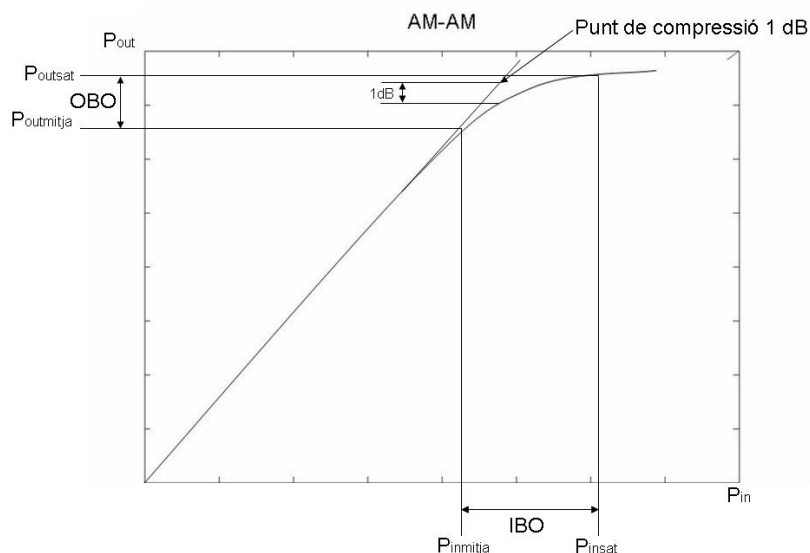


**Figura 1.1:** Corbes de la característica AM-AM i AM-PM d'un amplificador de potència (extreta de [1])

Un altre factor utilitzat per a la caracterització de la no linealitat dels APs és el punt de compressió 1dB. Aquest factor es correspon al punt on la corba real d'amplificació és 1dB per sota de la corba d'un amplificador lineal ideal.

Una altra mesura important és la relació entre el valor mig i el valor de pic de potència (*Peak to Average Power Ratio* - PAPR) dels senyals. Aquesta mesura és útil per identificar la zona treball òptima de l'amplificador i així poder evitar la saturació. Sent  $P_{pic}$  la potència de pic i  $P_{mitja}$  la potència mitja, el valor de PAPR es calcula de la següent manera:

$$PAPR[dB] = 10 \cdot \log_{10} \left( \frac{P_{pic}}{P_{mitja}} \right) \quad (1.2)$$



**Figura 1.2:** Representació de les mesures i els paràmetres més importants d'un amplificador de potència

Cal esmentar també els paràmetres de input back-off (IBO) i output back-off (OBO). Aquests ens delimiten el marge entre la zona lineal i el punt de saturació a l'entrada en el cas de l'IBO i a la sortida en el cas de l'OBO.

En les següents fórmules veiem la relació entre l'IBO i les potències de entrada de saturació i mitja,  $P_{in_{sat}}$  i  $P_{in_{mitja}}$  respectivament, i la relació entre l'OBO i les potències de sortida de saturació i mitja,  $P_{out_{sat}}$  i  $P_{out_{mitja}}$  respectivament.

$$IBO[dB] = P_{in_{sat}}[dBm] - P_{in_{mitja}}[dBm] \tag{1.3}$$

$$OBO[dB] = P_{out_{sat}}[dBm] - P_{out_{mitja}}[dBm] \tag{1.4}$$

### 1.4. Test de dos tons

Per a la caracterització de la no linealitat d'un amplificador es fa servir una prova consistent en generar dos tons a freqüències  $w_1$  i  $w_2$ , amb un marge de separació  $\Delta w$ , i utilitzar-los per alimentar l'amplificador.

A la sortida de l'amplificador obtenim com a resultat diversos components freqüencials que es corresponen amb cada una de les no linealitats comentades en l'apartat 1.2:

La distorsió dins de la banda treball es correspon amb l'aparició de components com:

$w_1 + w_1 - w_1 = w_1$  i  $w_2 + w_2 - w_2 = w_2$  es denominen freqüències de compressió.

$w_1 + w_2 - w_2 = w_1$  i  $w_2 + w_1 - w_1 = w_2$  es denominen freqüències de captura.

Els següents components es corresponen amb els harmònics:

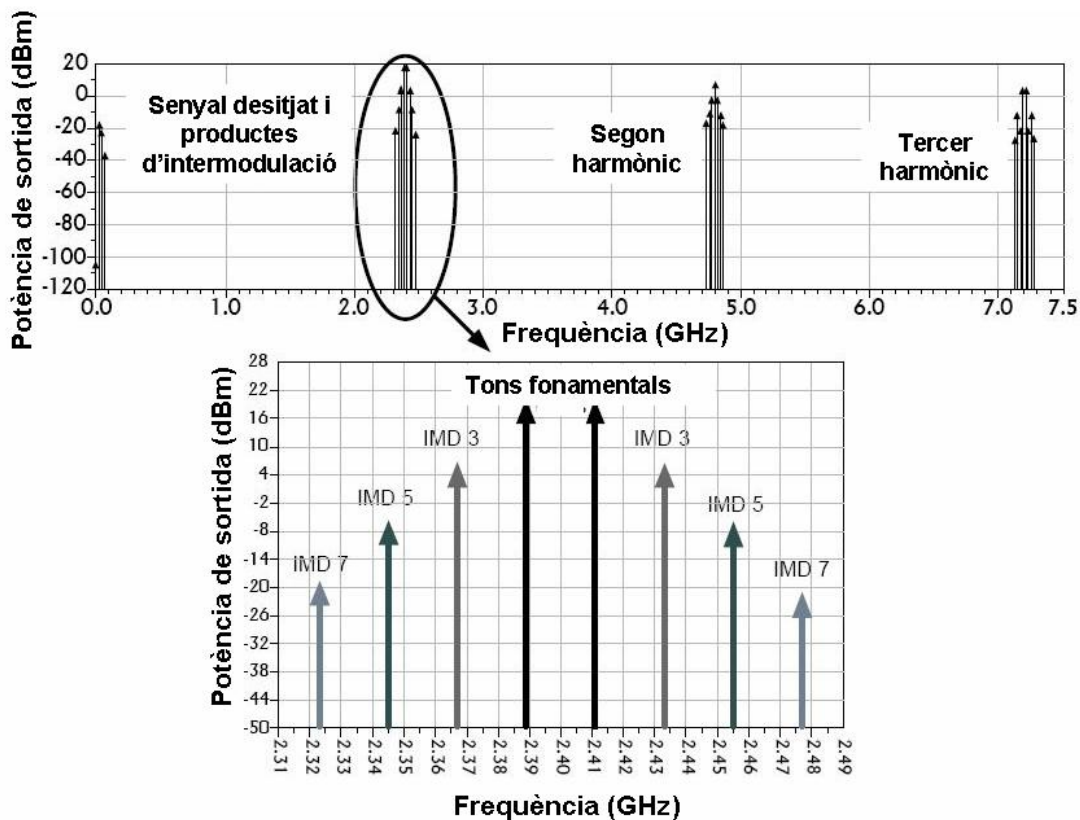
$2w_1$  i  $2w_2$  en el cas del segon ordre.

$3w_1$  i  $3w_2$  en el cas del tercer ordre.

Per últim representem la distorsió per productes d'intermodulació:

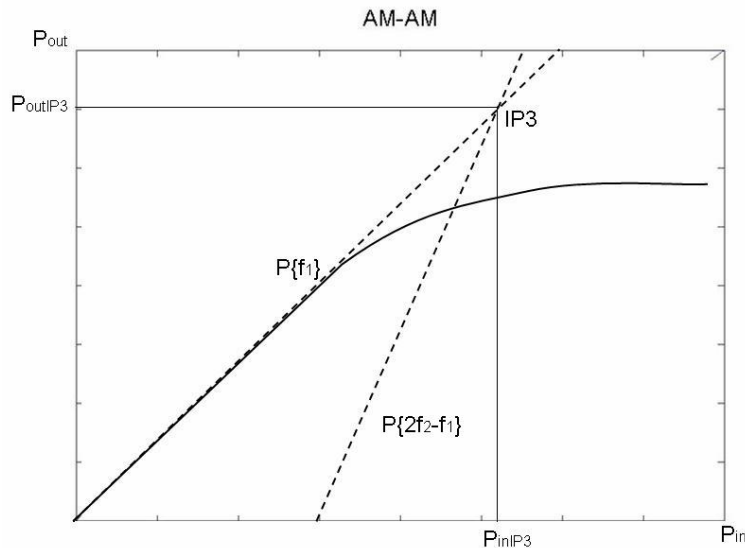
$w_1 + w_2$  i  $w_1 - w_2$  en el cas del segon ordre.

$2w_1 + w_2$ ,  $2w_1 - w_2$  i  $2w_2 + w_1$ ,  $2w_2 - w_1$  en el cas del tercer ordre.



**Figura 1.3:** Espectre del resultat d'un test de dos tons (extreta de [1])

L'IP3 és el punt on es creuarien la corba de relació AM-AM d'un AP lineal i la del producte d'intermodulació de tercer ordre. La relació AM-AM del producte d'intermodulació creix tres vegades més ràpid que la de l'amplificador.



**Figura 1.4:** Representació del punt d'intercepció IP3

La relació IMD3 també anomenada  $C/I_3$  es defineix com la relació entre la potència del to fonamental  $P_{out}\{f_1\}$  i la del to produït pel producte de intermodulació d'ordre 3  $P_{out}\{2f_2 - f_1\}$ .

$$C / I_3 = \frac{P_{out}\{f_1\}}{P_{out}\{2f_2 - f_1\}} \tag{1.5}$$

### 1.5. Mesures de no linealitat per a senyals modulats

Quan els senyals amb els que tracta un amplificador són modulats, tenen un espectre que presenten un determinat ample de banda de senyal i no tant sols un to. En aquests tipus de senyals també es poden veure els efectes de les no linealitats, les quals tenen un espectre continu en una determinada banda. Per a la mesura de la quantitat d'espectre que es troba fora de la banda del senyal principal s'utilitza l' ACPR (*Adjacent Channel Power Ratio*), que és la relació entre la potència que cau fora de banda  $P_{adjacent-channel}$  i la que està dintre  $P_{in-band}$ .

$$ACPR = \frac{P_{in-band}}{P_{adjacent-channel}} [dB] \tag{1.6}$$

En un senyal modulats es pot identificar el grau de la seva distorsió dintre de la banda del senyal amb el que s'anomena la magnitud del vector d'error (*Error Vector Magnitude* - EVM). Aquest càlcul determina la desviació dels símbols de la constel·lació respecte a la seva posició original, aquesta desviació rep el nom

de vector d'error. L'EVM és la relació entre la potència del vector d'error  $P_{error}$  i la potència del senyal ideal transmès  $P_{reference}$ .

$$EVM = 10 \log_{10} \frac{P_{error}}{P_{reference}} [dB] \quad (1.7)$$

## 1.6. Linealitat i eficiència

L'eficiència d'un AP és un factor crític per a la duració de la seva bateria. Els amplificadors són els elements que més energia consumeixen als circuit de RF, poden representar fins a un 70% de l'energia total consumida.

L'eficiència d'un AP es pot definir com la quantitat de potència DC que arriba a ser transformada en potència RF. Sent  $P_{out}$  la potència de sortida i  $P_{DC}$  la DC l'eficiència  $\eta$  es pot definir amb la següent fórmula:

$$\eta = \frac{P_{out}}{P_{DC}} [\%] \quad (1.8)$$

També es pot considerar només la quantitat de potència neta que és transformada a potència RF, en aquest cas no s'hauria de tenir en compte la potència RF introduïda a l'AP. Seguint aquest criteri obtenim l'eficiència de potència afegida (*Power Added Efficiency* - PAE) la qual es pot expressar amb la següent fórmula:

$$PAE = \frac{P_{in} - P_{out}}{P_{DC}} [\%] \quad (1.9)$$

L'eficiència d'un amplificador és òptima en el cas de treballar amb el pic de la potència de sortida a prop de la zona de saturació de l'amplificador. La linealitat en la zona de saturació de l'amplificador és però mínima. Aquest fet posa de manifest el compromís existent entre eficiència i linealitat. La zona de treball no és l'únic factor que determina la eficiència d'un amplificador, la classe de transistor de potència també influeix en aquest aspecte.

Segons la relació eficiència linealitat i la manera de treballar del transistor s'han definit diverses classes d'amplificadors. Les classes definides com A, AB, B i C tenen en comú l'actuació del transistor com a font de corrent, mentre que en les classes D, E i F actua com un interruptor. A la Taula 1.1 podem veure la classificació dels diferents APs segons els tres paràmetres esmentats.



| <b>Classe d'amplificador</b> | <b>Mode d'operació</b> | <b>Eficiència màxima [%]</b> | <b>Linealitat</b>           |
|------------------------------|------------------------|------------------------------|-----------------------------|
| Classe A                     | Mode font de corrent   | 50                           | Bona                        |
| Classe AB                    |                        | Millor que A i pitjor que B  | Millor que A i pitjor que B |
| Classe B                     |                        | 78,5                         | Mitja                       |
| Classe C                     |                        | 100                          | Pobra                       |
| Classe D                     | Mode interruptor       | 100                          | Pobra                       |
| Classe E                     |                        | 100                          | Pobra                       |
| Classe F                     |                        | 100                          | Pobra                       |

**Taula 1.1:** Relació entre les diferents classes d'amplificador [1]

## CAPÍTOL 2: TÈCNiques DE LINEALITZACIÓ

### 2.1. Introducció

Existeixen diferents tècniques orientades a proveir la màxima linealitat en els APs. En aquest capítol enumerarem i descriurem breument cada una d'aquestes tècniques i explicarem amb més detall la de predistorsió digital ja que és la que utilitzem en el nostre sistema.

Les tècniques de linealització es poden dividir en dos grups principals segons el propòsit del linealitzador: els linealitzadors orientats a reduir la distorsió i els linealitzadors orientats a evitar la distorsió.

Els linealitzadors per reduir distorsió es caracteritzen per ser tècniques utilitzades per reduir la distorsió a la sortida de l'AP. Hi ha dues tècniques basades en aquest mètode: Feedback i Feedforward.

Els linealitzadors per evitar la distorsió utilitzen tècniques de processat portades a terme abans del pas del senyal per l'AP amb l'objectiu d'aconseguir un amplificació lineal a la sortida del sistema. Hi ha quatre tècniques basades en aquest mètode: Power back-off tuning, LINC/CALLUM i la Predistorsió.

### 2.2. Linealitzadors Feedback

Aquest linealitzador es basa en les tècniques clàssiques de realimentació. La realimentació de la sortida del circuit proveeix al sistema d'una gran robustesa davant possibles pertorbacions, i l'estabilitat del sistema depèn dels anomenats marge de guany i marge de fase del llaç de realimentació[1][2].

El mètode per guanyar linealitat és aconseguir la magnitud òptima de la funció de transferència del circuit de realimentació. L'inconvenient de les tècniques de realimentació és que el que es guanya de linealitat es perd en guany d'amplificació i d'altra banda, aquesta tècnica és només aplicable en senyals amb amples de banda reduïts.

### 2.3. Linealitzador Feedforward

El sistema Feedforward és una eficaç tècnica de linealització, sobretot aplicada als sistemes de comunicació amb elevats amples de banda, com poden ser els senyals OFDM multiportadora.

El circuit d'alimentació Feedforward està compost per dos parts diferenciables: el circuit de la cancel·lació del senyal i el circuit de la cancel·lació d'error. Al primer d'aquest obtenim un senyal d'error el qual en la segona part del circuit

amplificarem i injectarem en contrafase per eliminar els productes d'intermodulació presents a la sortida de l'AP[1][2]. Un dels inconvenients dels sistemes de linealització basats en la tècnica feedforward és la sensibilitat dels llaços de cancel·lació del senyal i de l'error. Qualsevol variació o desajustament en els llaços implica una degradació significativa de la linealitat de tot el sistema.

## 2.4. Linealitzadors LINC i CALLUM

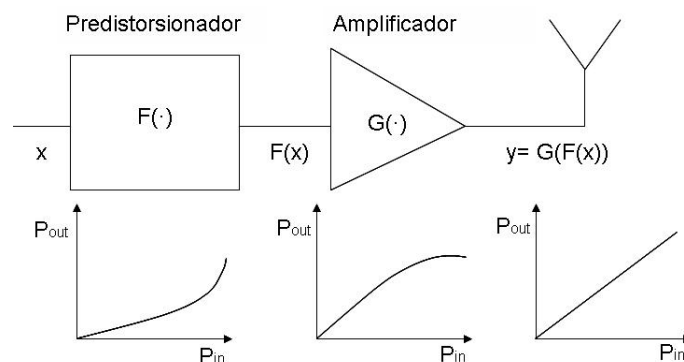
Els sistemes LINC (Linear Amplification using Non-linear Components) es basen en la divisió del senyal en dues d'amplitud constant. Aquestes dues senyals són amplificades independentment i després són novament sumades obtenint el resultat desitjat sense distorsió[1][2].

El sistema CALLUM (Combined Analog-Locked Loop Universal Modulator) és la versió realimentada del LINC. Un dels problemes associats a aquestes dues tècniques és la limitació que tenen pel que fa al processat de senyals amb grans amplituds de banda, ja que requereixen de processadors extremadament ràpids per dur a terme la separació i processat dels senyals amb amplitud constant[1][2].

## 2.5. Linealització per predistorsió

La linealització per predistorsió (PD) consisteix en utilitzar un predistorsionador abans de l'AP amb l'objectiu de contrarestar els efectes de les no linealitats de l'AP. El mètode per anular aquesta no linealitat és fer que el predistorsionador tingui un comportament invers al de l'AP, aconseguint d'aquesta manera una resposta lineal combinant els dos comportaments.

La figura 3.1 mostra l'esquema bàsic d'un sistema de predistorsionador i amplificador amb les seves corbes de relació AMAM.

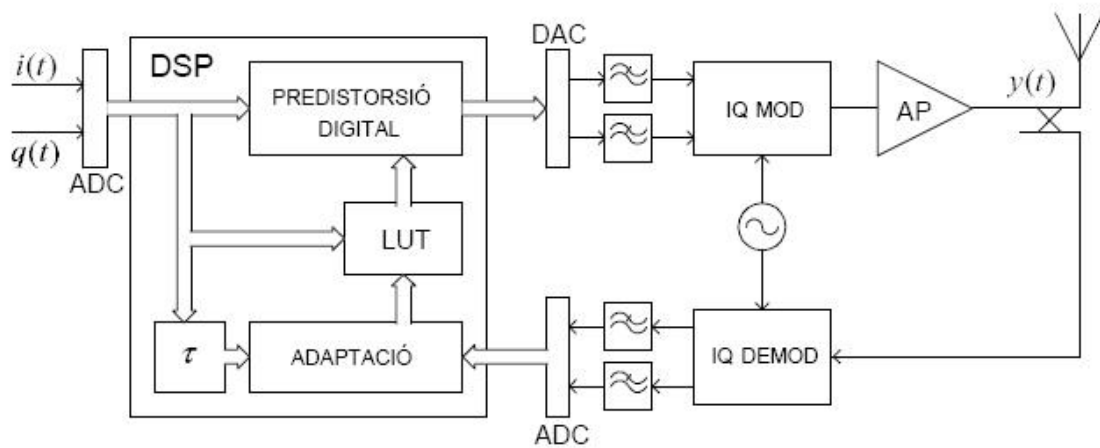


**Figura 3.1:** Esquema bàsic d'un predistorsionador

La gran majoria de predistorsionadors incorporen un mecanisme de realimentació amb la finalitat d'aconseguir una predistorsió adaptativa que permet una robusta actuació del linealitzador.

Segons la posició que ocupi el predistorsionador al sistema transmissor la predistorsió pot tenir lloc en radio freqüència, freqüència intermitja o banda base. La predistorsió es pot classificar en analògica o digital. En el nostre cas ens centrarem en la predistorsió digital en banda base [1][2].

La predistorsió digital en banda base fa servir diversos dispositius de processat digital com els '*digital signal processors*' (DSPs) o els dispositius *Field Programmable Gate Arrays* (FPGAs). La Figura 3.2 mostra l'esquema d'un sistema de predistorsió digital en banda base.



**Figura 3.2:** Esquema d'un predistorsionador digital (figura extreta de [2])

## CAPÍTOL 3: L'ALGORITME LMS

### 3.1. Introducció

En aquest capítol s'explicarà la utilitat de l'algoritme LMS i es farà una descripció d'aquest per passos. També explicarem la metodologia seguida per a la seva aplicació al PD per mitjà de LUTs (Look Up Tables).

### 3.2. L'algoritme LMS

L'algoritme LMS és utilitzat en funcions no lineals per trobar els coeficients d'aquestes que permeten obtenir el valor esperat mínim del quadrat del senyal d'error, aquest error es defineix com la diferència entre el senyal desitjat i el senyal obtingut.

Aquest algoritme pertany a la família dels estocàstics, és a dir, la funció no lineal s'adapta en funció de l'error a l'instant actual únicament.

Una característica important del LMS és la seva simplicitat. No requereix mesures de les funcions de correlació, ni tampoc inversió de la matriu d'autocorrelació.

L'algoritme LMS comprèn dos processos bàsics:

- Un procés de càlcul de la sortida generada per la funció no lineal, y la generació d'una estimació de l'error.
- Un procés adaptatiu, que realitza l'ajust automàtic dels coeficients de la funció no lineal d'acord amb l'estimació de l'error.

La funció de sortida estimada  $\hat{y}(n)$  ve donada en funció d'una entrada complexa  $x(n)$ . Es pot representar de forma polinòmica de la següent manera:

$$\hat{y}(n) = \sum_{i=1}^p w_i \cdot x(n) |x(n)|^i \quad (3.1)$$

$$\hat{y}(n) = w_0 \cdot x(n) + w_1 \cdot x(n) |x(n)| + w_2 \cdot x(n) |x(n)|^2 + \dots + w_p \cdot x(n) |x(n)|^p \quad (3.2)$$

També podem representar la sortida en forma matricial

$$\hat{y}(n) = [w_0 w_1 w_2 \dots w_p]^* \cdot \begin{bmatrix} x(n) \\ x(n) | x(n) | \\ x(n) | x(n) |^2 \\ \dots \\ x(n) | x(n) |^p \end{bmatrix} \quad (3.3)$$

$$\hat{y}(n) = \bar{w}^H \cdot \bar{x} \quad (3.3)$$

Sent  $w$  els coeficients de la funció complexa d'entrada. Els superíndex  $*$  i  $H$  representen el conjugat i l'Hermític (transposat i conjugat) respectivament. El guió sobre una variable representa vector.

El senyal d'error és la diferència entre la funció de sortida real  $y(n)$  i la funció de sortida estimada  $\hat{y}(n)$ .

$$e(n) = y(n) - \hat{y}(n) \quad (3.4)$$

$$e(n) = y(n) - \bar{w}^H \cdot \bar{x} \quad (3.5)$$

La funció d'adaptació dels coeficients de la funció que obtenim mitjançant l'algoritme LMS és la següent:

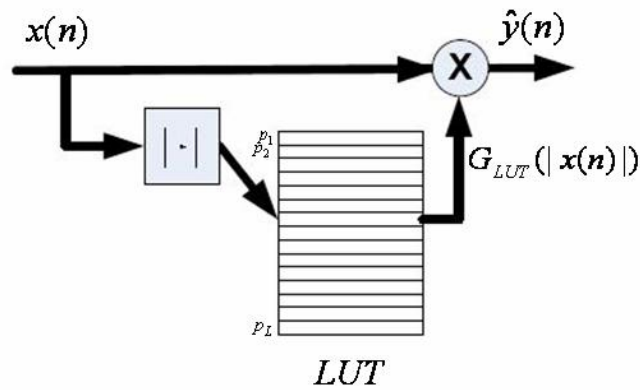
$$\bar{w}_{n+1} = \bar{w}_n + \mu \cdot \bar{x} \cdot e^*(n) \quad (3.6)$$

### 3.3. Aplicació de l'algoritme LMS al sistema Predistorsionador

En l'apartat 3.2 hem explicat l'algoritme LMS aplicat a funcions no lineals amb l'objectiu de trobar els coeficients òptims que minimitzin l'error quadràtic mig definit entre una sortida desitjada i la sortida del filtre adaptatiu.

En el sistema predistorsionador proposat en el capítol 6 no farem ús de l'LMS de la manera explicada prèviament, sinó que l'utilitzarem per a trobar els guanys òptims  $G_{LUT}$  que emmagatzemarem en una LUT. A continuació explicarem aquesta metodologia.

La figura mostra l'esquema bàsic d'entrada sortida a una LUT de guanys



**Figura 3.1:** Esquema bàsic d'entrada sortida d'una LUT de Guany

La variable  $p$  representa la posició de la LUT i el subíndex  $L$  la longitud de la LUT, és a dir, el nombre de guany que pot emmagatzemar.

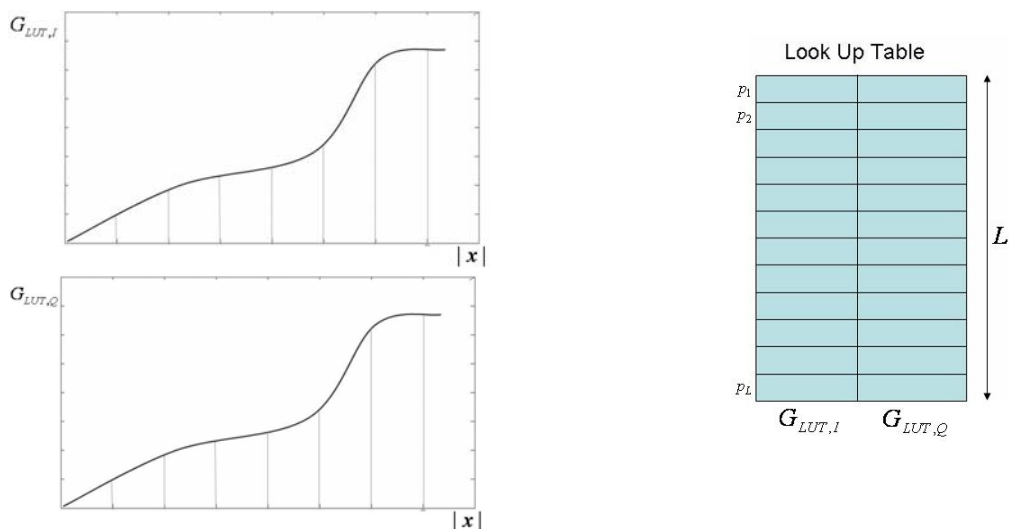
En aquest cas la funció de sortida estimada  $\hat{y}(n)$  es representaria de la següent manera:

$$\hat{y}(n) = x(n) \cdot G_{LUT}^*(|x(n)|) \tag{3.7}$$

$$e(n) = y(n) - x(n) \cdot G_{LUT}^*(|x(n)|) \tag{3.8}$$

La funció d'adaptació dels coeficients ve donada de la següent manera:

$$G_{LUT,p_L}^{n+1} = G_{LUT,p}^n + \mu \cdot x(n) \cdot e^*(n) \tag{3.9}$$



**Figura 3.2:** Representació de la indexació de Guanys a la LUT

La Figura 3.2 representa la indexació de la LUT de guanys de la funció  $x$ . Els guanys emmagatzemats a la LUT són complexos. És per això que es representen les parts reals i imaginàries per separat.



## CAPÍTOL 4: INTRODUCCIÓ A LA LòGICA DIFUSA

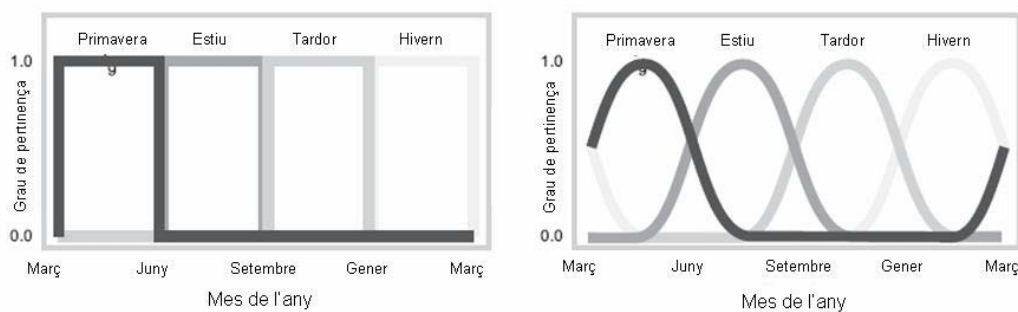
### 4.1. Introducció

La lògica difusa es basa en la idea de que en un instant donat no es possible precisar el valor d'una variable qualsevol, sols es pot conèixer el grau en el que pertany a cadascun dels conjunts en els quals ha participat el rang de variació de la variable[3].

### 4.2. Sèries difuses (Fuzzy sets)

La lògica difusa comença amb el concepte d'una sèrie difusa. Aquest tipus de sèries es caracteritzen pel fet de no tenir una frontera definida, és a dir, no es distingeixen clarament els límits de cada un dels elements de la sèrie.

Posarem un exemple de sèrie difusa utilitzant les estacions de l'any (Figura 4.1). Cada estació de l'any comença un dia determinat i acaba en un altre, per tant es té coneixement dels límits de les estacions i es podria representar com a la figura de l'esquerra. En canvi en la realitat el que notem és que les estacions varien de forma més gradual, com el que veiem a la figura de la dreta.



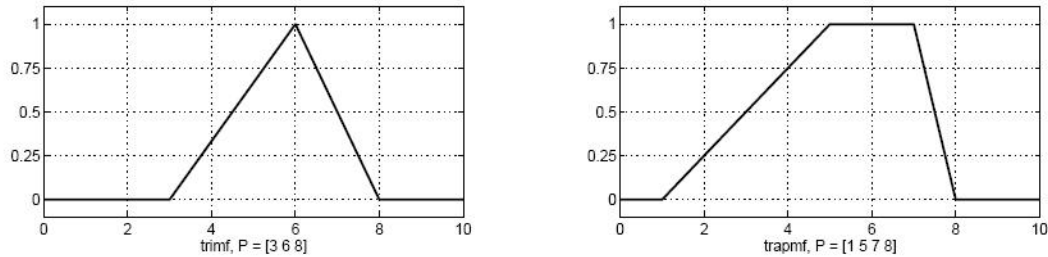
**Figura 4.1:** Representació de l'evolució de les estacions de l'any com una sèrie booleana i una sèrie difusa (extreta de [3])

### 4.3. Funcions de pertinença (Membership functions)

La funció de pertinença o 'membership function' (MF) és una corba que quantifica el grau de pertinença de la variable als elements d'una sèrie.

Les MF més simples són les formades utilitzant línies rectes. D'aquest tipus en tenim: la funció triangular, que és simplement la unió de tres punts, i la funció

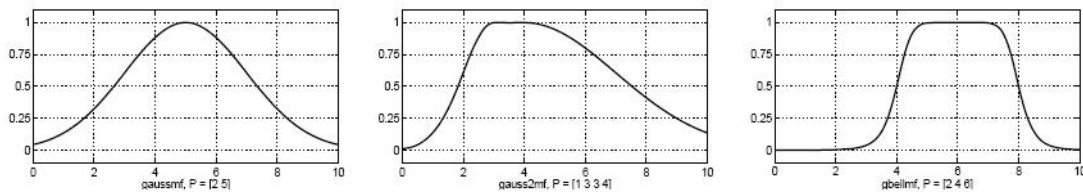
trapezoïdal que es forma a partir de quatre punts i caracteritza per ser horitzontal a la seva part superior (Figura 4.2).



**Figura 4.2:** Funcions triangular i trapezoïdal (extreta de [3])

Dues MF són formades a partir de la funció de distribució Gaussiana: la Gaussiana simple i composta (Figura 4.3). La funció gaussiana simple es caracteritza per tenir forma arrodonada i ser simètrica respecte l'eix vertical. Conté un paràmetre anomenat variància que determina l'amplada de la funció. En el cas de la funció gaussiana composta es requereix dos valors de variància, un que indica el grau de pujada i l'altre el de baixada.

La MF Campana està composta per tres paràmetres, és semblant a la Gaussiana simple però a la part superior està truncada. Aquestes funcions es caracteritzen per tenir una forma corbada i tenen l'avantatge de no tenir cap punt amb valor zero.



**Figura 4.3:** Funcions gaussiana, gaussiana composta i campana (extreta de [3])

#### 4.4. Operacions lògiques

En aquest apartat veurem de quina manera es connecten les MF amb les operacions lògiques.

Si mantenim els valors difusos als seus extrems (1 ó 0) les operacions lògiques es mantindran.

| A | B | A and B |
|---|---|---------|
| 0 | 0 | 0       |
| 0 | 1 | 0       |
| 1 | 0 | 0       |
| 1 | 1 | 1       |

**AND**

| A | B | A or B |
|---|---|--------|
| 0 | 0 | 0      |
| 0 | 1 | 1      |
| 1 | 0 | 1      |
| 1 | 1 | 1      |

**OR**

| A | not A |
|---|-------|
| 0 | 1     |
| 1 | 0     |

**NOT**

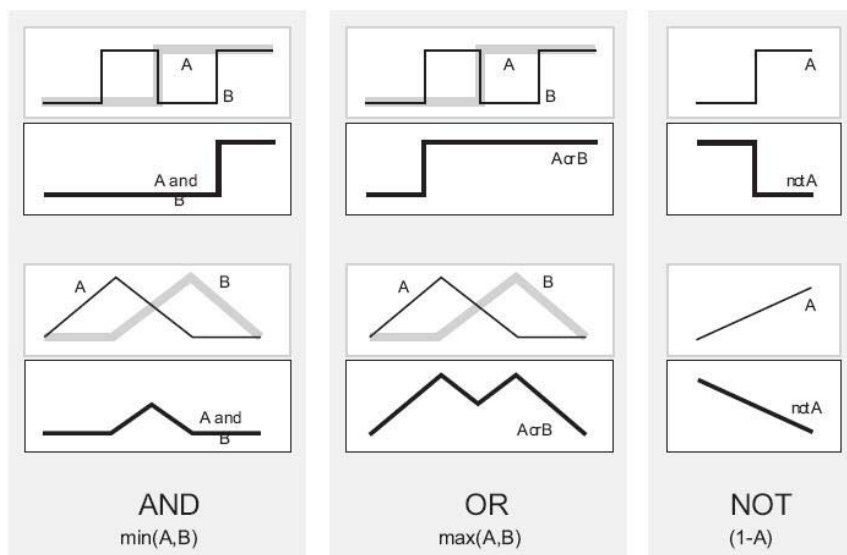
**Figura 4.4:** Taules de la veritat de les operacions lògiques AND, OR i NOT (extreta de [3])

Necessitem preservar els resultats de les taules de la veritat de la Figura 4.4 però amb valors entre 0 i 1. Per això fem ús de les següents funcions, les quals són equivalents a les tres operacions lògiques esmentades.

Per a l'operació **AND** la funció utilitzada és **min**. Aleshores **A AND B** es transformaria en **min(A,B)**. L'operació **OR** és substituïda per la funció **max**, per tant **A OR B** es transforma en **max(A,B)**. I per últim, l'operació equivalent a **NOT A** seria **1- A**.

Si es substitueixen les operacions lògiques de la taula anterior amb les funcions descrites veiem que obtenim els mateixos resultats.

La figura 4.5 demostra aquesta teoria fent ús de sèries difuses en comptes de la taula de la veritat. La part superior de la taula representa funcions de dos valors procedent de la taula de la veritat (A i B), mentre que la part inferior mostra el funcionament de les operacions amb funcions de rang de valors entre A i B.



**Figura 4.5:** Relació entre funcions i operacions lògiques (extreta de [3])

## 4.5. If-Then rules

Les regles If-Then s'utilitzen per a formular estats condicionals que comprenen la lògica difusa.

Una regla If-Then senzilla té la forma:

Si  $x$  és **A** y és **B**

On  $A$  i  $B$  són valors definits per sèries difuses amb rangs  $x$  i  $y$  respectivament. La part "x és A" s'anomena premissa, en canvi la part "y és B" és la conclusió.

Un exemple d'aquesta regla pot ésser:

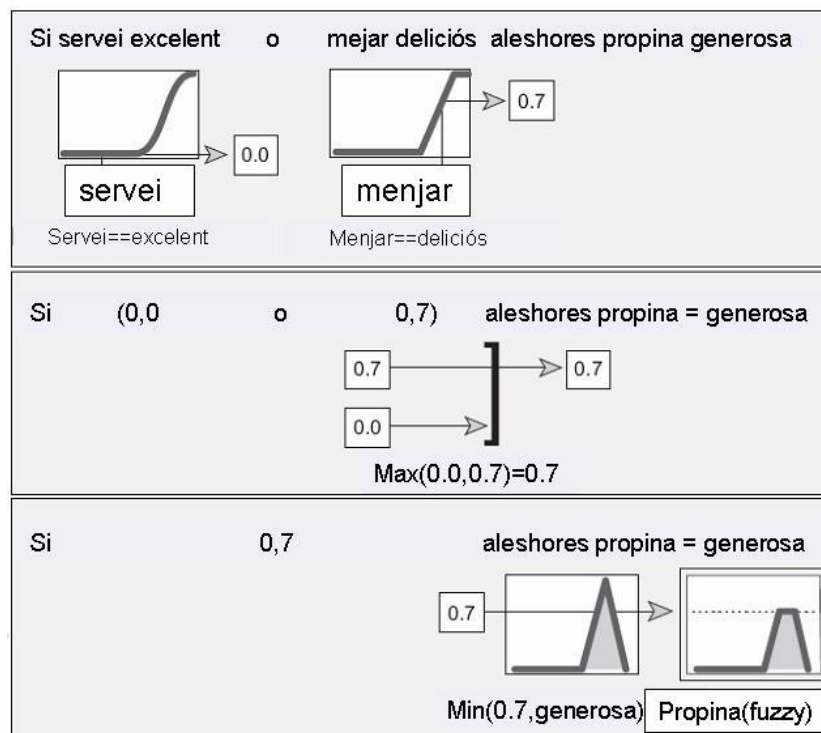
*Si el servei és bo la propina és mitjana*

El concepte *bo* es pot representar amb un número entre 0 i 1. En canvi el terme *mitjana* es correspon amb una sèrie difusa.

Interpretar una regla if-then comporta diverses parts: primer avaluar la premissa i després aplicar aquesta avaluació a la conseqüència.

La premissa pot tenir diverses parts les quals s'han de relacionar entre elles i amb la conseqüència amb operacions lògiques.

A la Figura 4.6 veiem un exemple complet de l'establiment de regles If-Then.



**Figura 4.6:** Exemple de l'establiment de regles If-Then (extreta de [3])

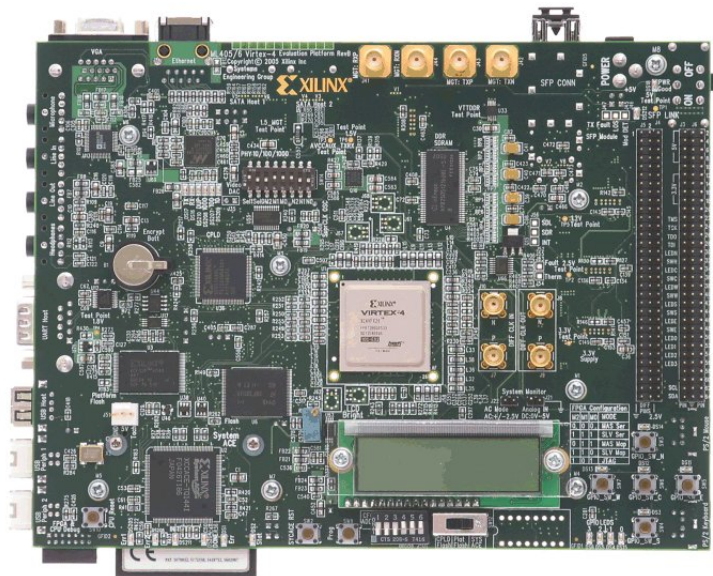
## CAPÍTOL 5: INTRODUCCIÓ AL SYSTEM GENERATOR

### 5.1. Introducció

En aquest apartat s'explicarà el concepte de FPGA i es farà una introducció al System Generator enumerant i descrivint els seus blocs principals.

### 5.2. FPGA

Una FPGA (Field Programmable Gate Array) és un dispositiu semiconductor que conté blocs lògics programables. Aquest blocs lògic poden realitzar des de operacions senzilles com OR o AND fins a operacions matemàtiques complexes.



**Figura 5.1:** Imatge d'una FPGA Virtex4

Algunes de les FPGAs, com la Virtex II Pro i la Virtex4 (Figura 5.1), poden combinar els blocs lògics amb microprocessadors.

La programació de les FPGAs definirà la funció a realitzar de cada bloc que la compon. Els llenguatges de programació utilitzats reben el nom de HDL (Hardware Description Language) i els més utilitzats són: VHDL, Verilog i ABEL.

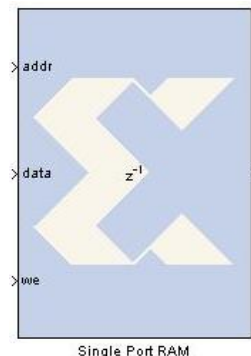
### 5.3. Introducció al System Generator

System Generator és una eina de Xilinx que utilitza l'entorn Simulink de Mathworks per al disseny d'algoritmes implementables en una FPGA. Aquesta eina ve proveïda amb els seus propis blocs per a la implementació dels algoritmes.

### 5.4. Block Set de Xilinx

El block set de Xilinx està compost per més de 90 blocs. Aquests blocs inclouen els blocs comuns com per exemple sumadors, multiplicadors i registres. També estan inclosos un conjunt de blocs complexos com blocs correctors d'error, FFTs, filtres i memòries. Existeixen blocs que fan la funció de passarel·la entre blocs de la llibreria Xilinx i la pròpia de Simulink i a l'inrevés.

#### 5.4.1 Memòria RAM d'un port (Single port RAM)



**Figura 5.2:** Memòria RAM d'un port

El bloc consta d'un port de sortida i tres d'entrada, dades d'entrada, adreça i el write enable (WE). Els valors que conté la memòria han de ser de característiques iguals: tipus, ample i posició del punt binari.

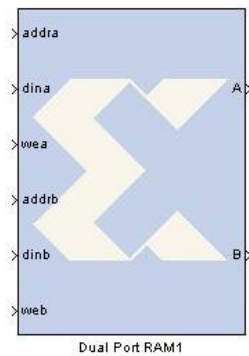
La profunditat de la memòria determina el nombre de paraules que pot emmagatzemar i cada paraula que conté la memòria està relacionada amb una adreça.

L'entrada del WE ha de ser booleana, quan el seu valor és 1, la dada d'entrada es escrita a la direcció corresponent amb la dada d'entrada del port d'adreça.

Al port de sortida obtenim la paraula escrita en l'adreça corresponent.

També existeixen ports opcionals com el Request que permet activar i desactivar la memòria. La seva entrada també ha de ser booleana, si és 0 es desactiva i si és 1 s'activa.

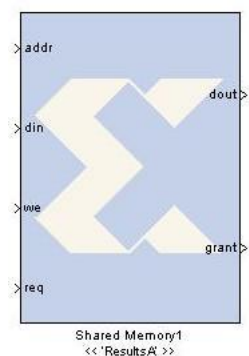
#### 5.4.2. Memòria RAM de dos ports (Dual port RAM)



**Figura 5.3:** Memòria RAM de dos ports

Les característiques són les mateixes que les memòries RAM d'un sol port, l'única diferència es que té dos ports de sortida i sis d'entrada. Això permet accedir i escriure a dues adreces diferents al mateix temps i obtenir dues sortides, una per cada port de sortida.

#### 5.4.3. Memòria compartida (Shared Memory)



**Figura 5.4:** Memòria compartida

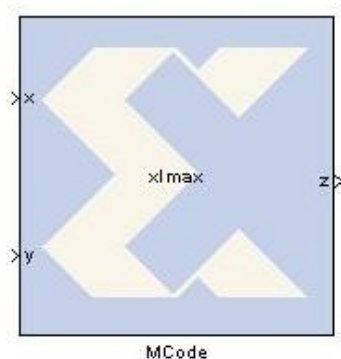
Aquest tipus de memòries es caracteritzen pel fet de poder ser identificades per un nom. Gràcies a aquest nom assignat és possible accedir a les dades de la

memòria des de Matlab o des de un altre model, és a dir, l'espai de la memòria pot ésser compartit per diverses simulacions i per al PC.

Els ports d'entrada i sortida són els mateixos que el d'una memòria d'un port amb la diferència que si els ports WE i Request estan a zero és possible accedir a les dades que emmagatzemen o assignar-li de noves des de Matlab.

## 5.5. Suport per a Matlab

A la llibreria de Xilinx hi és inclòs el bloc Mcode. Aquest bloc permet l'ús d'algoritmes senzills formats per operacions simples utilitzant un llenguatge de programació basat en Matlab. Al bloc existeix un paràmetre per assignar-li una funció escrita mitjançant un editor de Matlab.



**Figura 5.5:** Bloc Mcode

El bloc Mcode consta del mateix nombre de ports d'entrada i de sortida com de paràmetres d'entrada de la funció per al primer cas i de valors de retorn per al segon.

Les operacions que suporta aquest codi són les següents:

- Declaració de variables
- Condicionals If/else/elseif
- Expressions aritmètiques que només utilitzin sumes i restes.
- Sumes, restes i multiplicacions.
- Divisions per potència de dos.

Apart de les operacions també permet utilitzar operadors de relació:

- Menor que: <



- Menor o igual que: <=
- Major que: >
- Major o igual que: >=
- Igual que: ==
- Diferent que: ~=

Operadors de lògica:

- And: &
- Or: |
- Not: ~

## 5.6. Plataforma d'integració

System Generator proporciona una plataforma d'integració de sistemes per al disseny d'FPGAs que permet unir en una simulació i un sol entorn d'implementació els components de RTL, Simulink, Matlab i C/C++. System Generator incorpora un bloc 'black box' que permet que el RTL sigui importat a Simulink i co-simulat amb el simulador de Xilinx. System Generator també suporta la incorporació d'un processador per a executar programes en llenguatge C/C++.

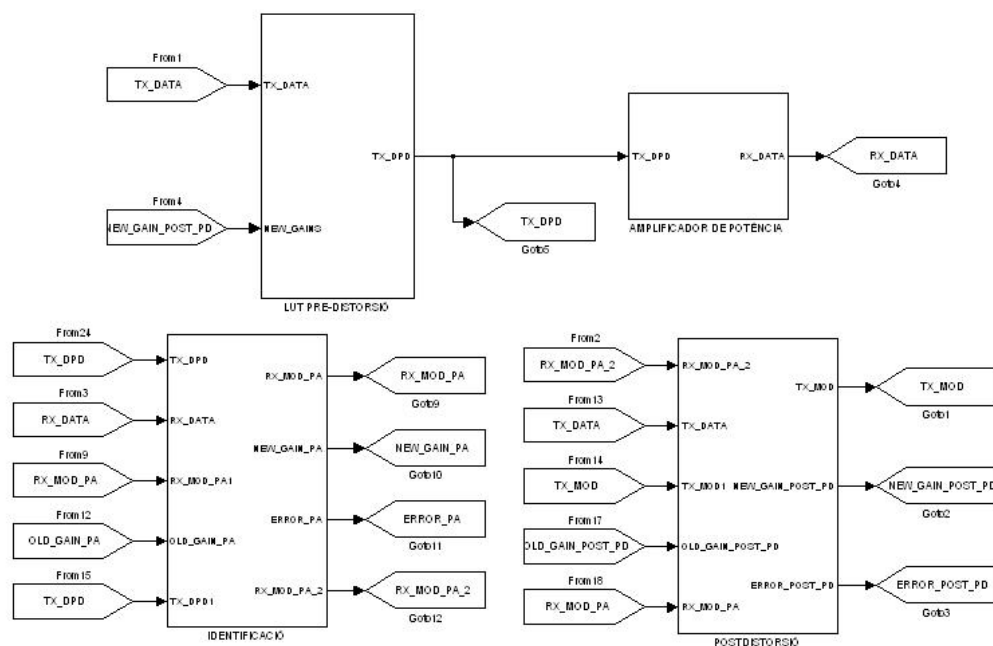
## CAPÍTOL 6: DESCRIPCIÓ DE L'ARQUITECTURA DEL PREDISTORSIONADOR DIGITAL

### 6.1. Introducció

En aquest apartat descriurem cada part del model de predistorsionador implementat mitjançant System Generator, s'explicarà el seu comportament i es comentaran els resultats obtinguts.

### 6.2. Arquitectura del Predistorsionador digital

El model de predistorsionador i AP dissenyat es mostra en la Figura 6.1, i està format principalment per quatre parts: la part de predistorsió, la d'amplificació, la d'identificació del model d'AP i la de postdistorsió o identificació del model de predistorsió.



**Figura 6.1:** Esquema del model d'amplificador i predistorsionador

Al iniciar la transmissió de dades la part de predistorsió no entra en funcionament. Això es degut a que el senyal TX\_DP0 no ha sofert els efectes de l'AP i per tant no s'ha identificat la seva relació AM-AM.

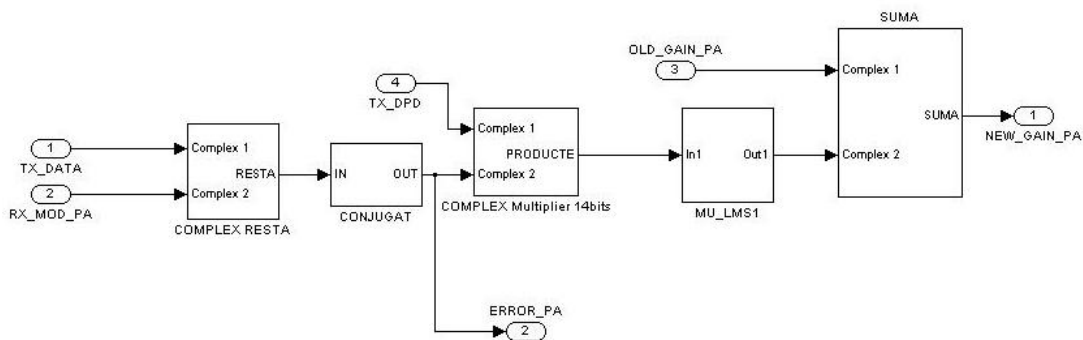
Una vegada el senyal s'ha vist afectat per l'AP el senyal rebut RX\_DATA és enviat a la part d'identificació del model d'AP, juntament amb el senyal d'entrada de l'amplificador TX\_DPD i el senyal transmès TX\_DATA.

Aquest bloc del sistema consta de tres subblocs: dues LUTs de identificació i un bloc LMS. La funció de la primera LUT d'identificació és la d'estimar la relació AM-AM i AM-PM entre el senyal d'entrada a l'amplificador TX\_DPD i el de sortida RX\_DATA a fi d'obtenir els guanys complexes que defineixen la funció no-lineal de l'AP. La segona LUT aprofita els guanys estimats en la primera LUT d'identificació i obté la sortida del model d'AP RX\_MOD\_PA2 a partir de la entrada RX\_DATA. El procés d'identificació dels guanys complexes que defineixen el comportament no lineal del AP es du a terme mitjançant l'algoritme LMS.

La funció del bloc LMS Figura 6.2 és la d'estimar els guanys complexes que es guardaran en les LUTs i que posteriorment permeten obtenir les sortides estimades RX\_MOD\_PA i RX\_MOD\_PA\_2, tot prenent com a referència la sortida del AP RX\_DATA. Els paràmetres d'entrada del bloc LMS són TX\_DATA, el paràmetre de pas  $\mu$ , RX\_MOD\_PA i OLD\_GAIN\_PA.

El senyal ERROR\_PA s'obté de la diferència entre la sortida real RX\_MOD\_PA i la estimada RX\_DATA i el valor de  $\mu$  és 0,1. El senyal OLD\_GAIN\_PA es suma als tres paràmetres anteriors. Segons les variables del nostre sistema, l'equació d'adaptació de l'algoritme LMS seria la següent:

$$NEW\_GAIN\_PA = (ERROR\_PA)^* \cdot TX\_DPD \cdot \mu + OLAD\_GAIN\_PA \quad (6.1)$$



**Figura 6.2:** Esquema LMS implementat en la part d'identificació

Els guanys NEW\_GAINS\_PA, obtinguts a la sortida del bloc LMS, són introduïts a les dues LUTs d'identificació, d'aquesta manera obtenim els senyals RX\_MOD\_PA i RX\_MOD\_PA\_2.

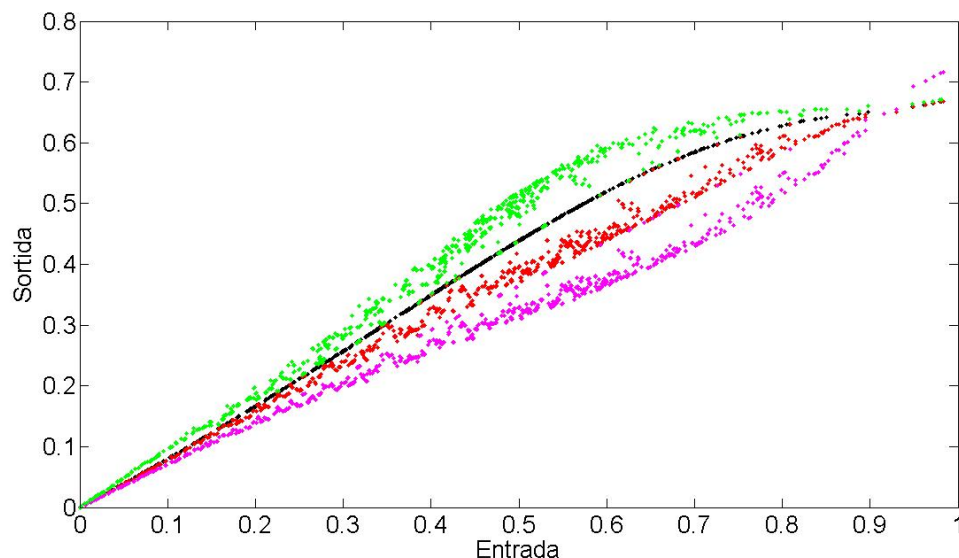
Un cop obtingut el senyal de sortida de l'identificador RX\_MOD\_PA\_2 s'envia a la part de postdistorsió. La funció d'aquesta part es estimar la corba AM-AM i

AM-PM inversa a la de l'AP. Aquesta estimació és fa possible al relacionar el senyal RX\_MOD\_PA\_2 amb el senyal TX\_DATA utilitzant també un altre bloc LMS de la mateixa manera que a la part d'identificació. En aquest cas el bloc LMS estima els guanys complexos que permeten obtenir un senyal de sortida del postdistorsionador TX\_MOD igual al del senyal d'entrada TX\_DATA.

Al mateix temps, els guanys estimats NEW\_GAINS\_POST\_PD es copien en la LUT de predistorsió, d'aquesta manera el senyal d'entrada TX\_DATA es veurà afectat per aquests guanys i obtindrem TX\_DPD, és a dir, el senyal predistorsionat que pretén contrarestar els efectes no-lineals introduïts pel AP.

### 6.3. Comportament del Predistorsionador

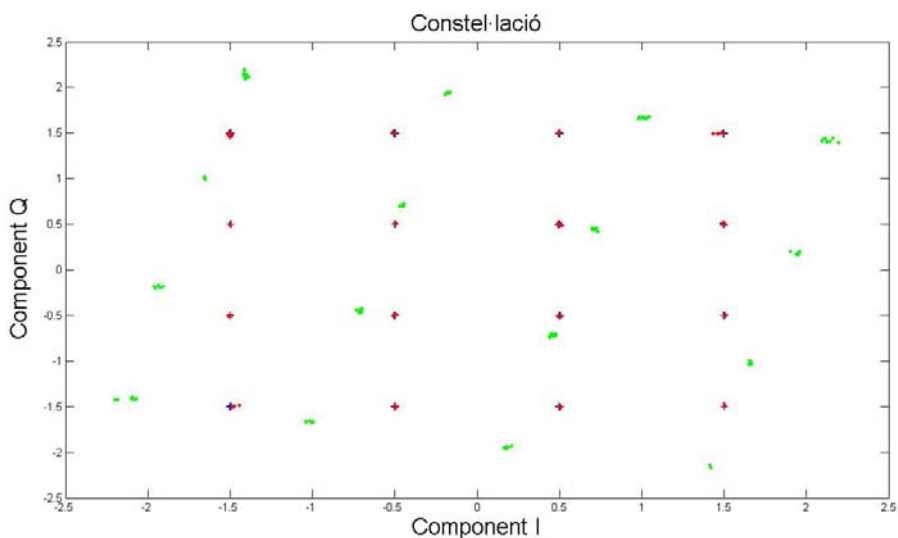
A la Figura 6.3 veiem representades les corbes de relació AM-AM de les diferents parts del sistema. El color negre representa la de l'AP, és a dir, el senyal d'entrada és TX\_DPD i el de sortida RX\_DATA, el color verd la de la part d'identificació del model de l'AP, senyal d'entrada TX\_DATA i senyal de sortida RX\_MOD\_PA2, el color rosa és la de predistorsió, senyal d'entrada TX\_DATA i senyal de sortida TX\_DPD, i la de color vermell representa la relació entrada sortida del sistema complet un cop s'ha completat la linealització, senyal d'entrada TX\_DATA i senyal de sortida RX\_DATA. Com podem observar la relació AMAM que s'aconsegueix a la part de predistorsió dibuixa una corba inversa a la de l'AP. Gràcies a aquest comportament obtenim una relació AMAM lineal del sistema complet



**Figura 6.3:** Relació AMAM de cada una de les parts del sistema

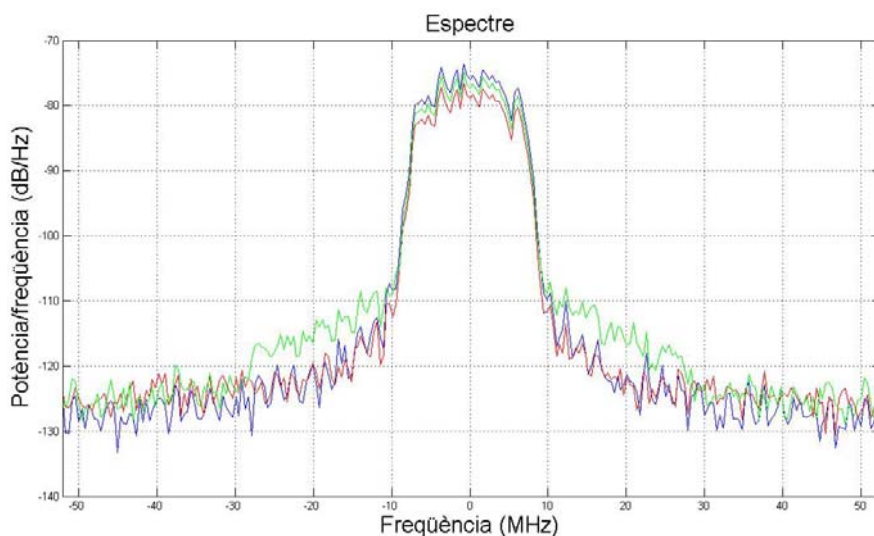
La Figura 6.4 ens representa la constel·lació dels senyals transmès, rebut i el distorsionat de color blau, vermell i verd respectivament.

Veiem com els símbols del senyal de sortida de l'amplificador no lineal estan lleugerament rotats, és a dir, pateixen una distorsió dintre de banda. La posició dels símbols del senyal rebut es correspon amb la dels del senyal transmès, això és degut a l'efecte que produeix el predistorsionador.



**Figura 6.4:** Constel·lació dels senyals d'entrada, de sortida i distorsionat

La Figura 6.5 representa els espectres dels senyals, transmès, rebut i amplificat amb distorsió de color blau, vermell i verd respectivament. Podem veure com el senyal de sortida del sistema és completament lineal, en canvi té menys potència que el d'entrada i el distorsionat. El senyal amplificat sense predistorsió produeix una emissió fora de banda que cal eliminar o mantenir sota uns nivells específics de ACPR.



**Figura 6.5:** Espectres dels senyals transmès, rebut i amplificat amb distorsió.

## CAPÍTOL 7: APLICACIÓ DE LA LòGICA DIFUSA A L'ALGORITME LMS

### 7.1. introducció

Al utilitzar l'algoritme LMS se'ns presenta un compromís entre el temps de convergència i la precisió de l'error quadràtic mig. El paràmetre de pas  $\mu$  és el que determina aquest compromís, si  $\mu$  té un valor alt la convergència de l'error és ràpida, en canvi, la seva precisió un cop assolit el nivell d'amplitud constant és baixa. Per una altra banda, si  $\mu$  pren un valor baix, el temps de convergència és elevat però l'amplitud de l'error en estat constant és menor i més estable que en el cas contrari.

En el nostre cas les fluctuacions de l'error no són constants. Durant les primeres mostres l'error pren valors d'amplitud alts i va decreixent de forma no constant. En el cas d'haver de triar un paràmetre de pas fix s'hauria de discernir entre si es prefereix una convergència ràpida o un valor d'error més acurat.

Amb la fi de no haver de patir aquest compromís s'ha implementat un algoritme basat en lògica difusa que permet fer variable aquest paràmetre de pas. Aquest algoritme s'ha incorporat a l'LMS implementat en un model d'amplificador del tipus Hammerstein amb polinomis de grau 7 i sense tenir en compte els efectes de memòria. La senyal d'entrada està composta per 24080 mostres, està modulada amb una 16-QAM filtrada amb un filtre conformador arrel de cosinus alçat amb factor de roll-off 0,35.

Són necessàries dues variables per a l'aplicació de la lògica difusa en l'algoritme LMS. En aquest cas l'error quadràtic mig es guardarà en una d'aquestes variables (Cost), l'altra variable de la que es farà ús serà la diferència entre l'error quadràtic mig instantani (Cost(n)) i l'error quadràtic mig de la passada iteració (Cost(n-1)), aquesta variable rebrà el nom de Dcost.

En aquest cas el paràmetre  $\mu$  es calcularà per a cada mostra del senyal, d'aquesta manera obtindrem el seu valor més eficient el major nombre de vegades possible.

A continuació s'implementarà l'algoritme LMS-Fuzzy seguint els següents passos: establiment de les regles lingüístiques, selecció de les funcions de pertinença i generant la taula de decisió.

### 7.2. Establiment de les regles lingüístiques

Les regles lingüístiques s'estableixen a partir del comportament de l'error del senyal en casos pràctics. S'ha comprovat que per a senyals amb un error alt i poc constant l'adaptació ha de ser ràpida, per tant un paràmetre de pas alt seria idoni. En el cas contrari, si l'error ja és estable, el valor òptim del

paràmetre  $\mu$  seria baix. Per al primer cas les variables Cost i Dcost prendrien valors alts i en el segon cas valors baixos.

La primera regla establerta és que en el cas de ser la variable Dcost negativa  $\mu$  no variaria, ja que això implica que la mostra anterior té un nivell d'error inferior al de la mostra actual. En el cas contrari utilitzem les següents regles:

R1 → Si el valor de Cost és petit i el de Dcost és petit,  $\mu$  ha de ser petita.

R2 → Si el valor de Cost és gran i el de Dcost és petit,  $\mu$  ha de ser mitja.

R3 → Si el valor de Cost és petit i el de Dcost és gran,  $\mu$  ha de ser mitja.

R4 → Si el valor de Cost és gran i el de Dcost és gran,  $\mu$  ha de ser gran.

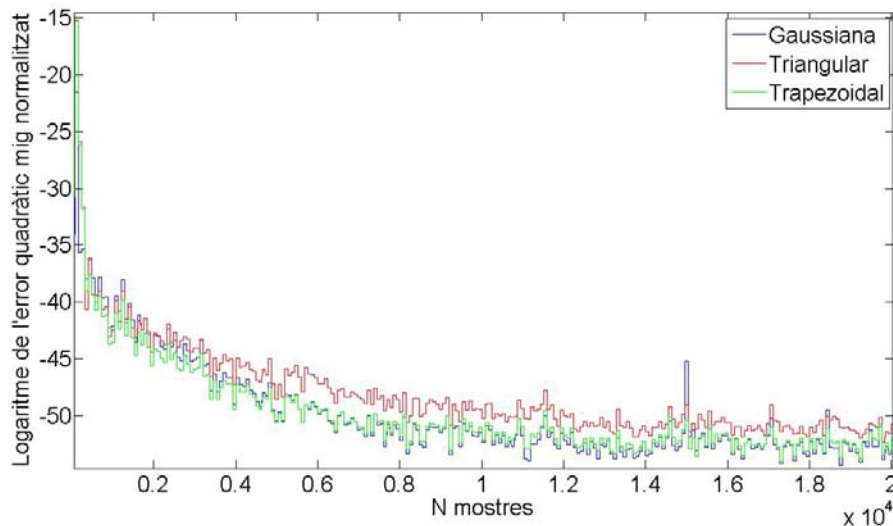
Amb aquestes quatre regles obtenim tres valors de  $\mu$ . Per al valor mínim de  $\mu$  s'ha triat 0.1, per al valor mig 0.8 i per al valor màxim 1.6.

No és necessari assignar només 3 valors de  $\mu$  a partir de l'establiment de les regles. S'han fet proves de simulació de resultats utilitzant un nombre més gran de valors mitjos però no s'ha apreciat cap millora significativa del comportament de l'error.

### 7.3. Selecció de les funcions de pertinença

Després de la selecció de les regles s'han de triar les funcions de pertinença. Aquestes funcions determinen el grau de pertinença entre una variable i un dels valors definits a les regles. Les variables Cost i Dcost són normalitzades amb la fi d'obtenir una funció senzilla. Per a aquesta tria s'han realitzat simulacions amb algunes de les funcions definides e l'apartat 4.2.

S'ha fet la mitja del logaritme del senyal d'error quadràtic mig normalitzat cada cinquanta mostres amb l'objecte d'afavorir la visualització del comportament d'aquest.





**Figura 7.1:** Estudi de les diferents funcions de pertinença aplicades a l'algoritme Fuzzy-LMS.

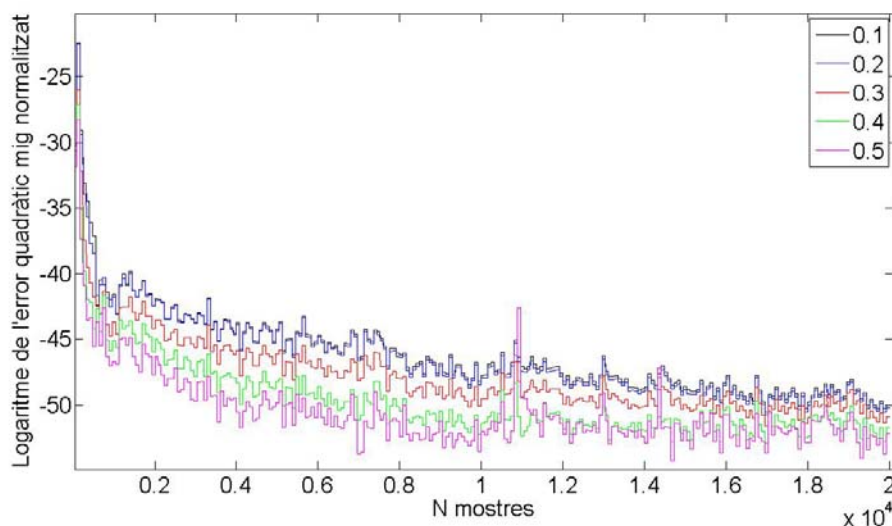
A la Figura 7.1 podem veure l'evolució del logaritme de l'error quadràtic mig per a cada una de les tres funcions de pertinença més utilitzades: la gaussiana amb una variància de 0.5, la triangular i la trapezoïdal.

Veiem que utilitzant la funció triangular l'error quadràtic necessita més temps per a convergir i el seu nivell d'amplitud constant és superior al de les altres dues funcions.

Amb la funció trapezoïdal l'error quadràtic convergeix més ràpid que en els altres dos casos, en canvi la seva amplitud en l'estat constant és més alta que en el cas d'utilitzar la funció gaussiana.

Utilitzant la funció gaussiana l'error quadràtic acaba assolint un nivell d'amplitud més baix que en els altres casos, en canvi, en algun punt aquest nivell es dispara.

A continuació es farà un estudi de la funció gaussiana modificant la seva variància:



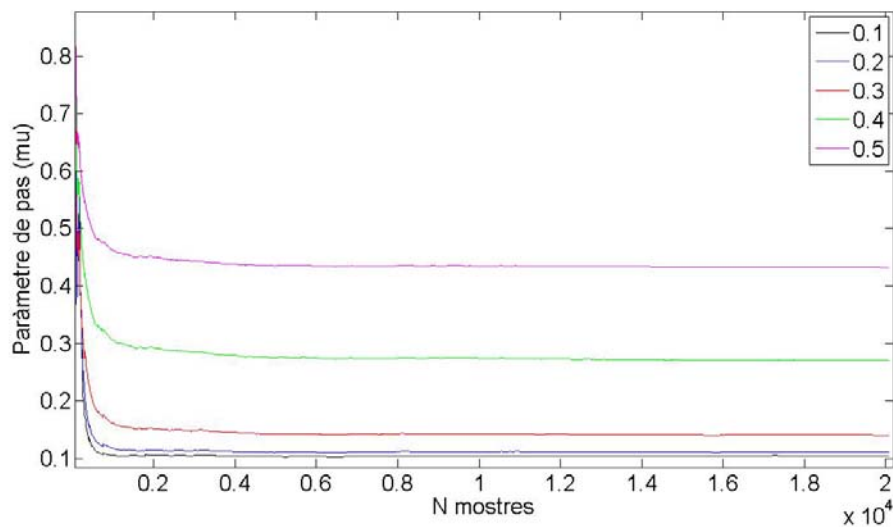
**Figura 7.2:** Estudi del comportament de l'error quadràtic en funció del valor de la variància de la funció Gaussiana

A la Figura 7.2 podem veure el comportament de la funció de pertinença gaussiana en cinc casos diferents, un per a cada variància. S'han fet proves per a valors de entre 0,1 i 0,5 ja que al sobrepassar el límit de 0,5 l'error divergeix i el sistema es tornaria inestable.

A la Figura 7.3 veiem l'evolució del paràmetre de pas  $\mu$  per a cada valor de la variància.

Durant les primeres mostres el valor de  $\mu$  és elevat però disminueix ràpidament i s'acaba estabilitzant. Veiem que el valor de  $\mu$  està relacionat amb el de la variància, si un creix l'altre també. Això implica que per a valors petits de variància l'amplitud de l'error és més constant que per a valors elevats, en canvi la seva convergència és més lenta i el seu valor en estat d'amplitud constant és més gran.

La variància òptima per a la funció gaussiana seria de entre 0,4 i 0,5, en el nostre cas s'ha triat la de 0,5.



**Figura 7.3:** Comportament del paràmetre de pas  $\mu$  en funció del valor de variància triat

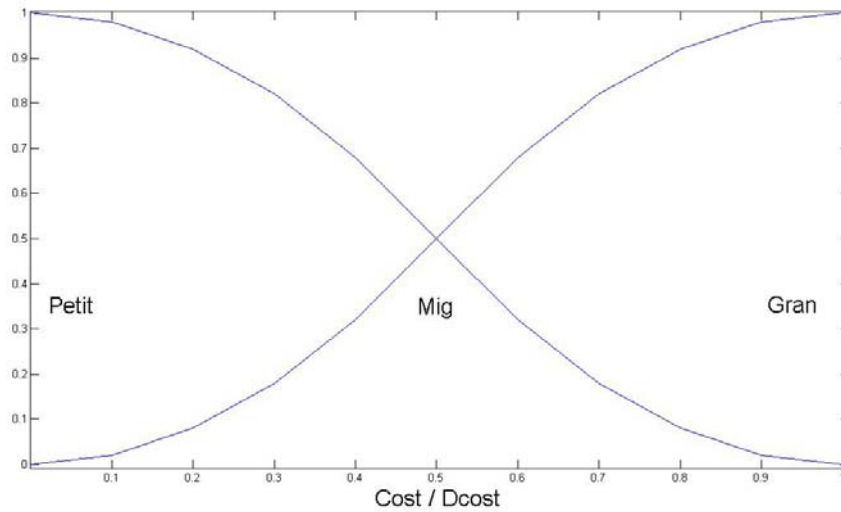
#### 7.4. Taula de decisió

Fen ús de les regles formulades a l'apartat 7.2 es crea una taula per a després obtenir el valor del paràmetre  $\mu$  a partir del mètode de centre de gravetat. Prenent  $\mu_s$ ,  $\mu_m$  i  $\mu_l$  com els valors petit, mig i gran de  $\mu$  respectivament i sent  $R_1$ ,  $R_2$ ,  $R_3$  i  $R_4$  les regles establertes a l'apartat 7.1.

| Cost/Dcost | Petit   | Gran    |
|------------|---------|---------|
| Petit      | $\mu_s$ | $\mu_m$ |
| Gran       | $\mu_m$ | $\mu_g$ |

**Taula 7.1:** Taula de decisió

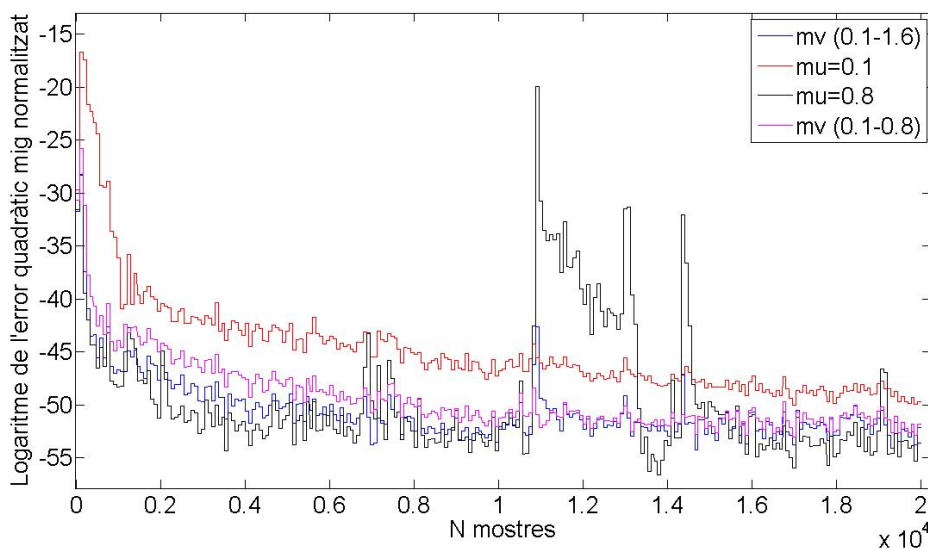
$$\mu = (R_1\mu_s + R_2\mu_m + R_3\mu_m + R_4\mu_l) / (R_1 + R_2 + R_3 + R_4)$$



**Figura 7.4:** Funció de pertinença de les variables Cost i Dcost

### 7.5. Resultats

A continuació es farà un estudi comparatiu entre els resultats obtinguts variant els marges del paràmetre  $\mu$  i els aconseguits amb valors fixos.



**Figura 7.5:** Comparació entre els algoritmes LMS i Fuzzy-LMS

A la Figura 7.5 veiem representats els resultats d'error quadràtic mig havent utilitzat l'algoritme LMS per a diferents valors de  $\mu$  fixa i l'algoritme LMS-Fuzzy amb dos intervals de  $\mu$  variable fent servir una funció de pertinença gaussiana amb variància de 0,5.

De color vermell veiem representat l'error amb una  $\mu$  mínima. Podem observar com l'error convergeix lentament i no acaba d'assolir un valor tan baix com a la resta de casos, si el senyal estigués format per una quantitat de mostres més gran acabaria arribant al nivell d'amplitud dels altres casos.

El color negre representa el mateix error però amb un valor de  $\mu$  fixa de 0.8. Amb aquest valor aconseguim una convergència més ràpida que en el cas anterior. En canvi veiem que arriba un punt en el que l'error es dispara considerablement arribant a un valor de -19,97dB.

S'ha utilitzat també un valor de  $\mu$  de 1,6 però l'error divergia i per tant no s'ha cregut necessària la representació.

Per últim tenim els resultats després d'utilitzar l'algoritme LMS-Fuzzy. De color rosa amb un marge de  $\mu$  de 0,1 fins a 0,8 i de color blau amb marge de 0,1 fins a 1,6. Veiem que per al primer cas la convergència de l'error és menor i el seu temps més gran, en canvi el seu comportament és més estable que en el cas d'un marge més ampli. En el segon cas podem observar algun pic de valor -42.5dB.

## **CAPÍTOL 8: INCORPORACIÓ DE L'ALGORITME FUZZY-LMS AL SISTEMA PREDISTORSIONADOR**

### **8.1. Introducció**

Un cop realitzades totes les proves necessàries per a decidir-nos a utilitzar l'algoritme Fuzzy-LMS, tractem d'incorporar-lo al sistema predistorsionador (PD).

En aquest apartat explicarem els canvis principals del model de predistorsionador al introduir el sistema Fuzzy-LMS i analitzarem els resultats obtinguts utilitzant aquest i els compararem amb els de l'algoritme LMS amb diferents valors per al paràmetre de pas  $\mu$ .

### **8.2. Mètode d'incorporació de l'algoritme Fuzzy-LMS al sistema predistorsionador**

Per a la incorporació de l'algoritme Fuzzy-LMS es van pensar diferents mètodes d'implementació.

En un principi el mètode més senzill semblava el de la utilització del bloc M-code de la llibreria Xilinx. Aquest permet la introducció de codi fent servir un llenguatge semblant a l'utilitzat a Matlab però amb diverses restriccions regides per la FPGA. La seva implementació no va ser possible ja que era necessari introduir les funcions de pertinença i una de les restriccions del bloc és que només permet fer divisions per potències de dos.

Un altre mètode considerat era el d'utilitzar un bloc divisor. La idea va ser refusada perquè aquest bloc introdueix un retard de mostra massa elevat, i per tant el sistema no s'ho pot permetre de cara a facilitar adaptació en temps real.

Al descartar els dos mètodes descrits es va haver de recorre a la implementació de l'algoritme per mitjà de Matlab. Es van crear dues noves memòries compartides, una per l'LMS de la part d'identificació i l'altre per al de la part de postdistorsió. La funció d'aquestes memòries és la d'emmagatzemar els valor  $\mu$  calculats prèviament per a després utilitzar-los al activar l'LMS.

El problema d'aquest tercer mètode és que, al ser un algoritme que calcula el paràmetre de pas a partir de l'error del sistema, i aquest només és accessible a partir d'una altra memòria compartida, cal esperar que tot el senyal sigui processat per poder obtenir els valors d'error. Per a solucionar aquest problema s'ha dividit el senyal d'entrada en diversos fragments del mateix nombre de mostres per així processar-los de forma independent. Un cop processat cada fragment s'aplica l'algoritme basat en lògica difusa i es carrega la memòria amb tants valors de  $\mu$  com mostres té el fragment. Després es recupera el senyal sencer agrupant novament els fragments.

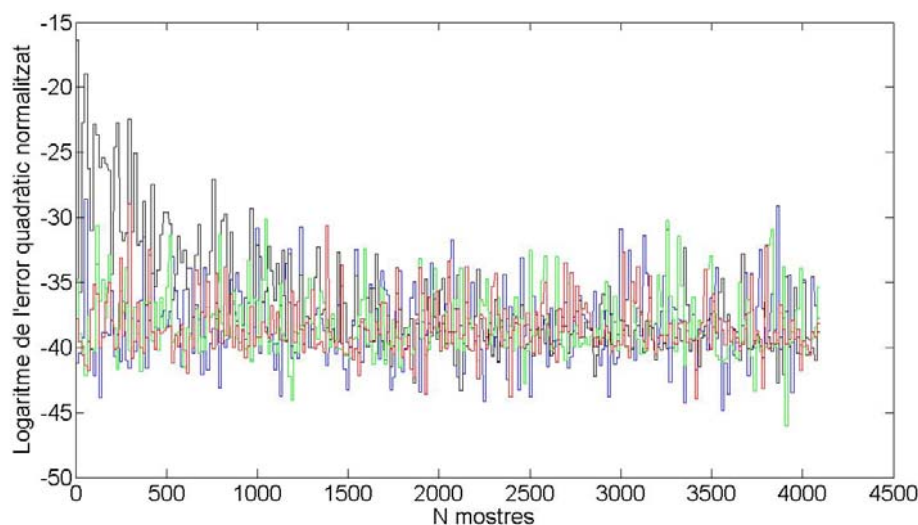
Les conclusions extretes de l'apartat 7.4 era que el marge de  $\mu$  òptim per a la implementació de l'algoritme Fuzzy era de 0.1 a 1.6. No ha estat possible la utilització d'aquest marge a l'hora de la incorporació de l'algoritme al sistema ja que aquest treballa amb paraules de bits del tipus [14 13]. Això significa que la seva llargada es de 14 bits i la posició del punt decimal és la 13, per tant el número màxim representat és 0.9.

S'ha fet la prova amb un senyal de 4096 mostres fragmentant-la en parts de seixanta-quatre amb  $\mu$  variable de marge 0.1-0.9 amb una funció de pertinença gaussiana de variància 0.5. Els resultats s'han comparat amb errors obtinguts a partir d'utilitzar  $\mu$  fixes de 0.1, 0.5 i 0.9.

### 8.3. Comportament de l'error quadràtic mig de l'LMS

La Figura 8.1 ens mostra el logaritme de l'evolució de l'error quadràtic mig normalitzat per a cada un dels valors fixos de  $\mu$  anomenats a l'apartat 6.5 i per al cas de  $\mu$  variable.

S'ha fet la mitja d'aquest senyal cada setze mostres amb l'objecte d'afavorir la visualització del comportament d'aquest.



**Figura 8.1:** Evolució del logaritme de l'error quadràtic mig normalitzat en tres casos de  $\mu$  fixa i en el de  $\mu$  variable

El color blau representa el logaritme del comportament de l'error quadràtic mig per al valor 0.1 de  $\mu$ , el color vermell per al valor de 0.5, el color negre per al valor de 0.9 i el color verd és el resultat d'utilitzar l'algoritme Fuzzy-LMS.

Com podem veure en els quatre casos l'error quadràtic mig té un comportament molt similar, amb la diferència de que en el cas de  $\mu$  de valor 0.1 obtenim una amplitud d'error més baixa.

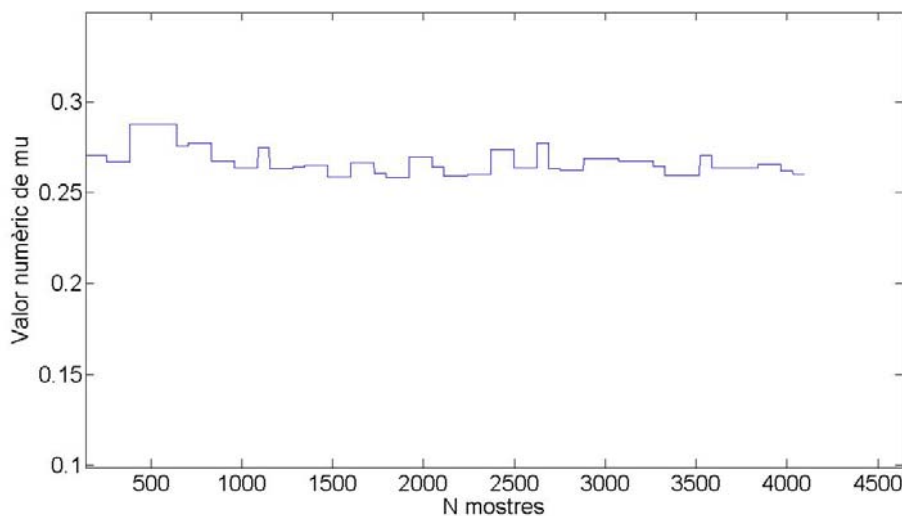
La Taula 8.1 ens mostra el valor mig i la variància dels senyals d'error per a cada un dels valors de  $\mu$  triat.

| Valor de $\mu$ | Mitja [dB] | Variància |
|----------------|------------|-----------|
| 0,1            | -38.1812   | 32.4913   |
| 0,5            | -38.4696   | 23.0827   |
| 0,9            | -36.6596   | 43.7345   |
| Fuzzy-LMS      | -38.2606   | 27.1155   |

**Taula 8.1:** Valor mig i variància per a cada un dels valors de  $\mu$ .

Basant-nos en els valors de mitja i de variància del logaritme de l'error quadràtic mig que ens mostra la Taula 8.1 podem extreure diverses conclusions. Amb una  $\mu$  de valor de 0,9 obtenim el valor més baix de mitja, en canvi el valor de variància és molt més alt que per a la resta de casos. En aquest cas es compleix la teoria explicada a l'apartat 7.1, encara que cal esmentar que els valors de mitja dels errors per a cada un dels casos varien molt lleugerament. La mínima variància la obtenim en el cas del valor de  $\mu$  igual a 0,5. Aquesta dada indica que aquest valor és el que proporciona un error més estable i per tant una millor adaptació. El fet d'utilitzar l'algoritme Fuzzy-LMS no proporciona resultats significatius. Obtenim una variància bastant baixa (el segon millor cas) i un valor de mitja molt similar al de la resta de casos.

La Figura 8.2 ens mostra l'evolució del paràmetre  $\mu$  en el cas d'haver utilitzat l'algoritme Fuzzy-LMS.



**Figura 8.2:** Evolució del paràmetre  $\mu$  variable

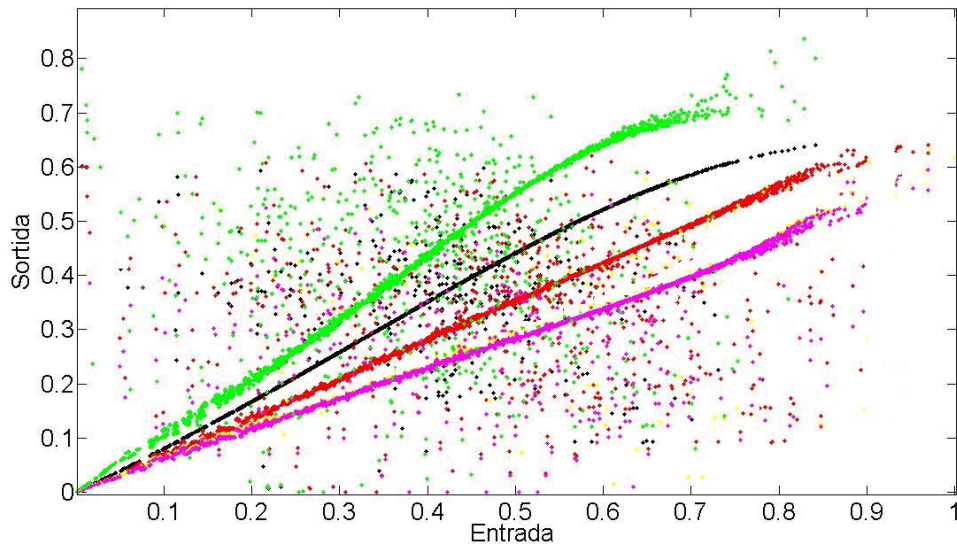
Per a la representació de l'evolució del paràmetre de pas  $\mu$  s'han suprimit les 128 mostres. Aquesta operació s'ha portat a terme perquè per a les dues primeres iteracions el valor inicial de  $\mu$  és forçat a 0.1, ja que no és possible utilitzar l'algoritme Fuzzy-LMS al dependre aquest de l'error quadràtic mig i de la diferència entre aquest i el de la iteració anterior.

Com podem veure, l'evolució del paràmetre és bastant constant, prenent un valor mig de 0.27 amb una variància de 0.02. En aquest cas veiem que el fet de que l'error del sistema no fluctuï gaire i no interressi utilitzar un paràmetre  $\mu$  variable es pot utilitzar l'algoritme Fuzzy per a obtenir un valor de  $\mu$  fix òptim per a l'LMS.

#### 8.4. Inconvenients a l'utilitzar l'algoritme Fuzzy-LMS

S'ha pogut veure a l'apartat anterior com l'algoritme basat en lògica difusa pot arribar a millorar el sistema LMS en l'aspecte de l'adaptació minimitzant l'error quadràtic mig. En canvi, al fer la comprovació del comportament del sistema predistorsionador s'observen diverses anomalies a tenir en compte.

La Figura 8.3 mostra les corbes de relació AMAM de les diferents parts en les que es divideix el sistema predistorsionador i amplificador.



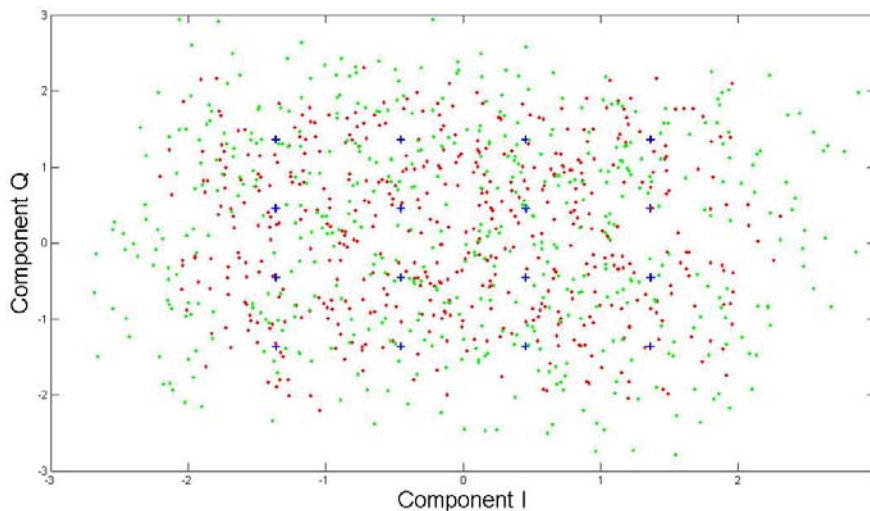
**Figura 8.3:** Relació AMAM de cada una de les parts del sistema predistorsionador utilitzant l'algoritme Fuzzy-LMS

El codi de colors utilitzat es correspon a l'utilitzat en la Figura 6.3. Com podem veure, es poden distingir les corbes de relació AMAM de cada part del sistema predistorsionador. La diferència notable amb la Figura 6.3 és l'aparició d'un



núvol de punts per a cada una de les representacions. Aquests punts provocarien una no linealitat important.

A la Figura 8.4 podem apreciar la manera en la que afecta aquest núvol a la constel·lació dels senyals.



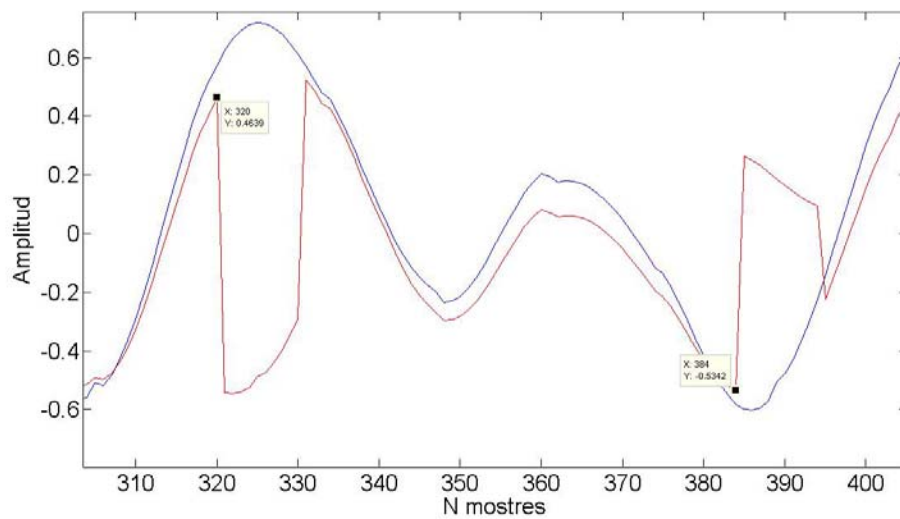
**Figura 8.4:** Constel·lació dels senyals d'entrada, de sortida i distorsionat utilitzant l'algoritme Fuzzy-LMS

Com s'ha dit a l'apartat 8.2, el mètode triat divideix el senyal d'entrada en fragments iguals de seixanta-quatre mostres per a processar-los de manera independent.

El sistema provoca un retard entre el senyal rebut i el transmès, per tant, cada fragment pateix un retard. Això fa que al tornar a agrupar els fragments per a la reconstrucció del senyal sencer es formin irregularitats entre fragment i fragment, és a dir, cada seixanta-quatre mostres.

La Figura 8.5 ens mostra la comparació des de la mostra 304 fins la mostra 405 entre la part real del senyal transmès, dibuixat de color blau i la part real del senyal rebut, dibuixat de color vermell.

Es poden identificar dues irregularitats en aquesta representació, una es produeix a la mostra 320, que es correspon amb el començament del cinquè fragment, i el senyal s'estabilitza a la mostra 331 mentre que l'altra comença a la mostra 384, inici del sisè fragment, i el senyal s'estabilitza a la mostra 395.



**Figura 8.5:** Comparació entre el senyal transmès i el senyal rebut

## CONCLUSIONS

En aquest TFC s'ha fet un estudi de l'algoritme Fuzzy-LMS per tal d'aconseguir una millora en l'adaptació dels guanys de les LUTs utilitzades, en diversos blocs, per a la implementació d'un Predistorsionador digital.

Abans de la incorporació de l'algoritme Fuzzy-LMS al Predistorsionador digital es va implementar en un model Hammerstein. D'aquesta manera es fa possible l'avaluació del comportament de l'error quadràtic mig en funció de la variació del paràmetre de pas  $\mu$  o l'elecció de les funcions de pertinença. Així que es va programar un algoritme que feia variable el paràmetre  $\mu$  per a cada mostra i es van fer els estudis corresponents.

Un cop triada la funció de pertinença adequada, establertes les regles If-Then, obtinguts els marges de  $\mu$  idonis i demostrada la millora en l'evolució de l'error quadràtic mig de l'algoritme Fuzzy-LMS respecte a l'algoritme LMS es tracta d'incorporar l'algoritme Fuzzy-LMS al Predistorsionador digital.

Aquesta incorporació es realitzà exitosament però amb algunes limitacions regides per la FPGA. Aquestes limitacions no van permetre una actualització del paràmetre  $\mu$  per a cada mostra i van obligar a fragmentar el senyal en parts iguals per a l'aplicació de l'algoritme.

La fragmentació del senyal va provocar una incorrecta recepció del senyal a la sortida de l'amplificador a causa d'una incorrecta actuació del sistema Predistorsionador. Aquest inconvenient és de major magnitud que les millores que l'algoritme Fuzzy-LMS pot garantir

Per aquest motiu es proposa utilitzar l'algoritme Fuzzy-LMS només per a poder identificar el valor del paràmetre de pas òptim i així utilitzar-lo com a valor fix.

En el cas en que el valor de  $\mu$  hagués de variar per la seguretat de l'adaptació del sistema es podria utilitzar l'algoritme utilitzant una variabilitat del paràmetre menys freqüent, és a dir, dividint el senyal en fragments d'un elevat número de mostres, així el retard no influiria amb tanta magnitud en el senyal de sortida del sistema.

## BIBLIOGRAFIA

- Tesis:

[1] Gilabert Pinal, P.L., *Multi Look up table digital predistortion for RF power amplifier linearization.*, Departament de Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya, Barcelona, (2007).

[2] Zozaya S. A. J., *Aportación a la linealización de amplificadores de potencia mediante la Teoría de Hiperestabilidad*, Departament de Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya, Barcelona.

- Manuals

[3] Fuzzy Logic toolbox Users Guide

[4] Simlunk Xilinx Help

- Articles

[5] Bertran Albertí, E.; Montoro López, G., *On the application of Fuzzy Logic to LMS algorithm*, ., Departament de Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya, Barcelona.

[6] Pham Van Su, Le Minh Tuan, Jewoo Kim, Giwan Yoon, *A new Fuzzy Logic Application to Variable Step Size LMS Algorithm for Adaptative Antenna in CDMA Systems*, Infomation & Communications University (ICU), Korea.

[7] Barkley, K., *Two tone IMD Measurement Techniques*, , PRIMEDIA Business Magazines & Media Inc, Estats Units, 2001.

- Pàgines web

[8] [www.physionet.cps.unizar.es/~eduardo/docencia/tds/librohtml/lms1.htm#Ej\\_LMS](http://www.physionet.cps.unizar.es/~eduardo/docencia/tds/librohtml/lms1.htm#Ej_LMS)

[9] [www.physionet.cps.unizar.es/~eduardo/docencia/tds/librohtml/adapt1.htm](http://www.physionet.cps.unizar.es/~eduardo/docencia/tds/librohtml/adapt1.htm)

[10] [www.tsc.uc3m.es/~mlazaro/FiltAdapt/LMS.pdf](http://www.tsc.uc3m.es/~mlazaro/FiltAdapt/LMS.pdf)

[11] [www.tsc.uc3m.es/~mlazaro/FiltAdapt/MaximaPendiente.pdf](http://www.tsc.uc3m.es/~mlazaro/FiltAdapt/MaximaPendiente.pdf)

## ANNEXOS

### A.1. Codi principal

```

L=2^6; L2=2^10;L3=2^12;
L_signal=L;
Ts=50e-9;
DELAY=0; Guany_Corr=4;
Hs=spectrum.welch;Hs.SegmentLength=2^8;
q=quantizer([14 13]); q1=quantizer([14 10]);
ms=0.1;
mm1=0.5;
mm2=0.5;
m1=0.9;
mu1(1:L)=m1;
mu2(1:L)=m1;
Mcost1(1:L)=0;
Mcost2(1:L)=0;
Mdcost1(1:L)=0;
Mdcost2(1:L)=0;
Mconstant1(1:L)=0;
Mconstant2(1:L)=0;
Vmu1(1:L3)=0;
Vmu2(1:L3)=0;
Verror1(1:L3)=0;
Verror2(1:L3)=0;
VTx_I(1:L3)=0;
VTx_Q(1:L3)=0;
VRx_I(1:L3)=0;
VRx_Q(1:L3)=0;
VRes_I2(1:L3)=0;
VRes_Q2(1:L3)=0;
VRes_I3(1:L3)=0;
VRes_Q3(1:L3)=0;
VRes_I4(1:L3)=0;
VRes_Q4(1:L3)=0;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

#### Inicialtzació de variables

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

Tx=SharedMemory('Tx_buffer2');
Rx=SharedMemory('Rx_buffer2');
Cont=SharedMemory('buffer_control2');
Gains_RX=SharedMemory('buffer_Gains_RX2');
Res=SharedMemory('ResultsA2');
Res2=SharedMemory('ResultsX2');

```

```

Res3=SharedMemory('ResultsY2');
Res4=SharedMemory('ResultsZ2');
Res5=SharedMemory('ResultsB2');
Delay=SharedMemory('Delay_correlator2');
Add=SharedMemory('Adreces2');
LUT_PA=SharedMemory('Amplifier2');
Gains_DPD=SharedMemory('Gains_DPD2');
Gains_TX=SharedMemory('Gain_TX2');
M1=SharedMemory('mul1');
M2=SharedMemory('mu2');

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

### Inicialització de memòries compartides (Shared Memories)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
GUANYS LUT PA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Guany_PA=.5;
[x1,x2]=MOD_16_QAM_RRCFilter_LONG2(4,236);
[Gain,y]=LUT_model_PA(x2,Guany_PA,L2);
LUT_PA(:)=bin2dec([num2bin(q,imag(Gain))num2bin(q,real(Gain))]);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

### Definim el comportament de l'AP i l'assignem els guanys.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MODULACIÓ ENTRADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ATT=1.65;
[x,x_filt]=MOD_16_QAM_RRCFilter_LONG2(ATT,300); %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

### Definim la modulació del senyal d'entrada al sistema

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

for m=1:L:L3
    for i=1:1:5
        Tx_I=real(x_filt(m:m+L-1));
        Tx_Q=imag(x_filt(m:m+L-1));

        Cont(0)=bin2dec([dec2bin(DELAY,10) '000000000']); %%
Request ha d'estar a 0 per escriure des de Matlab
        pause(Ts/10);
        Tx(:)=bin2dec([num2bin(q,Tx_Q) num2bin(q,Tx_I)]);
        pause(Ts/10);

        DELAY=4;
        Gain_Tx=0;
        Guany_Corr=1.35;%4.5*exp((pi/180)*133i);
        Gain_Rx_Mod=1.37;

```

```

Gain_Tx_Ref=1;
Gain_000=0.999999999999;

Gains_RX(0)=bin2dec([num2bin(q1,imag(Guany_Corr))
num2bin(q1,real(Guany_Corr))]);
Gains_DPD(0)=bin2dec([num2bin(q1,Gain_Tx_Ref)
num2bin(q1,Gain_Rx_Mod)]);
Gains_TX(0)=bin2dec([num2bin(q1,Gain_000)
num2bin(q1,Gain_Tx)]);

pause(Ts/10)
Cont(0)=bin2dec([dec2bin(DELAY,10) '0100001101']); %%
(sense UPDATE)
pause(Ts/10)
Cont(0)=bin2dec([dec2bin(DELAY,10) '0100011101']); %%
(Calcula LUT IDENTIFICACIÓ PA)
pause(Ts)
Cont(0)=bin2dec([dec2bin(DELAY,10) '0100111101']); %%
(Calcula LUT POST-DISTORSIO)
pause(Ts)
Cont(0)=bin2dec([dec2bin(DELAY,10) '0101111101']); %%
(Fa UPDATE de la LUT de DPD)
pause(Ts)
Cont(0)=bin2dec([dec2bin(DELAY,10) '0110101101']); %%
(SELECCIONA FER PREDISTORSIO)
Gain_Tx_Ref=1.05;
Gains_DPD(0)=bin2dec([num2bin(q1,Gain_Tx_Ref)
num2bin(q1,Gain_Rx_Mod)]);
pause(Ts)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Bucle en el qual activem les LUTs del circuit mitjançant els bits de sortida de la
memoria compartida Control. També fragmentem el senyal de 4096 mostres en
parts de 64 mostres

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Recuperem Tx & Rx %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

pause(Ts)
Cont(0)=bin2dec([dec2bin(DELAY,10) '0000000000']); %%
Request ha d'estar a 0 per escriure des de Matlab
pause(Ts)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%utilitzant fuzzy logic%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[Tx_I,Tx_Q,Rx_I,Rx_Q,error1,error2,Res2_I,Res2_Q,Res3_I,Res
3_Q,Res4_I,Res4_Q,mu1,mu2,Mcost1,Mcost2,Mdcost1,Mdcost2,Mco
stant1,Mconstant2,M1,M2]=Recuperar_Buffersfuzzy(L,q,Tx,Rx,Re

```





```
[VTxD_I,VTxD_Q,Guany_I,Guany_Q,Delay_I,Delay_Q]=Sincronitzar(VRx_I,VRx_Q,VTx_I,VTx_Q);
```

```
[VPost_PD_I,VPost_PD_Q,G_I_PD,G_Q_PD,D_I,D_Q]=Sincronitzar(VRx_I,VRx_Q,VRes2_I,VRes2_Q);
```

```
[VTx_DPD_I,VTx_DPD_Q,G_I,G_Q,De_I,De_Q]=Sincronitzar(VRx_I,VRx_Q,VRes3_I,VRes3_Q);
```

```
[VRx_MOD_I,VRx_MOD_Q,G_I_MOD,G_Q_MOD,D_I_M,D_Q_M]=Sincronitzar(VRx_I,VRx_Q,VRes4_I,VRes4_Q);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Utilitzem les funcions per a sincronitzar senyals

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOT RESULTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
plot(abs(complex(VTx_DPD_I,VTx_DPD_Q)),abs(complex(VRx_I,VRx_Q)),'.k')
```

```
    hold on;
```

```
plot(abs(complex(VTxD_I,VTxD_Q)),abs(complex(VRx_I,VRx_Q)),'.r')
```

```
plot(abs(complex(VRx_MOD_I,VRx_MOD_Q)),abs(complex(VPost_PD_I,VPost_PD_Q))/Gain_Rx_Mod,'.y')
```

```
plot(abs(complex(VTxD_I,VTxD_Q)),abs(complex(VTx_DPD_I,VTx_DPD_Q))/(Gain_Rx_Mod*Gain_000),'.m')
```

```
plot(abs(complex(VTx_DPD_I,VTx_DPD_Q)),abs(complex(VRx_MOD_I,VRx_MOD_Q))/Gain_Rx_Mod,'.g')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Representem gràficament de les corbes de relació AMAM de cada part del sistema

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    psd(Hs,complex(VTx_I,VTx_Q), 'CenterDC', true, 'fs', 105e6)
```

```
    hold on
```

```
    psd(Hs,complex(VRx_I,VRx_Q), 'CenterDC', true, 'fs', 105e6)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Representem gràficament l'espectre dels senyals

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
release(Tx);
release(Rx);
release(Cont);
release(Gains_RX);
release(Res);
release(Res2);
release(Res3);
release(Res4);
release(Res5);
release(Delay);
release(Add);
release(LUT_PA);
release(Gains_DPD);
release(Gains_TX);
release(M1);
release(M2);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Alliberem les memòries compartides dels guanys assignats

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## A.3. Funcions

### A.2.1. Funció per crear un senyal modulat en 16 QAM

```
function [x,x_filt]=MOD_16_QAM_RRCFilter_LONG2(ATT,Tf)

% ATT s'utilitza per controlar l'amplitud del senyal
% generat (per defecte
% poseu 1, i si després necessiteu canviar l'amplitud del
% senyal generat
% llavors el canvieu)

% Tf és la llargada del nombre de símbols sense filtrar (el
% nombre de símbols de x).

% "x" són els símbols 16-QAM sense filtrar
% "x_filt" són els símbols 16-QAM filtrats (els que heu
% d'utilitzar). El
% nombre de símbols serà un factor "nsamp" aprox. el nombre
% de símbols de "x"
```

```

%Generador de la modulació
bits_simbolo=4;
N_simbolos=2^(bits_simbolo);
mensaje_T=randint(Tf,1,N_simbolos);
s_T=qammod(mensaje_T,N_simbolos,0,'gray');

x=s_T/ATT;

%Filtratge Conformador Square-root raised cosine roll of
0.35
%nsamp=4;
nsamp=8;
filtorder =48; % Filter order
delay = filtorder/(nsamp*2); % Group delay (# of input
samples)
rolloff=0.35;

Filt=rcosine(1,nsamp,'fir/sqrt',rolloff,delay);
x_filt=rcosflt(x,1,nsamp,'filter',Filt);
end

```

## A.2.2. Funció per l'assignació de guanys a la LUT PA

```

function [Gain,y]=LUT_model_PA(x,Guany_PA,L2)

bits_LUT=log2(L2);
npol=9;
x_filt=x/max(abs(x));
[Out_memoria,Out_static]=PA_ClasseA_MemoryEffects(x_filt);
Out_static=(Out_static/max(abs(Out_static)))*Guany_PA;
plot(abs(x_filt),abs(Out_static)); hold on
%[u,d,d2,E]=ID_HAMMERSTEIN_amb_Taus_FIR_IIR_abs2(x_filt,Out
_static,7,0,0);
x_filt=x_filt*1;
[u,d,d2,E]=ID_HAMMERSTEIN_amb_Taus_FIR_IIR_abs(x_filt,Out_s
tatic,npol,0,0);
%[f1,Gain]=Generacio_LUT_abs2(x_filt,d,7,9);
[f1,Gain]=Generacio_LUT_abs(x_filt,d,npol,bits_LUT);
Addr=round(abs(x_filt)*2^bits_LUT);
for i=1:1:length(x_filt)
    if Addr(i)==0 Addr(i)=1; end
end
plot(abs(x_filt),abs(x_filt.*conj((Gain(Addr))).')), 'r');
y=x_filt.*conj((Gain(Addr))).';
end

```



```

Res3_I(i)=bin2num(q,Resultat3(15:1:28));
end

%%%%% Veure el valor de la LUT Resultats Z %%%%%%%%%%%

for i=1:1:L
Resultat4=dec2bin(Res4(i-1),28);
Res4_Q(i)=bin2num(q,Resultat4(1:1:14));
Res4_I(i)=bin2num(q,Resultat4(15:1:28));
end

%%%%% Veure el valor de la LUT Resultats B %%%%%%%%%%%

for i=1:1:L
Resultat5=dec2bin(Res5(i-1),28);
Res5_Q(i)=bin2num(q,Resultat5(1:1:14));
Res5_I(i)=bin2num(q,Resultat5(15:1:28));
end
error2=complex(Res5_I,Res5_Q);

%%%%%%%%%%%%%fuzzy logic%%%%%%%%%%%%%

for i=1:1:L
    Mcost1(i)=mean(abs((error1(1:L))./max(error1)).^2)
    if (Mconstant1(i)~=0)
        Mdcost1(i)=Mconstant1(i)-Mcost1(i);
        if (Mconstant1(i)<Mcost1(i))
            mul(i)=mul(i);
        else

            y1=gaussmf(Mcost1,[0.5 0 0 1]);
            y2=gaussmf(Mdcost1,[0.5 0 0 1]);
            y3=gaussmf(Mcost1,[0.5 1 0 0]);
            y4=gaussmf(Mdcost1,[0.5 1 0 0]);

            r1=min(y1,y2);
            r2=min(y3,y2);
            r3=min(y1,y4);
            r4=min(y3,y4);

mul(i)=(r1(i)*ms+r2(i)*mm1+r3(i)*mm2+r4(i)*ml)/(r1(i)+r2(i)
+r3(i)+r4(i));
        end
    end

    Mconstant1(i)=Mcost1(i);%%actualitzo variables
    mul(i)=mul(i); %%actualitzo variables
end

```



### A.3. Esquema del -Predistorsionador digital utilitzat en System Generator

#### A.3.1. Amplificador de potència

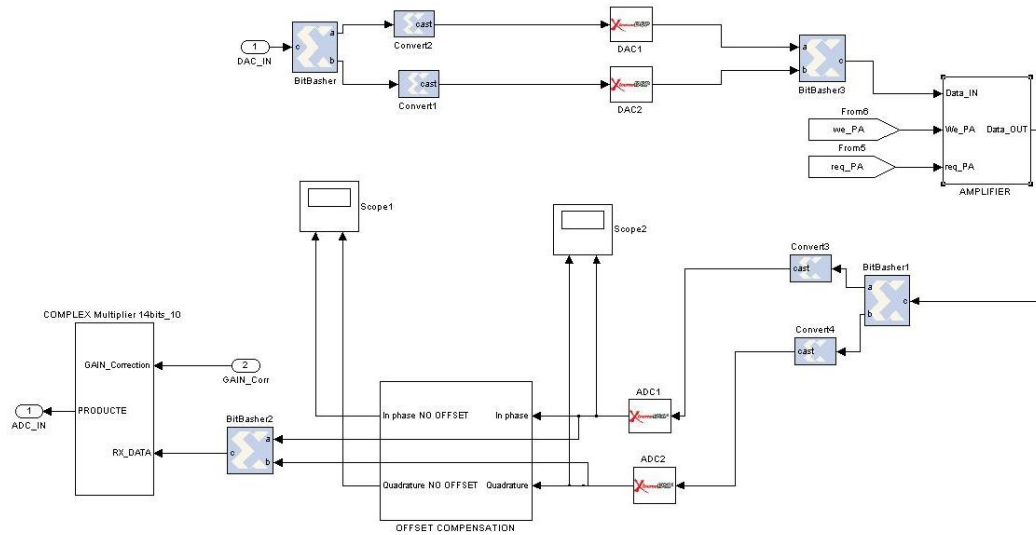


Figura A.1: Esquema d'amplificador de potència implementat en System Generator

#### A.3.2. Identificador

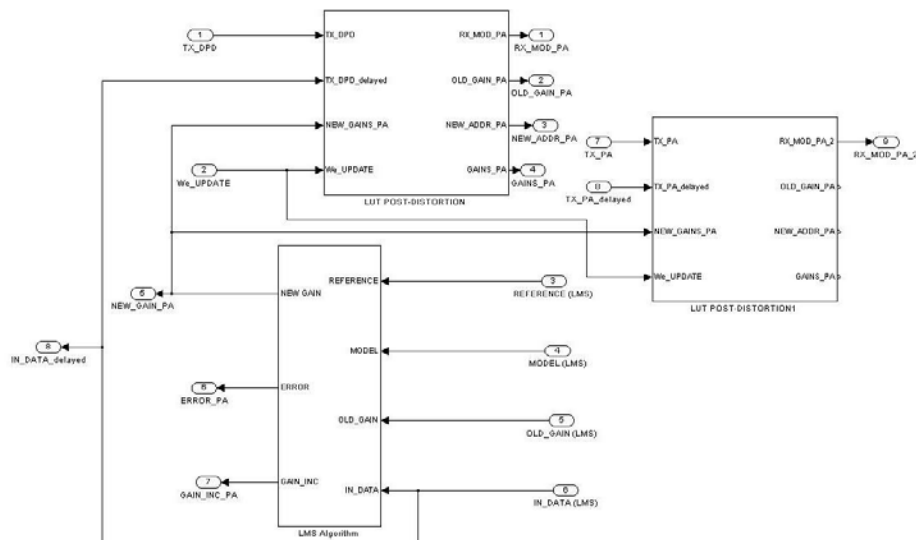


Figura A.2: Esquema d'identificador implementat en System Generator

### A.3.3. Postdistorsionador

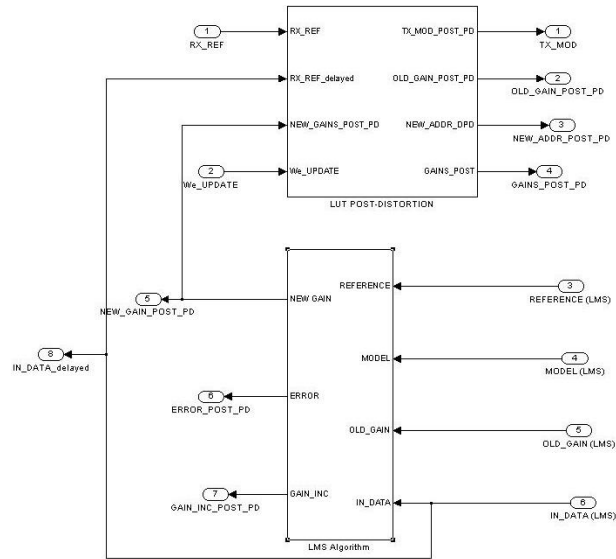


Figura A.3: Esquema de postdistorsionador implementat en System Generator

### A.3.4. Predistorsionador

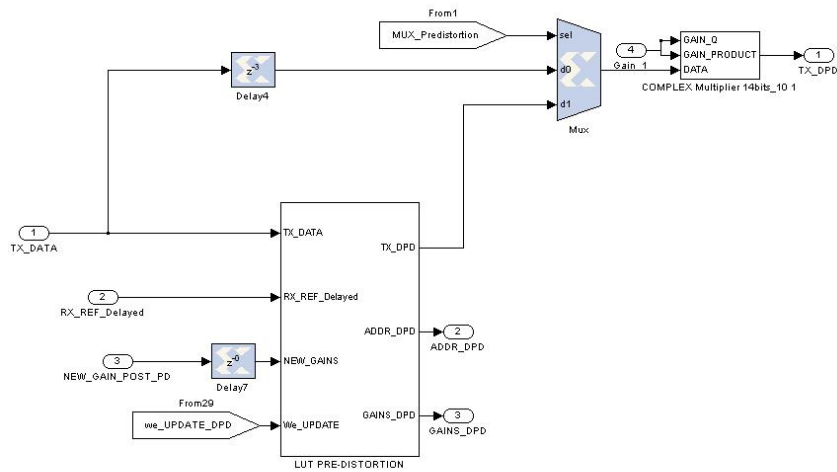


Figura A.3: Esquema de predistorsionador implementat en System Generator