

# **Escola Universitària Politécnica de Mataró**

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA  
DE CATALUNYA**

**Grau en enginyeria electrònica industrial i automàtica**

**Grau en enginyeria mecànica**

## **CONNEXIÓ A XARXA D'UN INVERSOR DE TRES NIVELLS AMB CONTROL PREDICTIU**

**Memòria**

**ALEJANDRO VIVARACHO PORCEL  
PONENT: SALVADOR ALEPUZ MENÉNDEZ**

PRIMAVERA 2015



**TecnoCampus  
Mataró-Maresme**



## **Agraïments**

Al meu ponent Salvador Alepuz, pel seu suport i ajuda al llarg de tot el desenvolupament del projecte. A Alex Calle, la reunió amb ell va suposar un abans i un després en aquest projecte. A Antonio Ortel, per sempre donar un cop de mà amb els problemes informàtics.

I molt especialment als meus pares, gràcies per tot.



## **Resum**

El present projecte consisteix en la millora de rendiment i funcions d'un projecte precedent. El projecte anterior presenta un convertidor CC/CA controlat per control predictiu que quan es sotmet a simulació funciona de forma molt eficient i efectiva però al fer-ho experimentalment apareixen diferències notables entre el senyal de referència i el real. Per a solucionar aquests errors es prenen les mesures possibles per a reduir el temps de resposta. Com a millora funcional s'aplica el circuit a la xarxa elèctrica com a generador elèctric fent servir el control predictiu i un PLL per a la sincronització.

## **Resumen**

El presente proyecto consiste en la mejora de rendimiento y funciones de un proyecto precedente. El proyecto anterior presenta un convertidor CC/CA controlado por control predictivo que cuando se somete a simulación funciona de forma muy eficiente y efectiva pero al hacerlo experimentalmente aparecen diferencias notables entre el señal de referencia i el real. Para solucionar estos errores se toman las medidas posibles para reducir el tiempo de respuesta. Como mejora funcional se aplica el circuito a la red eléctrica como generador eléctrico utilizando el control predictivo y un PLL para la sincronización.

## **Abstract**

This project consists of the improvement of performance and features of a preceding project. The previous project presents a DC/AC converter controlled by predictive control that works very efficiently and effectively when it operates under simulation but differences appear between reference and real signal when it operates experimentally.

In order to solve this errors all the possible measure are taken to reduce the turnaround time. As functional improvement the circuit becomes applied to mains as electric generator using the predictive control and a PLL for the synchronization.



# Índex.

Índex de figures. ....	III
Índex de taules. ....	V
Glossari de termes. ....	VII
1. Objectius. ....	1
1.1. Propòsit. ....	1
1.2. Finalitat. ....	1
1.3. Objecte. ....	1
1.4. Abast. ....	1
2. Antecedents i necessitats d'informació. ....	3
2.1. Antecedent. ....	3
2.2. Inversor CC/CA trifàsic de tres nivells amb topologia NPC. ....	4
2.3. Commutació de IGBTs en el convertidor CC/CA. ....	5
2.4. Control predictiu. ....	6
2.5. Llenguatge de programació C ....	11
2.6. Software Matlab-Simulink. ....	17
2.6.1. Funció de Subsistema, S-Function. ....	17
2.6.2. Estructura MEX S-Function en llenguatge Cl. ....	18
2.6.3. Funcions de Matlab-Simulink avançades. ....	21
2.7. DSP DS1104. ....	24
2.7.1. Hardware DS1104. ....	24
2.7.1.1. Connectors BNC. ....	25
2.7.1.2. Connectors I/O digital. ....	26
2.7.2. Software DS1104. ....	27
3. Procés experimental. ....	31
4. Resultats experimentals. ....	39
4.1. Convertidor amb càrrega RL. ....	39
4.2. Convertidor amb sincronització a xarxa. ....	43
5. Calendari del projecte. ....	47
6. Impacte mediambiental. ....	51
7. Conclusions. ....	53
8. Referències. ....	55





## Índex de figures.

Fig. 2.1. Esquema convertidor CC/CA de tres nivells amb topologia NPC.....	4
Fig. 2.2. Esquema de transistor IGBT.....	5
Fig. 2.3. Possibles vectors de tensió i estats de commutació generats per l'inversor de tres nivells.....	7
Fig. 2.4. Esquema general del convertidor CC/CA de tres nivells sincronitzat a xarxa per control predictiu.....	8
Fig. 2.5. Exemple de comportament de control predictiu a un pas.....	9
Fig. 2.6. Exemple de comportament de control predictiu a dos passos.....	10
Fig. 2.7. Connexions DSP DS1104.....	24
Fig. 2.8. Blocs d'entrada analògica independent i multiplexada.....	29
Fig. 3.1. Esquema electrònic del model amb càrrega RL.....	31
Fig. 3.2. Vista general exemple control DSpace.....	32
Fig. 3.3. Esquema per la correcció d'error en una senyal.....	33
Fig. 3.4. Esquema electrònic del model amb sincronització a xarxa.....	35
Fig. 3.5. Esquema electrònic de seguretat de xarxa.....	36
Fig. 4.1. Temps de resposta del model amb càrrega RL.....	39
Fig. 4.2. Tensions $v_p$ i $v_n$ per $v_{pn}$ 30V del model amb càrrega RL.....	40
Fig. 4.3. Tensions $v_p$ i $v_n$ per $v_{pn}$ 100V del model amb càrrega RL.....	40
Fig. 4.4. Corrents per $v_{pn}$ 30V, I 0.5A, $\Theta$ 0rad.....	41
Fig. 4.5. Corrents per $v_{pn}$ 100V, I 1.5A, $\Theta$ 0rad.....	41
Fig. 4.6. Resposta del model RL per flanc de pujada de corrent.....	42
Fig. 4.7. Resposta del model RL per flanc de baixada de corrent.....	42

Fig. 4.8. Temps de resposta del model amb sincronització a xarxa.....	43
Fig. 4.9. Tensions $v_p$ i $v_n$ del model amb sincronització a xarxa.....	43
Fig. 4.10. Tensions $v_{abc}$ i angle de fase del model amb sincronització a xarxa.....	44
Fig. 4.11. Corrents $i_{abc}$ del model amb sincronització a xarxa.....	44
Fig. 4.12. Tensió $v_a$ i corrent $i_a$ per $\Theta$ nul·la.....	45
Fig. 4.13. Tensió $v_a$ i corrent $i_a$ per $\Theta = +1\text{rad}$ .....	45
Fig. 4.14. Tensió $v_a$ i corrent $i_a$ per $\Theta = -1\text{rad}$ .....	46
Fig. 5.1. Gràfica temporal del calendari del projecte.....	49

## **Índex de taules.**

Taula 2.1. Corrents per branca del convertidor CC/CA de tres nivells.....	6
Taula 2.2. Configuració de IGBTs per branca.....	6
Taula 2.3. Operacions de llenguatge C.....	12
Taula 2.4. Connectors BNC de la DS1004.....	25
Taula 2.5. Connector I/O Digital.....	26
Taula 3.1. Optimització d'entrades analògiques.....	37
Taula 5.1. Calendari del projecte.....	47



## Glossari de termes.

CC	Corrent Continu
CA	Corrent Altern
<i>PLL</i>	<i>Llaç de seguiment de fase</i>
C, C++	Llenguatge de programació C, C++
DVD	Disc versàtil digital
RL	Resistència i bobina connectat en sèrie
<i>PWM</i>	<i>Modulació per ample de polsos</i>
<i>EUPMT</i>	<i>Escola universitària Politècnica de Mataró</i>
DSP	Processador Digital de Senyals
NPC	Subjectat al punt neutre
$V_{pn}$	Tensió continua d'alimentació del convertidor
A, B, C	Sortides del convertidor
IGBT, S	Transistor Bipolar de Porta Aïllada
G	Porta de la IGBT
$V_{GE}$	Tensió porta-emissor
$V_{CE}$	Tensió col·lector-emissor
$V, V_o$	<i>Tensió, tensió de sortida</i>
$I, I_o$	<i>Corrent, corrent de sortida</i>
$f$	<i>Freqüència</i>
$V_p$	<i>Tensió entre P i 0</i>

## VIII

$V_n$	<i>Tensió entre N i 0</i>
$P, N$	<i>Rail positiu, negatiu</i>
$0$	<i>Punt neutre</i>
$a', b', c'$	<i>Punts de sortida del convertidor entre la bobina i la xarxa.</i>
$V_s(k)$	<i>Tensió de la xarxa</i>
$i_{abc}$	<i>Corrents de sortida del convertidor</i>
$i_\beta, i(k)$	<i>Corrent real</i>
$i_\beta^*, i^*(k)$	<i>Corrent de referència</i>
$i_\beta^p, i(k+1)$	<i>Predicció de corrent</i>
$t_k, t_{k+1}$	<i>Temps en l' instant k, k+1</i>
$T_s$	<i>Temps de mostreig</i>
$BNC$	<i>Connector tipus Bayonet Neill-Concelman</i>
$CP$	<i>Connector de la targeta DS1104</i>
$v_a, v_b, v_c$	<i>Tensions en els punts A, B, C</i>
$i_a, i_b, i_c$	<i>Corrents en el punts A, B, C</i>
$GND$	<i>Terra</i>
$IO, I/O$	<i>Entrada/Sortida</i>
$TFG$	<i>Treball Final de Grau</i>
$R+D$	<i>Recerca i Disseny</i>

# **1. Objectius.**

## **1.1. Propòsit.**

Desenvolupament d'un programa sobre Matlab-Simulink amb llenguatge de programació C, per fer el control predictiu d'un inversor de tres nivells i la sincronització a la xarxa.

## **1.2. Finalitat.**

Reduir el temps de càlcul i per tant també el temps de mostreig per tal d'aconseguir un seguiment del senyal referent més precís i eficient i acoblar el sistema a la xarxa trifàsica a mode de generador.

## **1.3. Objecte.**

Al acabar el projecte els recursos físics que es generen són, juntament amb el document de memòria del treball, tres programes en format de DVD dels tres models finals amb els que s'experimenta al llarg del projecte, l'experiment amb càrrega RL i la simulació i experiment amb sincronització a xarxa llegibles amb la plataforma Matlab-Simulink. També es deixa el muntatge de hardware de l'experiment final corresponent a la sincronització a la xarxa elèctrica.

## **1.4. Abast.**

L'abast del projecte comprèn la reprogramació del programa precedent, on s'inclourà la substitució de les sortides PWM per sortides digitals, integració de blocs en C i ampliació de funcionalitat amb el disseny d'un PLL per a la sincronització a la xarxa.

No es contemplarà fer un redisseny de hardware o provar altres estratègies de control que no siguin les de control predictiu d'un o dos passos.





## **2. Antecedents i necessitats d'informació.**

En aquest apartat s'introdueix el projecte, s'analitza el projecte precedent i es donen a conèixer les informacions teòriques bàsiques per a entendre el desenvolupament del projecte.

### **2.1. Antecedent.**

El títol del projecte precedent [1] a aquest és “Control predictiu d'un inversor de tres nivells programat amb llenguatge C”, finalitzat l'any 2012 per Òscar García Benítez amb ponent Salvador Alepuz a la universitat Tecnocampus EUPMT.

En aquest projecte s'utilitza el control predictiu a un i dos passos per a controlar un convertidor CC/CA que ha de seguir el corrent trifàsic sinusoidal de referència. On s'afegeixen dues consignes, amplitud i angle de fase, per controlar la potència activa i reactiva que es produeix.

El programa es va dissenyar amb el programa Matlab-Simulink, en forma de diagrama de blocs, on el bloc de control es va programar amb llenguatge C.

Al final del projecte es va aconseguir fer una simulació per als dos tipus de control predictiu i també realitzar-ho amb control predictiu a un pas a nivell experimental, utilitzant components reals i executant el control en una targeta DSP.

Les conclusions extretes del projecte van ser que el control predictiu a dos passos tot i ser més laboriós funcionava amb més precisió, però encara hi havia marge de millora. Es podien fer alguns canvis a nivell bàsicament de software per a reduir el temps de mostreig, així com ampliar les funcionalitats del programa, com la sincronització a la xarxa.

## 2.2. Inversor CC/CA trifàsic de tres nivells amb topologia NPC.

L'inversor CC/CA que es mostra a la figura següent és el circuit electrònic que transforma tensió contínua de la entrada ( $V_{pn}$ ) en tensió alterna quadrada a cadascuna de les sortides (A, B o C) segons l'estat de commutació de les quatre IGBT (S) que cada branca [1].

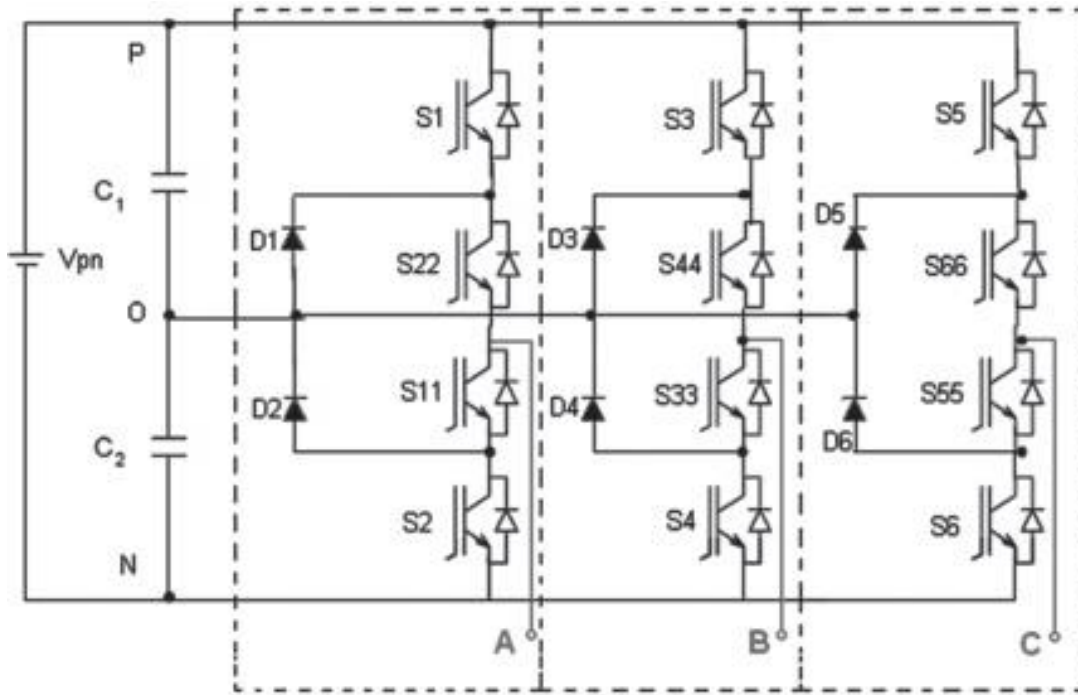


Fig. 2.1. Esquema convertidor CC/CA de tres nivells amb topologia NPC.

D'aquesta manera les sortides A, B i C són independents les unes de les altres i tenen un valor positiu (P), negatiu (N) o neutre (0) segons el control que s'empra sobre les IGBT que determina el seu estat (ON/OFF).

### 2.3. Commutació de IGBTs en el convertidor CC/CA.

Les IGBTs són semiconductors que s'utilitzen com a interruptors controlats per tensió en el pin de porta (G – gate) [2].

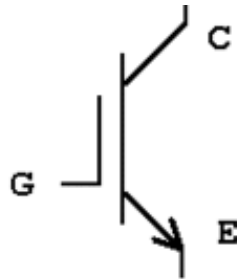


Fig. 2.2. Esquema de transistor IGBT.

Característiques de les IGBT:

- Control per tensió  $\rightarrow \begin{cases} V_{GE} > V_T & \text{ON (curtcircuit)} \\ V_{GE} < V_T & \text{OFF (circuit obert)} \end{cases}$
- Potència  $\rightarrow V_{CE}$  constant en conducció
- Freqüència  $\rightarrow f < 100\text{kHz}$

Els temps de mostreig necessaris per aquest projecte són al voltant de  $100\mu\text{s}$  per tant les IGBT, que donen una velocitat de resposta fins a  $10\mu\text{s}$  són perfectes en aquesta situació.

Per a connectar les branques del convertidor a voltatge Positiu, Negatiu o Neutre s'ha d'establir la configuració d'estats de IGBT que facin aquesta funció. En a la següent taula es poden observar les configuracions de les IGBT que s'utilitzen per connectar a P, 0 o N i la direcció que pren el corrent segons en el cas de ser positiu o negatiu [4].

	$V_o = +V_{pn}/2$ (connectat a positiu)	$V_o = -V_{pn}/2$ (connectat a negatiu)	$V_o = 0$ (connectat a zero)
$I_o > 0$			
$I_o < 0$			

Taula 2.1. Corrents per branca del convertidor CC/CA de tres nivells.

Existeixen més configuracions nominalment possibles però manquen de sentit i utilitat, ja que en bloqueig, i sempre que estigui equilibrat, els semiconductors han de bloquejar la meitat del bus de continua i per tant no queda altra solució que aquestes.

En la següent taula-resum es veu la configuració de IGBTs establert per a cada branca i cada nivell de tensió de sortida.

Branca	$V_o$	S1	S22	S11	S2	S3	S44	S33	S4	S5	S66	S55	S6
A	P	ON	ON	OFF	OFF	-	-	-	-	-	-	-	-
	N	OFF	OFF	ON	ON	-	-	-	-	-	-	-	-
	0	OFF	ON	ON	OFF	-	-	-	-	-	-	-	-
B	P	-	-	-	-	ON	ON	OFF	OFF	-	-	-	-
	N	-	-	-	-	OFF	OFF	ON	ON	-	-	-	-
	0	-	-	-	-	OFF	ON	ON	OFF	-	-	-	-
C	P	-	-	-	-	-	-	-	-	ON	ON	OFF	OFF
	N	-	-	-	-	-	-	-	-	OFF	OFF	ON	ON
	0	-	-	-	-	-	-	-	-	OFF	ON	ON	OFF

Taula 2.2. Configuració de IGBTs per branca.

Amb aquests possibles estats de les IGBTs s'obtenen un total de 27 possibles combinacions que resulten en 19 diferents vectors de tensió [3], tal i com es mostra a la propera figura [5].

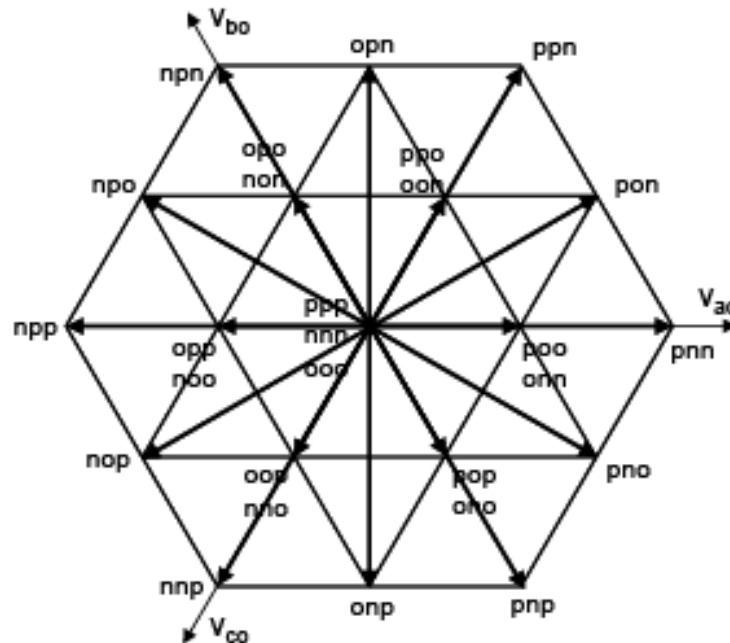


Fig. 2.3. Possibles vectors de tensió i estats de commutació generats per l'inversor de tres nivells.

Éssent p connectat a positiu, n a negatiu, i o a neutre i la posició corresponent a la branca, per exemple el vector anomenat pno vol dir que la branca A està connectada a positiu, la B a negatiu i la C al punt 0; o el que és el mateix, seguint la anterior taula, les IGBT S1, S22, S33, S4, S66 i S55 estan ON i les IGBT S11, S2, S3, S44, S5 i S6 OFF.

## 2.4. Control predictiu.

El control predictiu [6], [7] és una estratègia de control que suposa una gran opció alhora de controlar processos multivariable tant de sortida com d'entrada. Es basa en veure la tendència de les variables i tractar-les convenientment de cara a corregir la diferència entre la variable de referència i la variable real.

En la següent figura [3] podem veure de forma general el tractament de variables que s'empra per a controlar mitjançant control predictiu el convertidor de tres nivells.

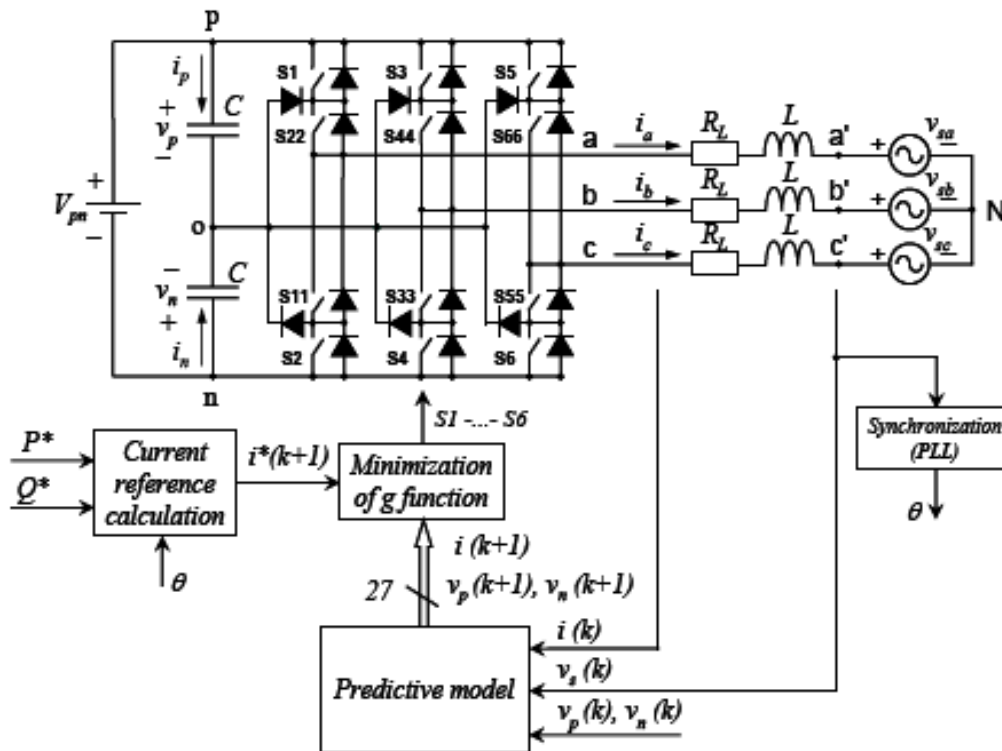


Fig. 2.4. Esquema general del convertidor CC/CA de tres nivells sincronitzat a xarxa per control predictiu.

El sistema actua com a generador d'energia elèctrica a la xarxa trifàsica. El convertidor, mitjançant dos condensadors iguals, obtenen les tensions  $V_p$ ,  $V_n$  i el punt neutre (0), que arribaran als punts A, B, C segons la commutació de les IGBT. En sèrie es col·loquen unes bobines per a la obtenció de la ona trifàsica sinusoidal típica de la xarxa elèctrica, col·locada en triangle en aquest punt, anomenat a', b' i c' i on s'extreu  $V_s(k)$  com a variable d'entrada per al model predictiu i pel PLL per la sincronització.

El model predictiu té com a variables d'entrada els corrents de sortida del convertidor ( $i_a$ ,  $i_b$ ,  $i_c$ ), les tensions  $V_p$  i  $V_n$  i la tensió trifàsica a sincronitzar ( $V_s(k)$ ).

Juntament amb el corrent de referència, calculat a partir de la potència activa, la reactiva i l'angle de desfasament amb la xarxa es calcula quin és, dels 27 vectors de tensió possibles, el vector òptim que deriva en l'estat de les IGBT per tal d'aconseguir igualar tensions i desfasar corrents respecte el voltatge si s'escau per a modificar el factor de potència i

aconseguir una càrrega inductiva o capacitiva i així sincronitzar de la forma més convenient amb la xarxa segons les necessitats del moment, el control predictiu també equilibra el punt neutre entre els condensadors de forma que es mantingui estable, sent  $V_p = -V_n$ .

En el cas d'aquest projecte s'aplica sobre els corrents dels punts a, b i c ( $i_{abc}$ ) i també sobre les tensions  $V_p$  i  $V_n$ . En la següent figura [6] es representa gràficament la evolució en el temps d'una única variable de corrent tractada amb control predictiu a un pas, sent  $i_\beta$  el corrent real (en la figura anterior  $i(k)$ ),  $i_\beta^*$  el corrent de referència (anteriorment anomenat  $i^*(k)$ ) i  $i_\beta^p$  la predicció (abans  $i(k+1)$ ).

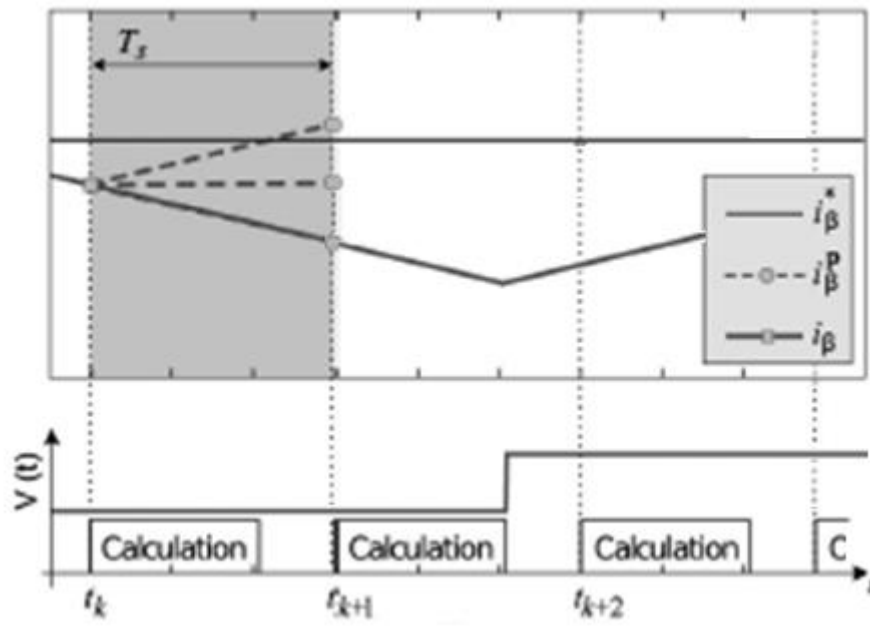


Fig. 2.5. Exemple de comportament de control predictiu a un pas.

La referència és la variable  $i_\beta^*$ , que per simplificar és constant, en l'instant  $t_k$  la variable real  $i_\beta$  està disminuint el seu valor segons l'estat actual de les IGBTs de la branca, distanciant-se cada cop més de la referència per la banda negativa, encara a l'instant  $t_k$  el control predictiu comprova l'estat actual de les IGBTs i la variable real i fa tres prediccions en corresponents a connectar el punt a positiu, negatiu o neutre i actua d'acord amb la que s'apropa més a la referència, que en aquest moment és la de pendent positiva. La acció, però, no és immediata, el primer temps de mostreig (entre  $t_k$  i  $t_{k+1}$ ) el control predictiu a un

pas analitza l'estat de les IGBT i a l'instant  $t_{k+1}$  ordena el canvi d'estat per a connectar a P però encara es retarda el temps de càlcul i origina errors, de forma que el canvi d'estat sempre s'executa endarrerit l'equivalent a la suma del temps de mostreig i el temps de resposta.

Pel control predictiu a dos passos s'aconsegueix reduir aquest temps en un temps de mostreig.

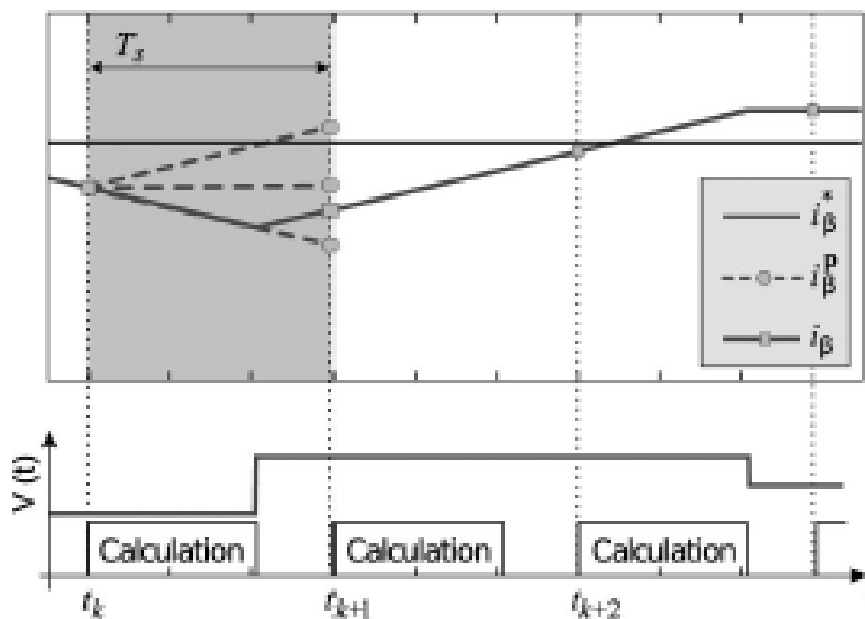


Fig. 2.6. Exemple de comportament de control predictiu a dos passos.

El control predictiu a dos passos fa la predicció a dos temps de mostreig, avançant un  $T_s$  respecte el control predictiu a un pas.

En la il·lustració anterior, el sistema arriba a l'instant  $t_k$  en la mateixa situació que arribava amb anterioritat però amb la diferència que sap com evolucionarà el corrent en aquest interval de temps, d'aquesta manera en el mateix instant  $t_k$  pot governar l'estat de les IGBT que canviaran un cop passat el temps de resposta i d'aquesta manera disminuir l'error entre la variable real i la referència.

El control, tant sigui predictiu o no, no és perfecte, per què s'aplica en un entorn real no ideal i per aquest motiu la referència i la variable real mai seran iguals. Per a reduir aquest error al màxim el que s'ha de fer és reduir al màxim el temps de resposta i el temps de mostreig.



## 2.5. Llenguatge de programació C.

Aquest llenguatge es caracteritza per ser un llenguatge de propòsit general de fàcil sintaxis, amb control de flux i un bon conjunt d'operacions. Aquestes característiques proporcionen un camp d'aplicació molt ampli i una bona facilitat per a aprendre a utilitzar-lo.

A continuació s'expliquen alguns conceptes bàsics claus per començar a programar en el llenguatge C amb el focus d'atenció en les eines que s'acaben utilitzant en l'actual projecte i també en l'antecedent [8].

### Importació d'altres continguts d'arxius

`#include<llibreria.h>`: mitjançant aquesta ordre s'importen els continguts (funcions, variables, tipologies, etc.) d'aquest arxiu a l'arxiu font. El llenguatge C té la convenció d'utilitzar extensions `.h` per arxius de capçaleres ("header").

Exemple: `"#include<stdio.h>"`.

### Declaració de constants

`#define nom_constant valor`: s'estableix el valor d'una constant, poden ser de tipus real o enter.

Exemple tipus enter: `"#define R 5"`. Exemple tipus real: `"#define R 5.0"`.

### Declaració de variables

Enteres:

Declara una variable del tipus enter, amb una mida de 32 bits. Pot inicialitzar-se el seu valor en la mateixa línia o no.

Exemple: `int Vp = 100;`

Reals:

Declara una variable del tipus real. Pot inicialitzar-se el seu valor en la mateixa línia o no.

Exemple: `real iref;`

Punts flotants:

Es declaren mitjançant “float” o “double” depenent de la mida que es necessita. En aquest treball només s'utilitza per a la declaració de vectors, tot i que hi ha diverses formes de fer-ho que no necessiten aquest d'aquest tipus.

Exemple: `float vector[1][3]={0.0, 1.0, 2.0} ;`

Altres:

Hi ha més variables però no s'utilitzen, almenys de forma directa, en aquest projecte, com per exemple caràcters (char), booleans (bool), void, enumerat (enum), etc.

### Operacions

A continuació es mostra una taula amb les principals operacions que es poden fer per operar variables en el llenguatge de programació C.

Símbol	Operació	Símbol	Operació
+, -	Suma, resta	&&,   , !	AND, OR, NOT lògic
++, --	Increment, decrement	==, !=	Igualtat, desigualtat
*, /, %	Multiplicació, divisió, mòdul	<, <=	Menor que, menor o igual que
&,	AND, OR booleà	>, >=	Major que, major o igual que

Taula 2.3. Operacions de llenguatge C.

## Comentaris

Els comentaris són observacions que s'escriuen en el propi codi com a forma de guia o ajuda per a la seva entesa. En el llenguatge C es poden escriure comentaris de dues formes, el primer obre comentari amb el símbol “/\*” i tanca amb “\*/”, el que hi hagi escrit entre aquests dos símbols és el comentari. El segon comença el comentari utilitzant dues barres “//” i tanca al canviar de línia de codi.

Exemple 1:

```
Codi; /* Comentari 1
```

```
Comentari 2 */
```

```
Codi;
```

Exemple 2:

```
Codi; //Comentari
```

```
Codi;
```

## Funcions elementals

A continuació es veuran dues funcions claus, “if” i “for”, el seu format i la seva utilitat mitjançant un exemple.

- Funció for:

La funció “for” és una sentència iterativa, també anomenat bucle, s'utilitza per a per a executar una sèrie d'ordres fins a assolir una condició desitjada.

Format:

```
for (inicialització; test; increment) { Codi; }
```

Exemple:

```
for (i = 0; i <= 10; i++) { var=var+i; }
```

Aquest senzill exemple de la funció for està guiat per la variable “i”. S’inicialitza la variable a zero, i s’incrementa en una unitat a cada iteració fins a que el test deixi de complir-se ( $i \in \{0,1,2,\dots,10\}$ ), per a cada iteració s’executarà el codi, el valor de la variable “var” sumarà el valor de la propia variable “i” .

- Funció if:

La funció “if” és una manera de sotmetre a una condició a alguna part del cos del programa.

Format:

```
if (test) {codi;} else if (test2){codi2;} else{codi3}
```

Exemple 1:

```
if(a>3) { b=0;} else if (a=3) {b=5;} else {b=2;}
```

Aquest exemple assigna el valor de “b” segons el valor de “a”. Si a és major de tres “b” serà zero, si “a” és igual a tres “b” obtindrà el valor 5 i en el cas que no es doni cap de les condicions anteriors (és a dir si “a” és menor de tres) llavors el valor de “a” serà 2.

Exemple 2:

```
n=(x>3 ? 4 : 0)
```

Aquest altre exemple és una simplificació de la funció “if”, amb el format (Test ? valor\_si\_test\_TRUE : valor\_si\_test\_FALSE) i seria una ordre equivalent “if” següent: `if (x>3) {n=4} else {n=0}`.

- Funció While:

La sentència és iterativa, semblant a la funció “for” en quant a funcionalitat però diferent en format.

Format:

```
While (test) { codi; }
```

Exemple:

```
While (var<=valor) { var++; valor--; }
```

En aquest exemple es fa una iteració, en el que es comparen les variables “a” i “valor”, per cada iteració s’incrementa en una unitat el valor de “a” i es disminueix “valor”, quan “a” sigui major o igual que la variable “valor” s’acabarà la iteració.

#### Variabls locals i variabls globals:

Al igual que cada variable ha de declarar-se com a entera, real, booleana, void, etc. també té una característica que determina si la variable s’ha de guardar externament al codi, ja sigui per utilitzar-la en un altre part del programa o per a reutilitzar-la en el mateix bloc la següent vegada que aquest s’executi. Si la variable s’inicialitza cada vegada i per tant només modifica el propi bloc programat en C es diu que és una variable local, en cas que la variable es salvi un cop es finalitza d’executar el bloc per utilitzar-la en el conjunt del programa es diu que és una variable global.

La forma de declarar si una variable és local o global és depenent del lloc on es fa la declaració, si es vol que la variable sigui local llavors es declara amb normalitat, a dintre de la funció principal “main”; si pel contrari es vol que sigui una variable global la declaració ha de fer-se abans d’aquesta funció, al mateix lloc on s’escriuen els “#define” i “#include”.

La forma de cridar una variable global existent a un bloc C en el que no està declarada és mitjançant la ordre “extern”, amb el format “extern tipus nom” per exemple “extern int amplada”.

La ordre contrària a “extern” és “static” aquesta ordre determina que la variable, o més sovint la funció que segueix a la ordre no ha d'estar disponible per altres blocs del programa, té el mateix format que la funció “extern”.

### Funcions void

Algunes funcions poden portar aquesta ordre davant, en comptes d'un tipus de variable, aquest prefix determina que aquesta funció no ha de retornar cap variable i per tant no necessita tipus.

### Punters

Les variables d'entrada i sortida de blocs tampoc tenen un tipus determinat, són punters, o dit d'una altra manera, són adreces a dintre de l'ordinador. Per tractar aquestes variables primer s'hauran d'adquirir mitjançant una funció característica i després transformar-les a variables que es puguin operar (enters, reals, ...).

La transformació és tan senzilla com accedir a la ubicació del punter i assignar-la a una variable ja descrita, la forma d'accedir al punter és: “ \*nomdelpunter[índex]; ”

L'asterisc determina que és un punter, i l'índex s'utilitza si és una entrada multiplexada, en cas que no ho sigui en índex s'escriu “0” (zero), en aquest projecte no s'utilitzen, així sempre s'haurà d'indexar a zero.

Exemple d'assignació a una variable d'un punter:

```
Vp = *uPtrs3[0];
```

La declaració del punter adquirit es fa mitjançant la ordre `InputRealPtrsType`, la funció per a la adquisició és `ssGetInputPortRealSignalPtrs(S, index)` i la d'escriptura `ssGetOutputPortRealSignal(S, index)`, aquestes funcions s'explicaran, junt amb altres funcions a l'apartat corresponent a les S-Functions de Simulink.

## 2.6. Software Matlab-Simulink.

Simulink és un entorn de diagrama de blocs del Matlab i una eina per simulació i experiments. Conté llibreries de blocs predefinides i personalitzables per a modelització de sistemes en temps continu i discret, editor gràfic per construir i gestionar els diagrames de blocs, eines de codi per importar programes en llenguatge C o C++ a models i compatibilitat amb arxius amb la extensió `.sdf` que serà la tipologia d'arxius per a fer projectes amb el programa DSpace de la targeta DSP DS-1104 que s'utilitza per a fer aquest projecte.

### 2.6.1. Funció de Subsistema, S-Function.

Les S-Function (funcions de Sistema) són una eina molt útil per ampliar les capacitats de Simulink, és una descripció de bloc de Simulink escrita en C, C++ o Fortran [9].

S'ha de compilar com a extensió `.mex`, d'aquesta manera es vincula a Matlab i el pot carregar i executar de forma automàtica, com si d'un bloc predeterminat es tractés.

N'hi ha de cinc formes diferents d'implementar una S-Function: S-Function de Matlab de nivell 1, S-Function de Matlab de nivell 2, MEX S-Function, constructor d'S-Functions i eina de codi heretat, a continuació es comparen les característiques de les cinc possibles opcions a l'hora d'implementar-les.

#### S-Function de nivell 1:

És la S-Function més bàsica i senzilla però manca d'accessos a algunes interfícies de programació. Permet la generació de funcions que només tenen temps de mostreig continu.

### S-Function de nivell 2:

Proporciona accés a un conjunt més extensiu d'interfícies de programació i permet la generació de codi. Permet la generació de funcions amb temps de mostreig continu i discret. Es recomana utilitzar aquest tipus d'implementació si es vol utilitzar la funció en Matlab.

### MEX S-function:

Aquest tipus d'implementació d'S-Function té la major flexibilitat de programació, pots implementar l'algoritme directament escrivint el codi amb llenguatge C, C++ o Fortran. Aquesta opció requereix un coneixement avançat de la interfície de programació de la S-Function així com el llenguatge de programació del compilador objectiu.

### Constructor d'S-Function:

És una interfície gràfica per programar un petit conjunt de funcionalitats de les S-Function, de forma que no s'ha d'utilitzar la interfície de programació.

### Eina de codi heretat:

És un conjunt d'ordres del Matlab que ajuden a crear una S-Function per incorporar codi C/C++ heretat. Proporciona menys accessos que qualsevol de les anteriors implementacions tractades.

## **2.6.2. Estructura MEX S-Function en llenguatge C.**

Es decideix que el millor mètode per a utilitzar al present projecte és una MEX S-Function, tot i ser la més complexa ofereix la major varietat de possibilitats i per tant major possibilitat a l'hora d'adaptar-se a l'entorn de treball en aquest i en possibles futurs projectes [9].



El format general d'una MEX S-function en C es mostra a continuació.

```
#define S_FUNCTION_NAME nom_de_la_funció_aquí

#define S_FUNCTION_LEVEL 2

#include "simstruc.h"

static void mdlInitializeSizes(SimStruct *S)

{

}

/*codi llenguatge C*/

static void mdlTerminate(SimStruct *S)

{

}

#ifdef MATLAB_MEX_FILE

#include "simulink.c"

#else

#include "cg_sfun.h"

#endif
```

Es procedeix a desglossar l'estructura línia a línia per entendre el procediment que segueix el motor de Simulink i el significat de cada ordre.

```
#define S_FUNCTION_NAME nom_de_la_funció_aquí
```

Es defineix el nom de la S-Function.

```
#define S_FUNCTION_LEVEL 2
```

Defineix el nivell de la S-Function, nivell 2.

```
#include "simstruc.h"
```

Inclou l'arxiu simstruct.h, capçalera en llenguatge C que defineix la estructura de dades de Simulink i les macros d'accés, els paràmetres i outputs del bloc.

```
static void mdlInitializeSizes(SimStruct *S) { ... }
```

Especifica el nombre d'entrades, sortides, estats, paràmetres i altres característiques de la S-Function. És la primera funció que s'executa. A l'interior d'aquesta funció es posaran altres per a la inicialització.

```
/*codi llenguatge C*/
```

En aquesta part de la estructura s'escriurà el codi del programa, en general assignació d'entrades, declaració i tracte de variables i assignació de sortides.

```
static void mdlTerminate(SimStruct *S){ ... }
```

Fa alguna acció requerida al final de la simulació.

```
#ifdef MATLAB_MEX_FILE
```

Comprova si l'arxiu s'està compilant com a arxiu MEX.

```
#include "simulink.c"
```

En cas afirmatiu inclou el mecanisme d'interfície d'arxiu MEX.

```
#else
```

En cas negatiu..

```
#include "cg_sfun.h"
```

... inclou un arxiu de generació de codi.

```
#endif
```

Final de l'ifdef i de la S-function.

Aquesta és la estructura bàsica, però es necessiten més funcions predefinides que encara no hem tractat, a continuació s'expliquen aquestes funcions avançades, es pot ampliar la informació amb les pàgines d'ajuda de Matlab, adjuntes al document d'annexos.

### **2.6.3. Funcions de Matlab-Simulink avançades.**

```
ssSetNumSFcnParams(S, n)
```

Especifica el nombre de paràmetres que té el bloc (n), els arguments representen l'anomenat SimStruct (S) i el nombre de paràmetres que té. El segon argument ha de ser enter positiu.

```
ssGetNumSFcnParams(S)
```

Representa el número de paràmetres que s'espera que el bloc tingui.

ssGetSFcnParamsCount(S)

Retorna el nombre de paràmetres del codi del bloc.

ssSetNumInputPorts(S,x), ssSetNumOutputPorts(S, y)

Ordres utilitzades per especificar el número d'entrades(x)/sortides(y) del bloc. Utilitzar en mdlInitializeSizes. El segon terme ha de ser enter positiu.

ssSetInputPortWidth(S, i, a)/ ssSetInputPortWidth(S, i, amplada)

Utilitzar en mdlInitializeSizes. Estableix l'amplada de la entrada/sortida indexada (i, enter positiu). Per entrades/sortides no multiplexades amplada=1.

ssSetInputPortDirectFeedThrough(S, i, b)

Utilitzar en mdlInitializeSizes. Especifica si aquesta s'utilitza en mdlOutputs la entrada indexada (i), l'argument b ha de ser 0 ó 1, en cas de ser 1 vol dir que s'utilitzen, i en cas que sigui 0 no s'utilitzen.

ssSetNumSampleTimes(S, ST)

Determina el nombre de temps de mostreig del bloc, generalment ST=1.

ssSetOptions(S, opció\_1 | opció\_2 | ... | opció\_n)

Especifica les funcions de la S-function. És una funció bastant extensa, més informació als annexos.

mdlInitializeSampleTimes(S)

Funció d'ús obligatori, necessària per determinar el temps de mostreig de la S-Function.

`ssSetSampleTime(S, i, període)`

Indica el període(t) en segons del temps de mostreig indexat (i). En cas d'utilitzar únicament un valor de temps de mostreig com s'ha explicat a la funció `ssSetNumSampleTimes` llavors `i=0`.

`ssSetOffsetTime(S,i,offset)`

Aplica un temps d'offset al temps de mostreig indexat, en segons, si no es vol aplicar `offset=0`.

`mdlStart(S)`

Funció que s'aplica només una vegada al principi de la simulació, útil per a la inicialització d'estats.

`mdlOutputs(S, tid)`

El motor de Simulink executa aquesta funció a cada pas de simulació, s'utilitza com a introducció per al cos del programa, on hi ha el codi que s'ha d'executar a cada pas.

`ssGetInputPortRealSignalPtrs(S, i)`

Adquireix el valor de la entrada indexada (i) de tipus double.

## 2.7. DSP DS1104.

És el dispositiu més important del present projecte, en aquest apartat s'analitza la DS1104 a nivell de hardware i de software.

### 2.7.1. Hardware DS1104.

En la següent il·lustració es mostren els connectors disponibles en la DSP DS1104 [10].

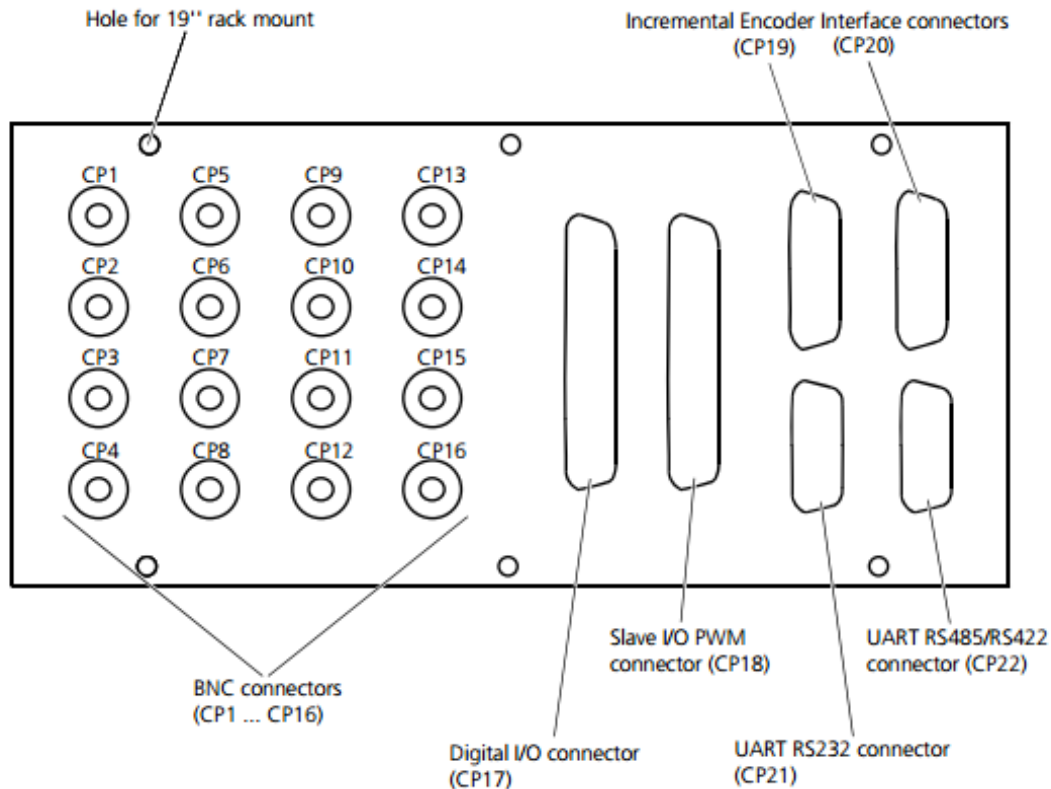


Fig. 2.7. Connexions DSP DS1104.

En l'antecedent s'utilitzaven cinc connectors BNC per a entrada dos sensors de tensió ( $V_p$  i  $V_n$ ) i tres sensors de corrent ( $i_{abc}$ ), el connector PWM com a sortida de senyal per al control de les IGBT, i una sortida digital constant de +5Vcc per a l'alimentació de dispositius de control d' ON/OFF, enable i trip, aquesta última va ser facilitada pel professor ponent i no entrava dintre de l'abast de l'antecedent.

En l'actual projecte, i de forma més ostensible a l'últim model dissenyat, es necessiten més prestacions que en l'anterior, per aquest motiu s'utilitzen vuit connectors BNC, s'afegeixen tres sensors de tensió (vabc) per al PLL de sincronització en xarxa, i es substitueix la sortida per PWM (CP18) per la sortida digital, amb l'objectiu de reduir el  $T_s$ . Com en l'anterior s'utilitza una sortida digital per al banc de control.

### 2.7.1.1. Connectors BNC.

La DSP DS1104 conté 16 connectors BNC femella (CP1 ... CP16), 8 convertidors analògic/digital (entrades analògiques) i 8 convertidors digital/analògic (sortides analògiques), distribuïts tal i com es mostra a la taula següent [10]:

Connector	Senyal	Connector	Senyal	Connector	Senyal	Connector	Senyal
CP1	ADCH1	CP5	ADCH5	CP9	DACH1	CP13	DACH5
CP2	ADCH2	CP6	ADCH6	CP10	DACH2	CP14	DACH6
CP3	ADCH3	CP7	ADCH7	CP11	DACH3	CP15	DACH7
CP4	ADCH4	CP8	ADCH8	CP12	DACH4	CP16	DACH8

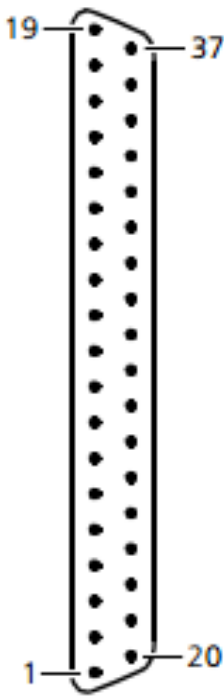
Taula 2.4. Connectors BNC de la DS1004.

Com s'ha explicat amb anterioritat aquests canals només es volen utilitzar com a entrades per a la mesura de corrents i tensions, per tant només s'utilitzaran els CP1 ... CP8.

Una altra característica destacable d'aquests connectors és que mentre que els CP5 ... CP8 són entrades independents CP1 ... CP4 és una entrada multiplexada, aquesta diferència fa que a nivell de software les crides siguin diferents (aquesta petita diferència s'especifica a l'apartat de software de la DSP) i a nivell de funcionalitat les independents són més ràpides mentre que la multiplexada és més pesada. Per aquest motiu s'assegura que les senyals més canviants es connectin a les entrades independents i les més estables a la multiplexada

### 2.7.1.2. Connectors I/O digital.

La DS1104 conté un bus d'entrades/sortides digitals (CP17) amb 37 pins distribuïts segons mostra la taula següent [10]:

Pins del connector	Pin	Senyal	Pin	Senyal
	19	GND	37	+5Vcc
	18	GND	36	+5Vcc
	17	GND	35	GND
	16	GND	34	GND
	15	IO19	33	IO18
	14	IO17	32	IO16
	13	GND	31	GND
	12	IO15	30	IO14
	11	IO13	29	IO12
	10	GND	28	GND
	9	IO11	27	IO10
	8	IO9	26	IO8
	7	GND	25	GND
	6	IO7	24	IO6
	5	IO5	23	IO4
	4	GND	22	GND
	3	IO3	21	IO2
	2	IO1	20	IO0
	1	GND		

Taula 2.5. Connector I/O Digital.

Dels 37 pins hi ha 15 GND, dues sortides de +5Vcc i els altres 20 (IO0 ... IO19) són pins configurables com a entrades o sortides digitals, segons convingui. El mètode de configuració i de llegir/escriure es pot veure en l'apartat següent.

Dels 20 pins IO s'utilitzen únicament 6, que deriven a les portes de les 12 IGBT. El motiu d'aquesta simplificació és per reduir codi i en conseqüència el temps de càlcul i mostreig. Si s'observa la taula 2.2. es pot comprovar que els estats de les IGBT S1 i S11 són



complementaris, així com la S2 i S22, S3 i S33, etc. Per aquest motiu els senyals de les sortides cap a les portes de les IGBT poden reduir-se a la meitat.

Els pins que s'utilitzen pel control de IGBTs són els pins 20, 2, 21, 3, 23 i 5, corresponents a les senyals IO0...IO5. Els pins 35 i 36 són els pins que s'utilitzen per alimentació de dispositius.

Per tal de connectar cadascun dels pins al circuit es necessiten fer soldadures d'estany per de aconseguir una connexió robusta.

### **2.7.2. Software DS1104.**

Els dos programes amb els que es treballa en aquest projecte són Matlab-Simulink i DSpace. Simulink és un programa que ja s'ha explicat amb anterioritat de forma general, en aquest apartat es dóna a conèixer les formes de les que es disposa per comunicar-se amb la DS1104 i com transformar el programa per a tractar-lo amb DSpace.

DSpace és un programa de comunicació i control per la DSP, amb una presentació visual pràctica on es pot veure la evolució de les variables a temps real i fins i tot modificar algunes parts del programa fet prèviament amb Simulink.

Anteriorment s'ha explicat que per a programar sobre una MEX S-Function s'ha de conèixer el llenguatge de programació del compilador objectiu, en aquest cas el compilador de la DS1104 posseeix algunes funcions de comunicació amb les entrades/sortides digitals que ens són de gran utilitat i que s'expliquen a continuació [11].

```
DS1104_bit_io_init(DS1104_DIO0_config0| .... | DS1104_DIO19_config19);
```

Amb aquesta ordre s'inicialitzen les entrades/sortides digitals. Per tal d'evitar alarmes i/o errors s'aconsella inicialitzar les vint (DIO0 ... DIO19) que disposa la DSP, per cada bit s'ha de configurar (config0 .... config19) com a entrada (IN) o sortida (OUT) tots ells independents entre sí.

Exemple:

```
ds1104_bit_io_init( DS1104_DIO0_OUT | DS1104_DIO1_OUT |  
  
    DS1104_DIO2_OUT | DS1104_DIO3_OUT |  
  
    DS1104_DIO4_OUT | DS1104_DIO5_OUT |  
  
    DS1104_DIO6_OUT | DS1104_DIO7_OUT |  
  
    DS1104_DIO8_OUT | DS1104_DIO9_OUT |  
  
    DS1104_DIO10_IN | DS1104_DIO11_IN |  
  
    DS1104_DIO12_IN | DS1104_DIO13_IN |  
  
    DS1104_DIO14_IN | DS1104_DIO15_IN |  
  
    DS1104_DIO16_IN | DS1104_DIO17_IN |  
  
    DS1104_DIO18_IN | DS1104_DIO19_IN );
```

```
DS1104_bit_io_write(output)
```

Escriu als bits de sortida el nombre enter escrit a l'argument output, que després es traduirà a binari i s'escriurà als bits de sortida.

Exemple: DS1104\_bit\_io\_write(53);

Al nombre de l'exemple correspon al 110101 en binari per tant establirà els bits IO0, IO2, IO4 i IO5 a 5V i IO1 i IO3 a terra.

En l'antecedent la senyal de sortida va processar per mitjà de la sortida PWM (CP18), en l'actual projecte, en canvi, s'opta per fer-ho per les sortides digitals. D'aquesta manera, juntament amb la reducció del nombre de blocs mitjançant la traducció a C, serà el mètode pel qual es redueixi al màxim que permet el hardware el temps de resposta i per tant el de mostreig.

Per altra banda, les entrades analògiques, tant les independents com les multiplexades, s'introdueixen al programa directament des dels blocs de la llibreria de la DS1104.

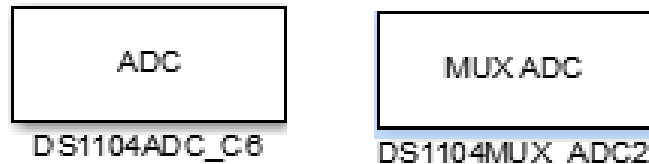


Fig. 2.8. Blocs d'entrada analògica independent i multiplexada.

Una vegada acabat el programa en Matlab-Simulink s'ha de transformar a la extensió .sdf per tal de poder tractar-lo amb DSpace, aquesta extensió es podrà trobar una vegada es compili el programa a Simulink a la carpeta directori de Matlab, amb el mateix nom però amb la nova extensió esmentada.

En principi un dels objectius a assolir va ser aconseguir fer tot el programa mitjançant un sol bloc S-Function programat en C per tal de reduir al màxim el temps de resposta, però al fer aquest pas necessari va sorgir un problema de compatibilitat entre els dos programes. El problema és que DSpace no considera les variables que provenen de dintre d'una S-Function de Simulink, només es poden visualitzar les variables entrants i sortints dels blocs, per aquest motiu es va descartar l'objectiu d'aconseguir programar tot en un sol bloc, prioritzant la finalitat del projecte de sincronització a xarxa i millora de senyal sobre el mitjà per dur a terme la reducció del temps de mostreig de la traducció en C, tot i que si que s'ha utilitzat, en gran part, la programació en C.



### 3. Procés experimental.

Abans de començar a fer cap canvi el primer que es fa és posar a punt l'equip i el muntatge de la mateixa manera que va quedar un cop va finalitzar el projecte anterior, comprenent els següents passos.

1. Instal·lació de Software: Matlab-Simulink i DSpace.

Matlab-simulink s'utilitza com a plataforma per a programar mentre que DSpace és el programa que l'utilitza per la comunicació amb la targeta DS1104, per al controlar i visualitzar el comportament del sistema en temps real.

2. Muntatge del circuit amb l'esquema mostrat a continuació:

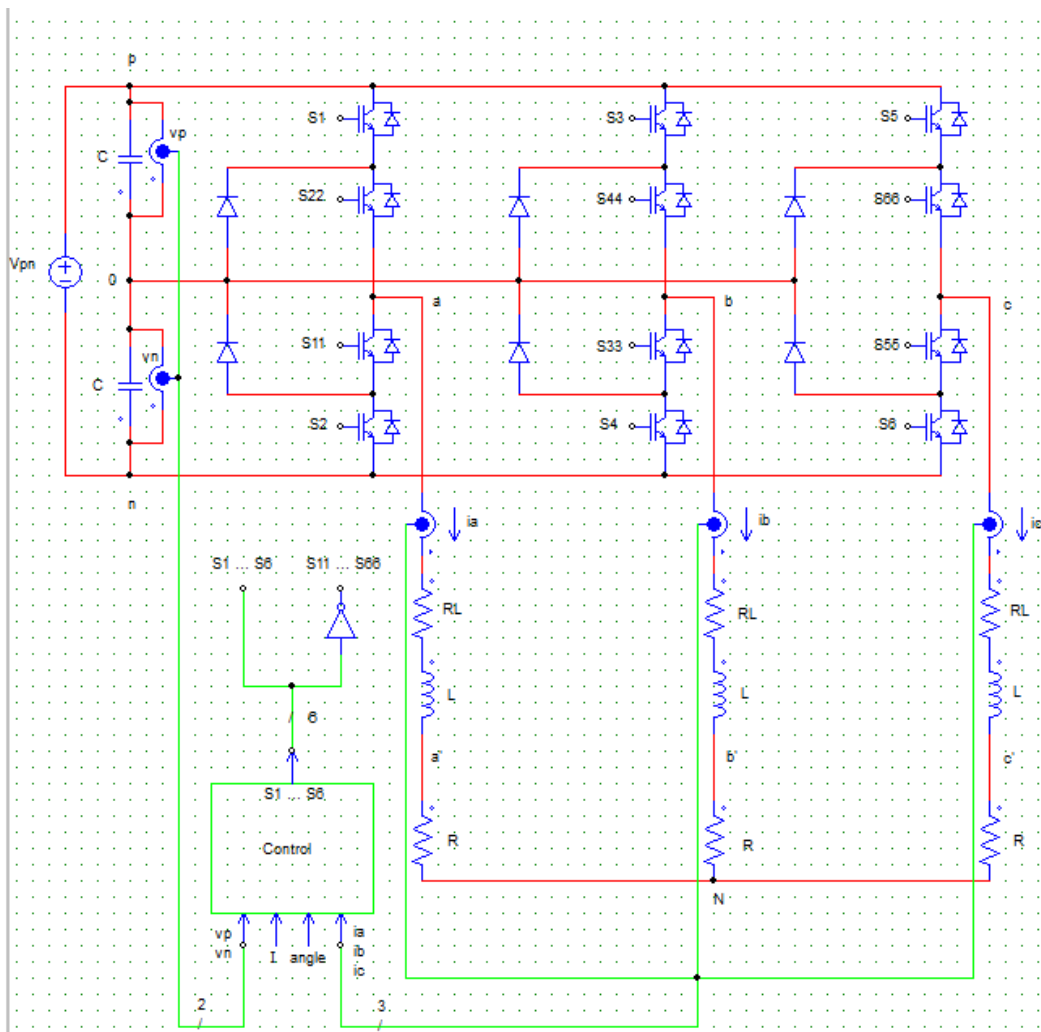


Fig. 3.1. Esquema electrònic del model amb càrrega RL.

3. Comprovació de funcionament: transformació del programa Simulink per tractar-ho amb DSpace i comprovació del funcionament. Per seguretat es verifica primer per  $V_{pn}=30V$  i després per  $V_{pn}=100V$ .

a. Equilibri de les tensions  $v_p$  i  $v_n$ .

Per al primer experiment  $v_p$  i  $v_n$  han de ser  $+15V_{cc}$  i  $-15V_{cc}$  respectivament, pel segon  $+50V_{cc}$  i  $-50V_{cc}$ .

b. Corrents  $i_a$ ,  $i_b$  i  $i_c$  formen un corrent sinusoidal trifàsic.

Amb aquesta comprovació es verifica que el control predictiu està actuant de forma correcta i segueix el corrent sinusoidal de referència.

c. Les consignes  $I$  i angle modifiquen l'amplitud i el desfasament de  $i_{abc}$ .

Al DSpace a part de fer displays per a la visualització de corrents, tensions, temps, etc. també es fan dues finestres d'entrada numèrica per a entrar el desfasament entre tensió i corrent i l'amplitud de corrent, així com botons per a controlar el reset i el trip. A continuació es mostra una captura d'imatge amb un panell general del programa DSpace.

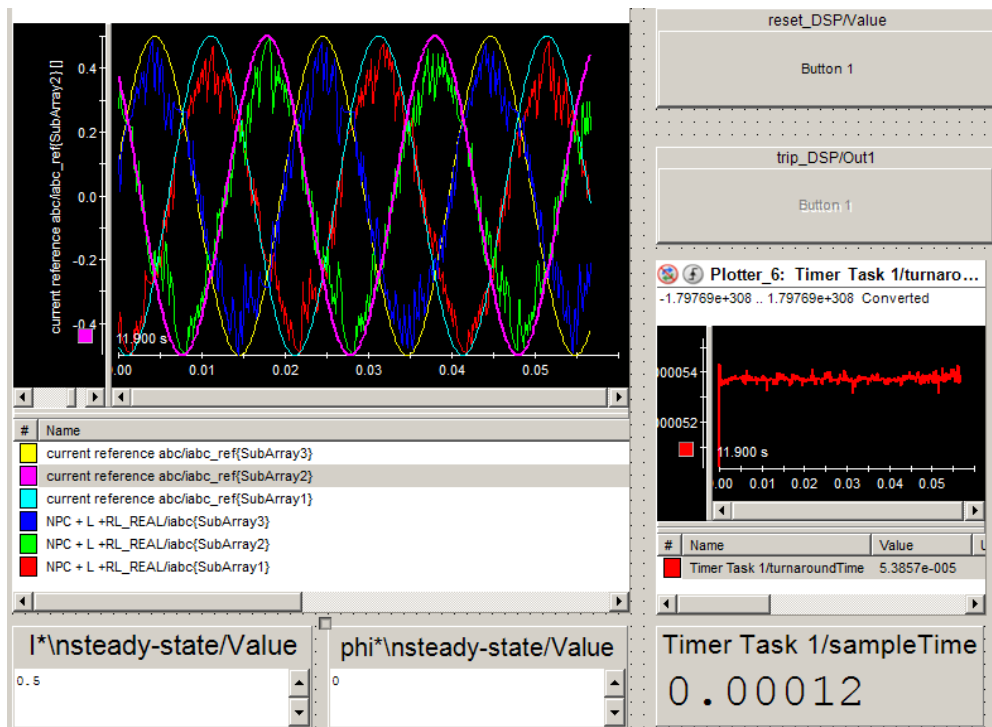


Fig. 3.2. Vista general exemple control DSpace.

4. A partir d'aquest model es comencen a fer innovacions a nivell de programa en la seva majoria per tal de millorar el rendiment de la targeta DS1104, reduint el temps de resposta i així millorar el senyal trifàsic  $i_{abc}$ .

Les mesures que es prenen per a assolir aquest objectiu són:

- a. Recalibració dels sensors.

En la recalibració de sensors s'ajusten les entrades analògiques amb una correcció optimitzada a una recta. L'esquema de blocs és el següent:

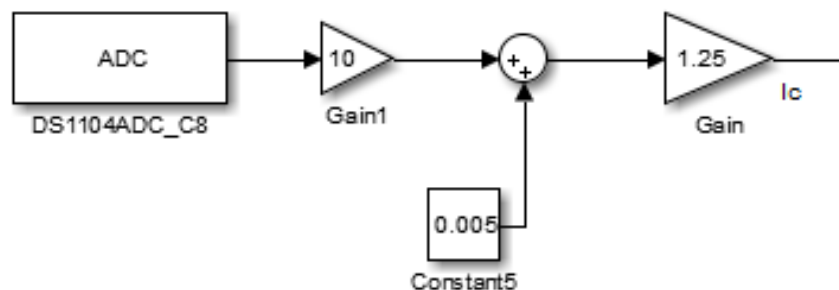


Fig. 3.3. Esquema per la correcció d'error en una senyal.

Per a conèixer l'error es estàtic es fa la mesura (en exemple de la corrent amb circuit obert, i en tensió en curtcircuit) de forma que doni igual a zero i es suma l'error amb una constant. Per a conèixer l'error proporcional s'aplica un corrent o tensió coneguts i es corregeix mitjançant un guany.

- b. Substitució de sortides del port PWM per sortides digitals.

Aquest canvi es fa a nivell de software i hardware. Es substitueixen els blocs d'escriptura per PWM per l'escriptura sobre les sortides digitals directament en codi de la S-Function de control predictiu. A nivell de hardware es suprimeix la sortida PWM i es fan soldadures a les sortides digitals per tal de fer una connexió robusta.

- c. Simplificació del model, integrant al màxim els blocs en una S-Function.

Es suprimeixen els blocs necessaris per a dur a terme la mesura anterior, fent la seva funció de la mateixa manera en la funció de subsistema .

El resultat de les mesures anteriors van provocar una reducció del temps de mostreig gairebé del 50% (de 150 $\mu$ s original a 80 $\mu$ s) i en conseqüència una notable millora de la senyal.

El programa resultant, així com tots els altres que es fan al llarg del projecte, es troben al CD adjunt amb la memòria, als annexos es pot trobar el codi de programació del control predictiu i un anàlisi més ampli sobre els resultats en l'apartat de resultats experimentals.

Una vegada assolit aquest primer objectiu es va continuar amb la sincronització amb xarxa.

Al haver-hi grans canvis a nivell de hardware i software, en aquest tram del projecte pren especial rellevància la seguretat.

En el model anterior, al no incorporar encara la xarxa no hi ha gaire perill de malmesa de components o dispositius. Com a mesura de seguretat és suficient el limitador de corrent de la font de tensió que alimenta el circuit ( $V_{pn}$ ).

En el pròxim model en canvi, calen unes mesures de seguretat més rigoroses. Primerament, abans de provar el model directament sobre el model físic es fa un model de simulació, comprovant el funcionament amb components simulats i una senyal trifàsica de referència també simulada.

En aquesta simulació encara s'accentua més la optimització del control predictiu, amb una programació molt més compacta i amb millor rendiment, i també s'introdueix el PLL de realimentació de l'angle de desfasament de  $v_{abc}$ .

Una vegada s'obtenen resultats favorables es procedeix amb el nou muntatge, l'esquema electrònic es pot veure a continuació:



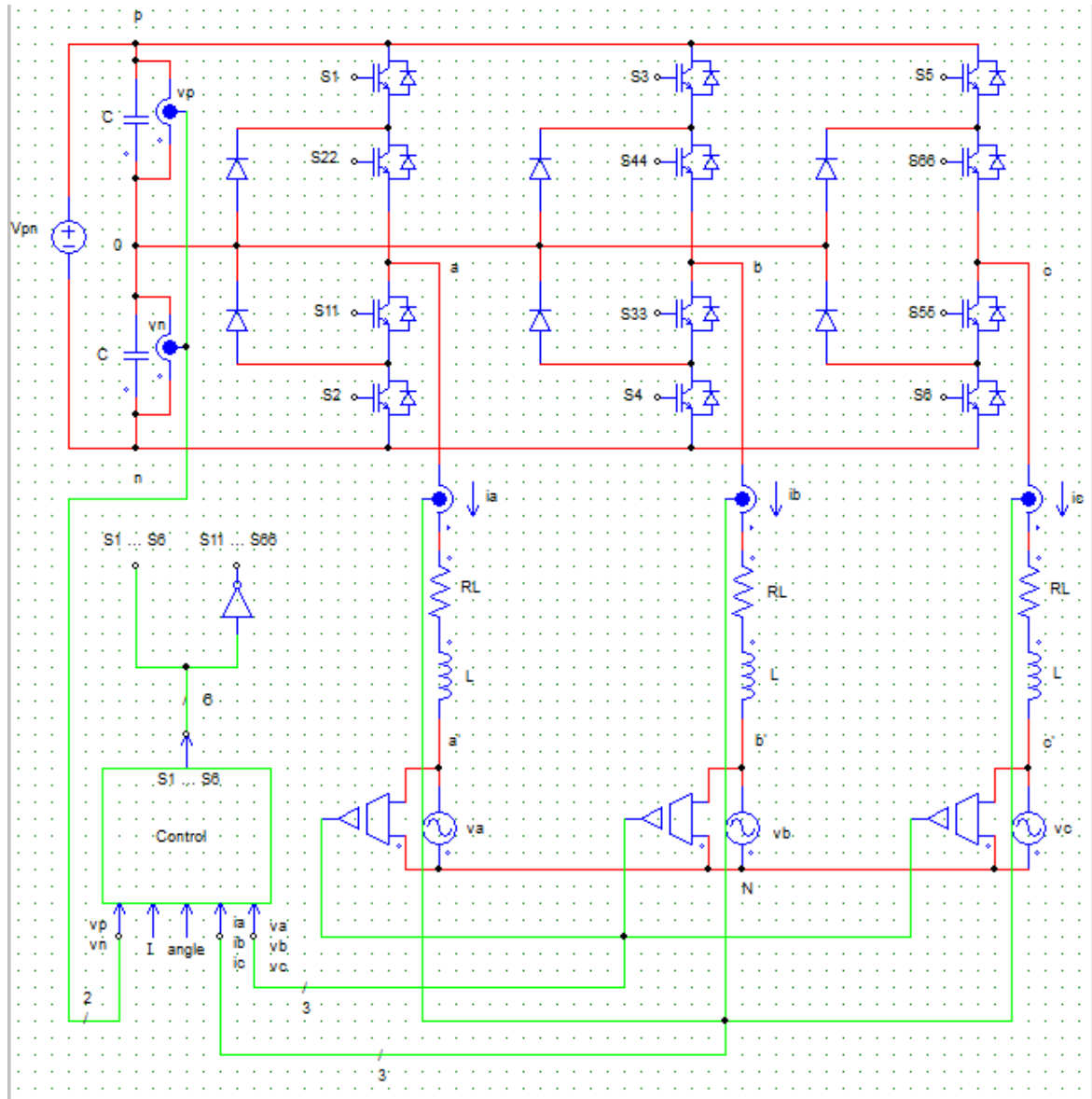


Fig. 3.4. Esquema electrònic del model amb sincronització a xarxa.

En el projectes experimentals i de programació, on la metodologia és la coneguda com a prova i error, la seguretat és un aspecte clau que permet no fer malbé el dispositius o components i per tant no tenir grans desviaments tant en el pressupost com en la planificació i terminis, per aquest motiu es fa especial èmfasi sobre la seguretat en el mateix procediment experimental.

La xarxa se sotmet, per evitar possibles curtcircuits i les seves possibles conseqüències, a un aïllament elèctric mitjançant un transformador trifàsic connectat amb la configuració Y-Y i un regulador de tensió en sèrie tal i com es mostra en la il·lustració següent:

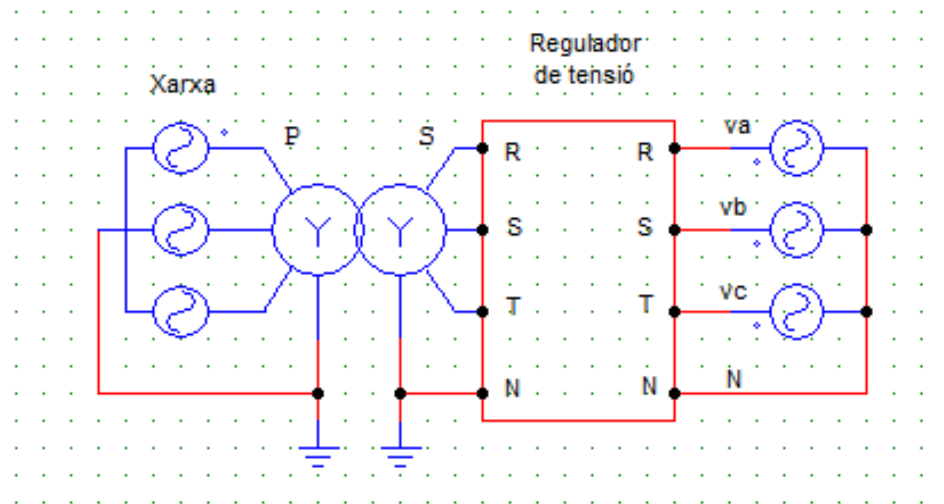


Fig. 3.5. Esquema electrònic de seguretat de xarxa.

El motiu pel qual s'ha fet la configuració Y-Y del transformador és que el sistema està pensat per mesurar tensions fase-neutre, per no haver de tenir en compte la diferència entre fase-fase i fase-neutre la banda baixa ha de tenir la configuració estrella. La banda alta s'estableix també en estrella per tenir correspondència entre les columnes del transformador i esdevenir un grup horari zero (Yy0).

Respecte el model anterior es fan les següents modificacions:

- a. Substitució les tres resistències col·locades en estrella per la xarxa trifàsica.  
En el pla del hardware, per tal d'utilitzar el sistema com a generador a la xarxa elèctrica es substitueixen les resistències que feien de càrrega per la xarxa, on s'ha d'aplicar el corrent.
- b. S'afegeixen tres sensors de tensió LEM LV 25-P, es recalibren i es reordenen.  
Els sensors corresponents a les tensions  $v_a$ ,  $v_b$  i  $v_c$  s'afegeixen en hardware i es connecten a la DS1104 amb cables BNC, s'aplica la mateixa correcció feta amb els altres sensors del model anterior. A l'apartat de connexions BNC s'ha explicat que

hi ha entrades independents i multiplexades, les més ràpides ( $i_a$ ,  $i_b$ ,  $i_c$ ) han de col·locar-se amb les entrades analògiques independents (C5-CP8) mentre que les més estables ( $v_p$ ,  $v_n$ ) a les entrades multiplexades (CP1-CP4).

Per tant, la configuració òptima de les entrades analògiques és:

Connector	Senyal	Connector	Senyal
CP1	vp	CP5	vc
CP2	vn	CP6	ia
CP3	va	CP7	ib
CP4	vb	CP8	ic

Taula 3.1. Optimització d'entrades analògiques.

- c. Modificació íntegra de la S-Function, tenint en compte les noves variables ( $v_{abc}$ ), més compacta i amb millor rendiment.  
Es fa una nova versió del control predictiu, amb moltes menys línies de codi tot i tenir majors prestacions que en el primer model.
- d. Incorporació de llaç tancat per la sincronització (PLL).  
Al programa de Simulink s'incorporen els blocs pertanyents a la PLL, per a la sincronització a la xarxa, on s'extreu l'angle a partir de les tensions  $v_a$ ,  $v_b$  i  $v_c$ .
- e. Aïllament elèctric de la xarxa.  
S'utilitza un transformador i un regulador de tensió com a mesura de seguretat tal i com s'ha explicat amb anterioritat.

Els programes de la simulació i els dos models dissenyats es poden trobar al document Annexos del projecte o també en format digital al CD-ROM adjunt.



## 4. Resultats experimentals.

A continuació s'analitzen els resultats dels dos experiments que s'han fet en aquest projecte, convertidor amb càrrega RL i convertidor amb sincronització a la xarxa.

Les gràfiques que es mostren a continuació no tenen unitat als eixos, el programa DSpace no les dóna i les gràfiques són extretes captures de pantalla d'aquest, les unitats dels eixos d'abscisses són sempre en segons i les ordenades es diuen a la descripció anterior o posterior a cada gràfica.

### 4.1. Convertidor amb càrrega RL.

Per model amb càrrega RL es fan dos experiments, el primer té com a  $V_{pn}=30V$  i el segon  $V_{pn}=100V$ .

La primera dada a mirar és el temps de resposta (o temps de càlcul) del programa, aquest primer model presenta un temps de mostreig que no supera en cap moment els  $55\mu s$ . El temps de càlcul permet acotar el temps de mostreig, que sempre ha de ser major a aquest, es proposa un marge de  $25\mu s$ , per tant s'estableix el temps de mostreig en  $80\mu s$ .

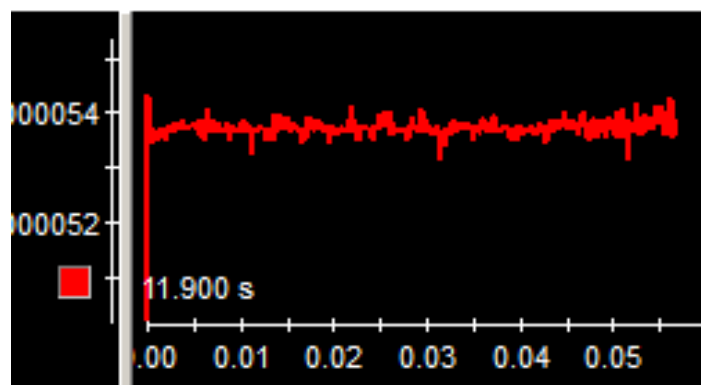


Fig. 4.1. Temps de resposta del model amb càrrega RL.

Una altra dada fonamental a tenir en compte és l'equilibri de les tensions  $v_p$  i  $v_n$ , per tal que el convertidor funcioni adequadament el control predictiu estableix el punt neutre i fa que aquestes tensions siguin simètriques respecte el zero.

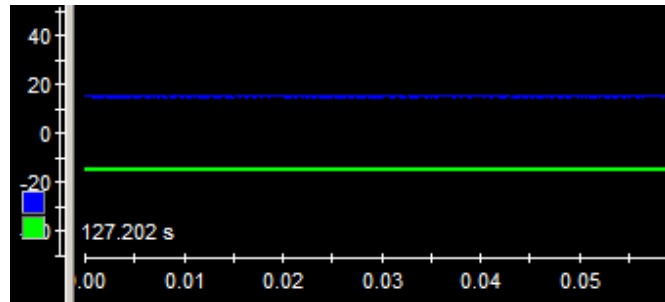


Fig. 4.2. Tensions  $v_p$  i  $v_n$  per  $v_{pn} = 30V$  del model amb càrrega RL.

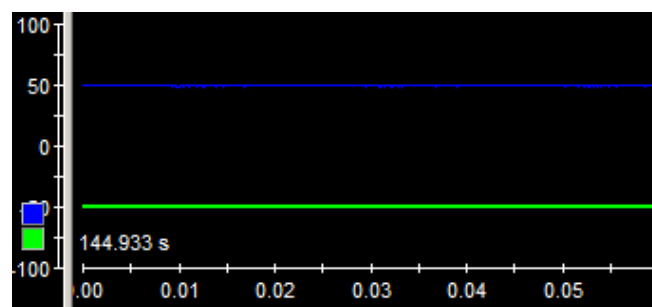


Fig. 4.3. Tensions  $v_p$  i  $v_n$  per  $v_{pn} = 100V$  del model amb càrrega RL.

Per tots dos casos es comprova el correcte funcionament dels condensadors C1 i C2 i també la part corresponent del control predictiu: per  $v_{pn} = 30V$ :  $v_p = +15V$  i  $v_n = -15V$ , per  $v_{pn} = 100V$ :  $v_p = +50V$  i  $v_n = -50V$ .

A continuació es procedeix a mirar si el model predictiu funciona de manera adequada, per fer-ho es verifica el seguiment del senyal de referència per les dues tensions.

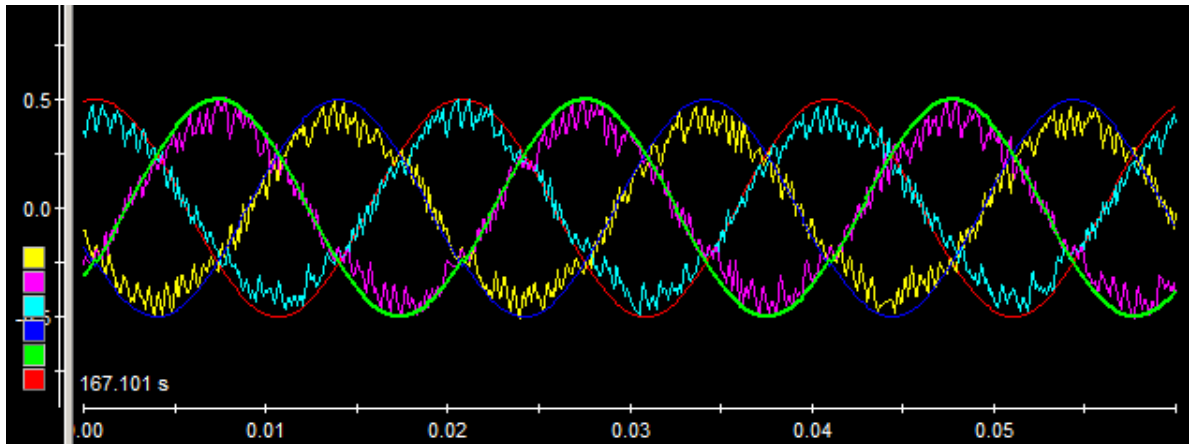


Fig. 4.4. Corrents per  $v_{pn}$  30V, I 0.5A,  $\Theta$  0rad.

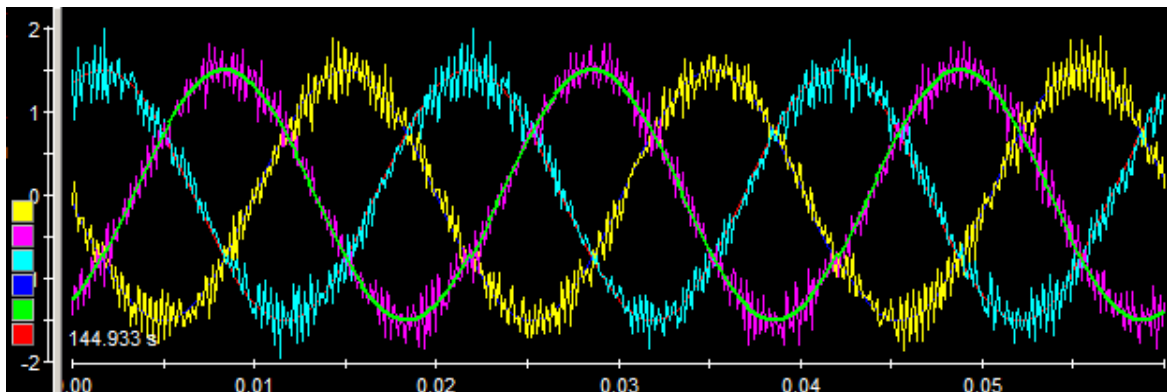


Fig. 4.5. Corrents per  $v_{pn}$  100V, I 1.5A,  $\Theta$  0rad.

En les dues gràfiques temporals anteriors es pot apreciar que per a major tensió i corrent el model funciona millor, aquesta diferència pot generar-se per diferents motius, entre ells la caiguda de tensió en les IGBTs o la poca linealitat de les bobines, considerades en el control predictiu com ideals.

Finalment es fa la última prova concloent del model amb càrrega RL presentant referències no transitòries de pujada i de baixada, per  $V_{pn}=100V$ .

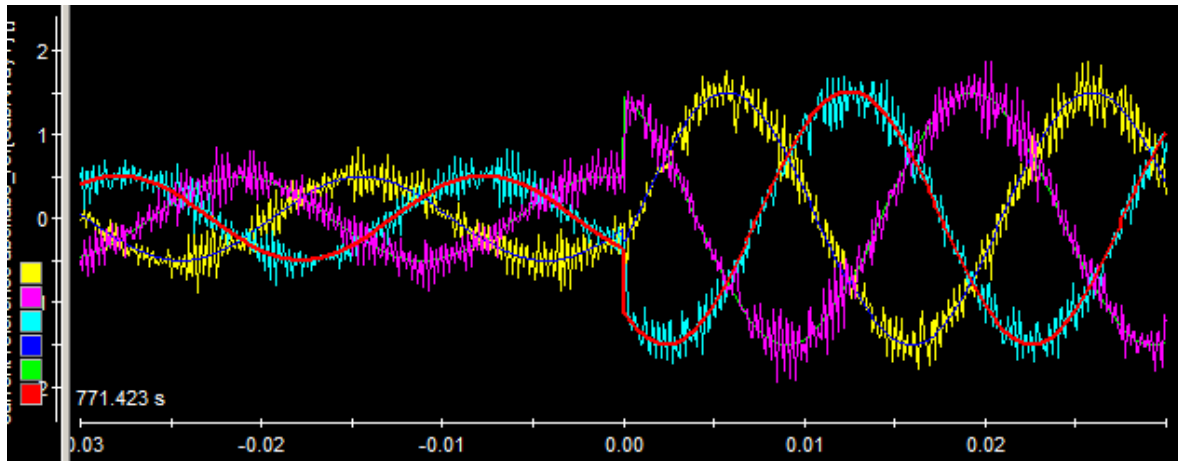


Fig. 4.6. Resposta del model RL per flanc de pujada de corrent.

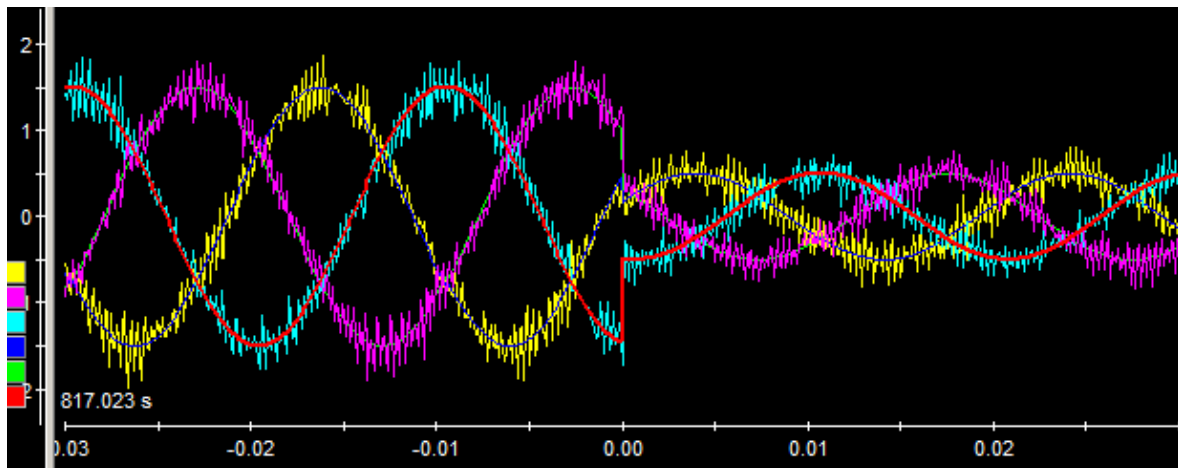


Fig. 4.7. Resposta del model RL per flanc de baixada de corrent.

Amb aquesta última prova constata que el sistema funciona molt satisfactòriament en la seva totalitat, tant el programa dissenyat com el model i els seus components.



## 4.2. Convertidor amb sincronització a la xarxa.

Per al model amb sincronització a la xarxa es procedeix a fer comprovacions similars abans de mirar el resultat final.

En aquest model es preveu un increment notable del temps de resposta, donades les noves funcions afegides al sistema.

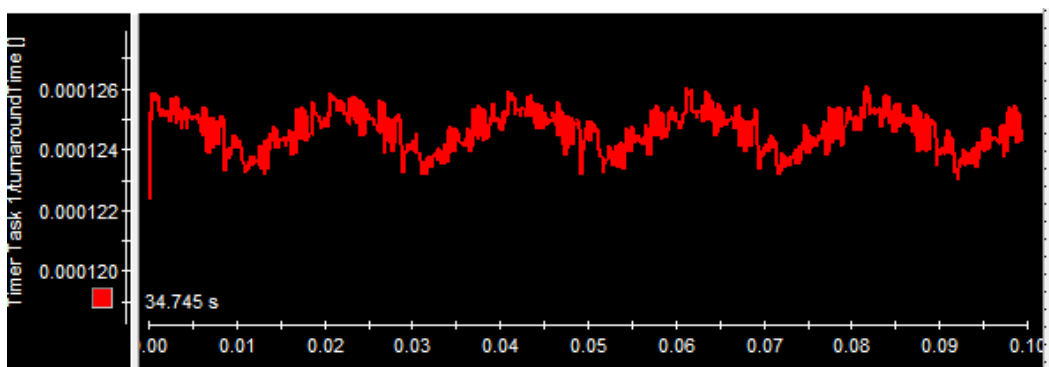


Fig. 4.8. Temps de resposta del model amb sincronització a xarxa.

En aquest últim experiment s'obté un temps de resposta per sota de  $126\mu\text{s}$ , afegint els  $25\mu\text{s}$  de marge que es van utilitzar al primer model i funcionaren correctament s'obté el nou temps de mostreig de  $150\mu\text{s}$ .

L'experiment es duu a terme per  $V_{pn}=100\text{V}$ , en la següent gràfica es mostra la simetria de  $v_p$  i  $v_n$ , en volts.

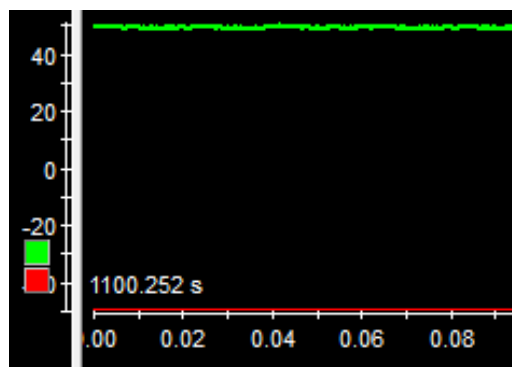


Fig. 4.9. Tensions  $v_p$  i  $v_n$  del model amb sincronització a xarxa.

Es poden observar els valors de  $v_p$  i  $v_n$  són +50V i -50V respectivament.

També es comprova si l'angle de fase que proporciona el PLL es correspon amb la lectura de les tensions de la xarxa.

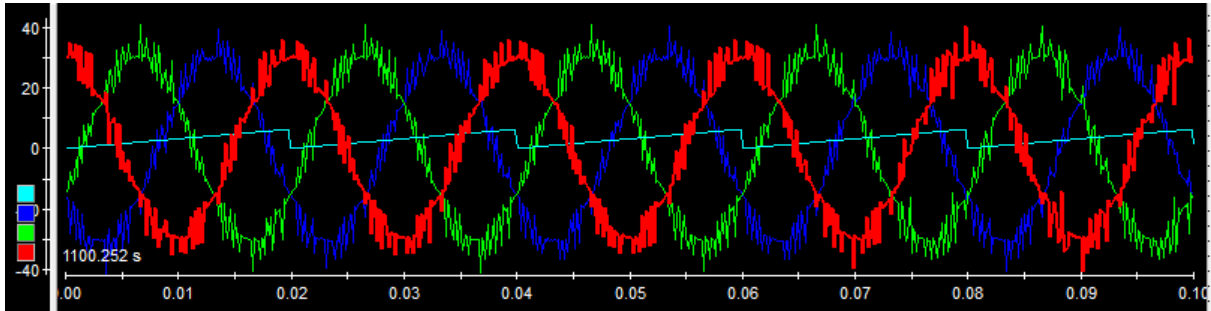


Fig. 4.10. Tensions  $v_{abc}$  i angle de fase del model amb sincronització a xarxa.

En aquest diagrama es pot veure l'angle (en radians i de color cian) sincronitzat amb la ona sinusoidal vermella, corresponent a la tensió  $v_a$  (en volts).

Així mateix es pot veure com la ona no és sinusoidal perfecta, les commutacions es reflecteixen com a soroll i fan que la forma no sigui neta.

En el mateix període de temps es fa una captura dels corrents  $i_{abc}$ , les consignes de l'experiment són  $I=1.5A$  i  $\Theta=0$ .

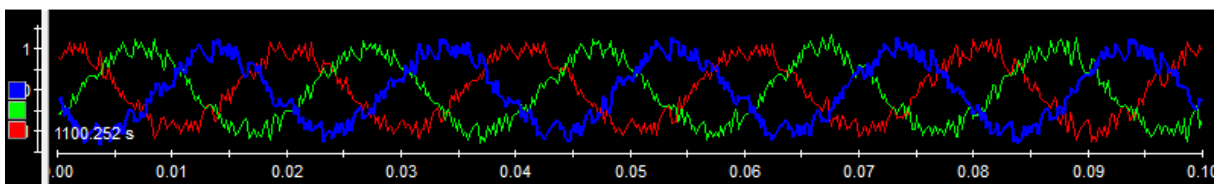


Fig. 4.11. Corrents  $i_{abc}$  del model amb sincronització a xarxa.

Es pot comprovar la bona relació de tensió i corrent però també l'error en règim permanent que ja estava present al model anterior. En les conclusions s'analitzen els possibles motius d'aquesta diferència entre la consigna i el corrent real.

Per últim es verifica el comportament de l'angle consigna, que determina el desfasament entre tensió i corrent.

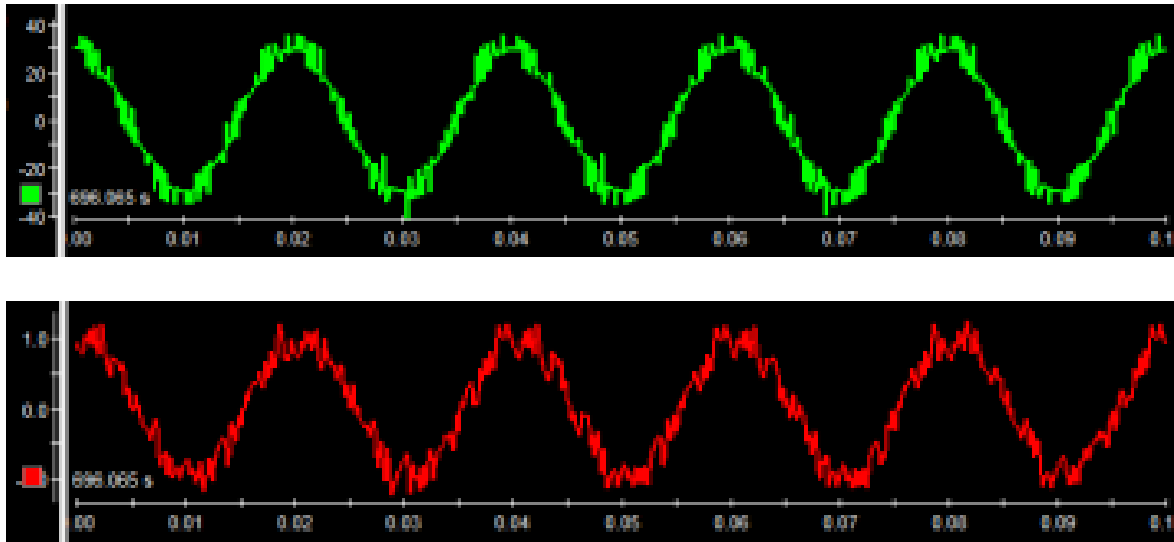


Fig. 4.12. Tensió  $v_a$  i corrent  $i_a$  per  $\Theta$  nul·la.

En aquests diagrames s'observa que no hi ha cap tipus de desfasament entre tensió i corrent, en aquesta situació la potència activa i aparent són iguals i en conseqüència el factor de potència  $\text{fdp}=1$ .

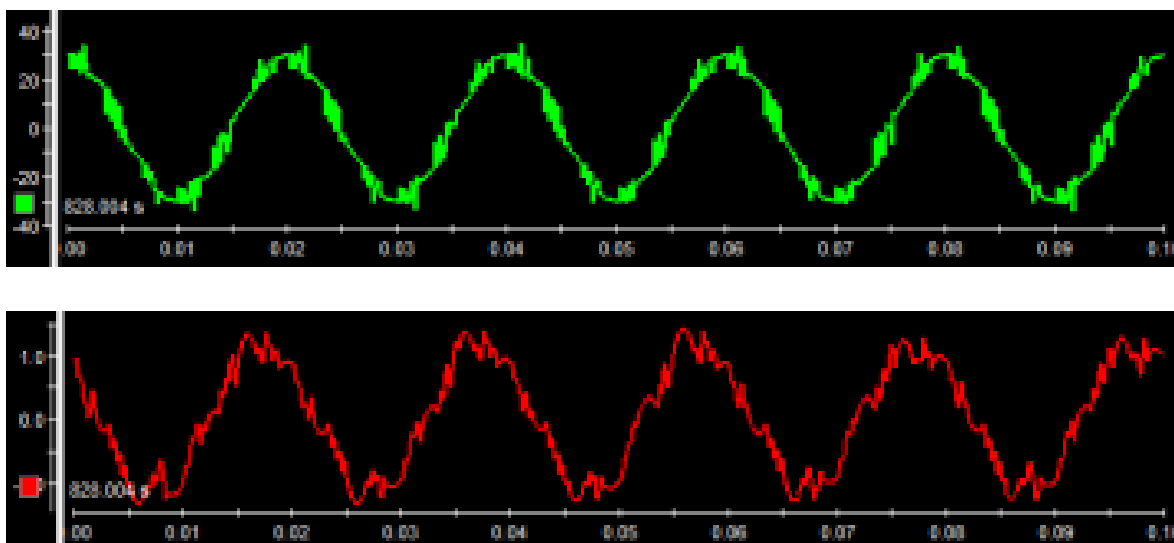


Fig. 4.13. Tensió  $v_a$  i corrent  $i_a$  per  $\Theta = +1\text{rad}$ .

Al imposar un angle  $\Theta$  positiu el corrent s'avança a la tensió, per tant en aquesta situació es genera corrent capacitiu cap a la xarxa.

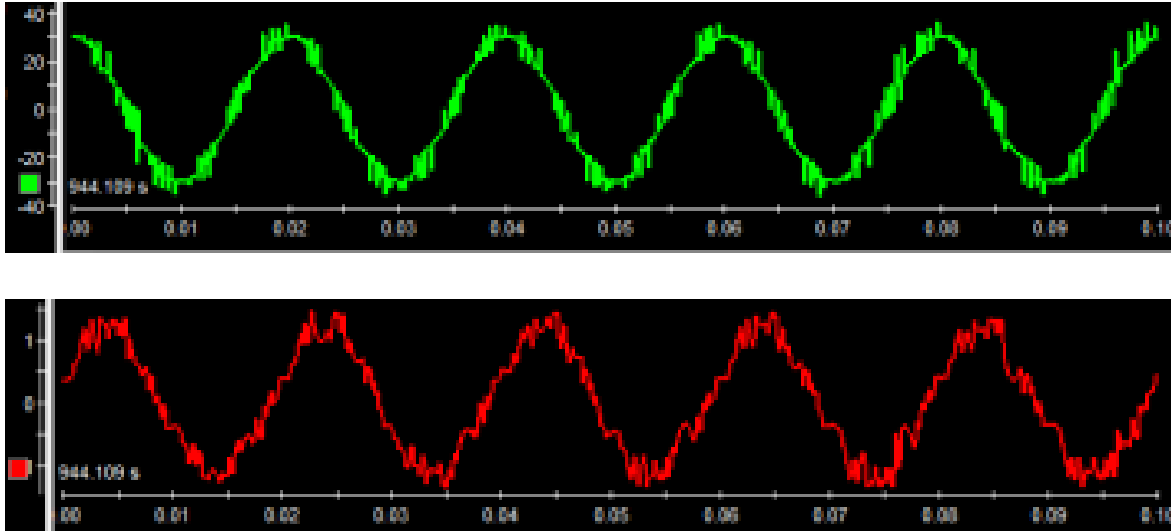


Fig. 4.14. Tensió  $v_a$  i corrent  $i_a$  per  $\Theta = -1\text{rad}$ .

En el cas contrari, si s'estableix  $\Theta$  negatiu, el corrent queda retardat respecte la tensió i es genera corrent inductiu.

## 5. Calendari del projecte.

En aquest apartat es mostra el calendari real que s'ha seguit al llarg del projecte, amb les tasques clau i les dates d'entrega, un resultat diferent de la planificació donada a l'avantprojecte, amb retards de temps que han ocasionat la pròrroga de la presentació.

Nom de la tasca	Duració	Començament	Finalització	Recursos utilitzats
Avantprojecte	60 hores	3/11/2014	15/01/2015	Ordinador, Office
Entrega avantprojecte	-	16/01/2015	16/01/2015	
Preparació de l'equip	14 hores	03/02/2015	16/02/2015	Ordinador
Aprenentatge llenguatge C	18 hores	20/01/2015	02/02/2015	
Entesa projecte antecedent	33 hores	03/02/2015	23/02/2015	
Posta en marxa antecedent	37 hores	24/02/2015	09/03/2015	Bancada de treball
Entesa comunicació DSP	18 hores	02/03/2015	06/03/2015	
PWM per sortides digitals	70 hores	09/03/2015	24/04/2015	Bancada de treball
Reunió amb Alex Calle	3 hores	13/04/2015	13/04/2015	
Simplificació programa	39 hores	27/04/2015	01/05/2015	Bancada de treball
Experimentació circuit RL	19 hores	04/05/2015	08/05/2015	Bancada de treball
Remuntatge sistema	24 hores	11/05/2015	15/05/2015	Bancada de treball
Programació simulació	35 hores	18/05/2015	02/06/2015	Matlab-Simulink
Provatures de la simulació	19 hores	03/06/2015	09/06/2015	Bancada de treball
Programació xarxa real	28 hores	10/06/2015	18/06/2015	Bancada de treball
Muntatge per sincronització	4 hores	19/06/2015	19/06/2015	Bancada de treball
Optimització de codi	41 hores	22/06/2015	29/06/2015	Matlab-Simulink
Experimentació sincronitz.	18 hores	08/07/2015	09/07/2015	Bancada de treball
Redacció de documentació	120 hores	25/05/2015	25/09/2015	Ordinador, Office
Entrega de documentació	-	30/09/2015	30/09/2015	

Taula 5.1. Calendari del projecte.

La primera tasca destacable que es va fer un cop entregat l'avantprojecte va ser la preparació de l'equip, per a treballar tant a la universitat amb la bancada de treball, com des de casa per les simulacions, l'ordinador de la universitat es va preparar amb el software necessari: Matlab, Simulink i DSpace. També es va ampliar la memòria RAM de 2GB a 4GB per accelerar els processos més lents.

La primera tasca pràctica que es va dur a terme un cop assolida la posta en marxa del sistema de forma que funcionés de la mateixa manera que l'últim dia de l'anterior projecte va ser la substitució de les sortides pel port PWM per sortides digitals, aquesta és la segona tasca més duradora després de la redacció de documentació, el motiu és que tot i ser la primera era força complicada, per fer-la s'havien de tenir molt clar tots i cada un els conceptes estudiats prèviament i després de molts intents sense èxit es va haver de recórrer a un expert en la matèria per veure quins eren els errors que s'estaven cometent.

La reunió amb Alex Calle va ser un punt d'inflexió en aquest projecte, a partir d'aquest dia s'assoliren objectius cada una o dos setmanes com a màxim i no es va arribar més al punt de no saber què més fer per a fer funcionar el programa de la manera que es volia.

Cal destacar també els esdeveniments no previsibles que van suposar una pèrdua de temps força significativa, la primera va ser a finals de maig, al assistir al laboratori amb el tutor per una reunió de seguiment ens vam adonar que algú havia tocat el circuit, provocant que el circuit no funcionés i havent d'inspeccionar i tornar a muntar el circuit en busca de la falla. La segona circumstància imprevisible va ser al mes de juny, l'ordinador de la universitat, que feia dies que es reiniciava de forma aleatòria, va deixar de funcionar, el responsable tècnic Antonio Ortel va haver d'instal·lar de nou el disc dur i es va haver de preparar l'equip de nou.

Aquests dos incidents provocaren una pèrdua de temps productiu d'aproximadament dues setmanes.

En la última reunió amb el tutor Salvador Alepuz del mes de juliol es varen assolir tots els objectius que es proposaren a l'inici, deixant un ampli marge de temps per a la redacció del projecte.

En la pàgina següent es mostra una il·lustració amb la dispersió temporal de les tasques descrites a la taula anterior.

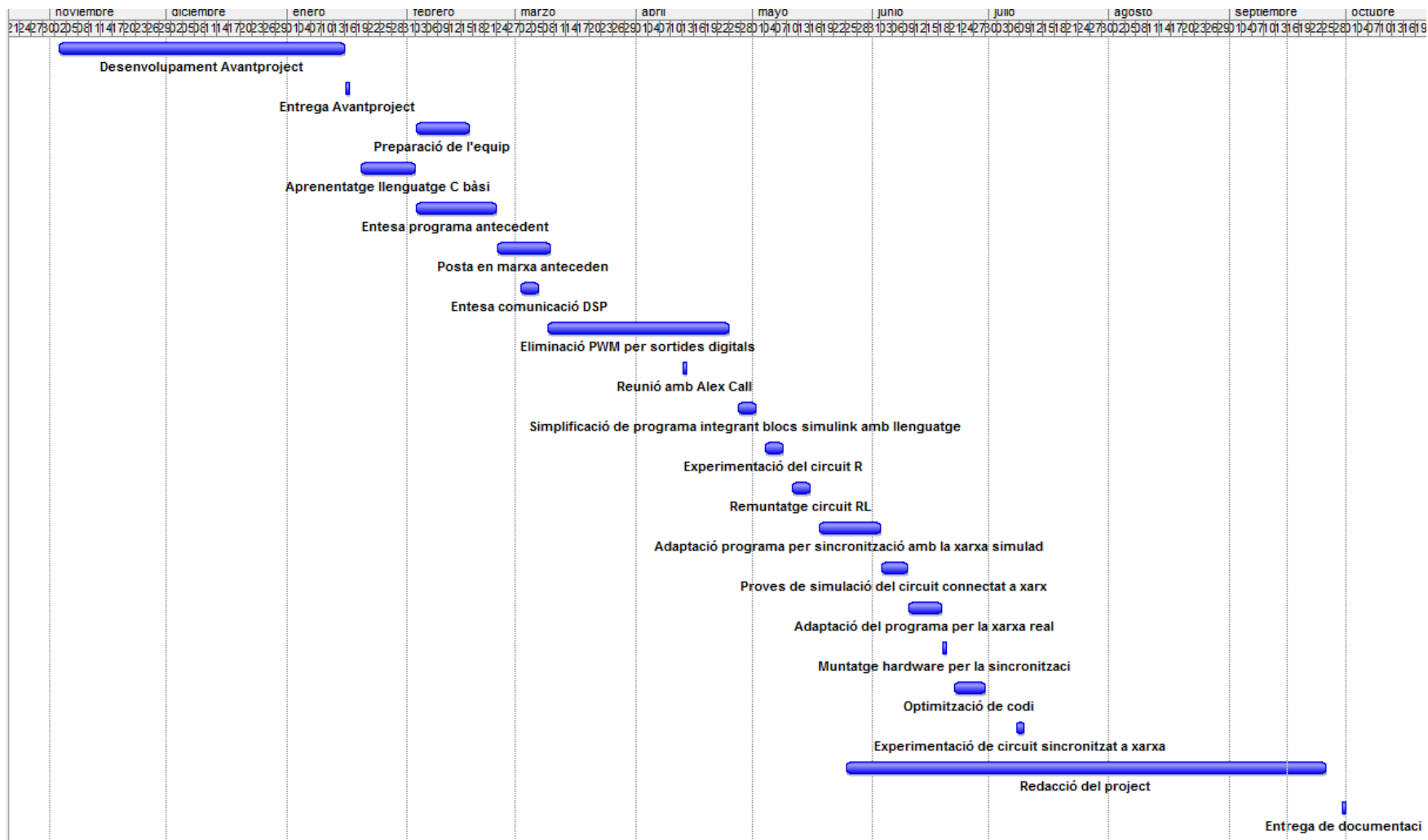


Fig. 5.1. Gràfica temporal del calendari del projecte.





## **6. Impacte mediambiental.**

Aquest projecte no té l'objectiu de dissenyar un producte per a produir en sèrie i vendre, sinó que és un projecte de R+D que es basa en un control predictiu per a un convertidor en sincronització a xarxa i és part d'un sistema de generació d'energia renovable, amb aquesta premissa el projecte no té en cap cas impacte mediambiental negatiu fora de la energia elèctrica que es consumeix al llarg de la experimentació i de l'activitat normal de treball.

Pel que fa al hardware ha estat tot amb material reutilitzat i l'equip que s'ha utilitzat han estat recursos disponibles de la universitat EUPMT.

L'únic recurs que s'ha hagut de substituir per un de nou ha estat un disc dur que va espatllar-se, es va deixar el disc dur defectuós en mans de la universitat, per poder arreglar-lo o sinó deixar-ho com a deixalla electrònica, que més tard la universitat s'encarrega de portar a una planta de reciclatge.



## 7. Conclusions.

Al finalitzar el projecte es pot afirmar que s'han assolit els objectius que s'han proposat a l'avantprojecte, les fites que s'han assolit al llarg de projecte són:

Posta en marxa del model de control predictiu amb càrrega RL fet amb anterioritat al projecte precedent.

Reducció del temps de càlcul i temps de mostreig del sistema mitjançant la integració de programa en el bloc de funció de subsistema (S-Function) en llenguatge C i la programació de les sortides mitjançant els pins d'entrada i sortida digitals en comptes dels ports de modulació de polsos (PWM). Aconseguint una reducció de temps de 70 $\mu$ s sobre el model amb càrrega RL, reduint fins a 80 $\mu$ s els 150 $\mu$ s inicials.

Disseny del nou model amb connexió a xarxa, en forma de simulació i posteriorment de forma experimental.

Sincronització a la xarxa elèctrica mitjançant un llaç tancat PLL i el control predictiu a dos passos, aconseguint també una reducció de codi de control.

Control manual de l'amplitud i desfasament del corrent respecte la tensió, per aconseguir que el sistema actuï de forma capacitiva o inductiva segons conveniència.

En l'últim model el sistema augmenta el temps de resposta a 126 $\mu$ s fet que fa augmentar el temps de mostreig a 150 $\mu$ s. El motiu d'aquesta diferència entre els models es deu a les noves prestacions del programa, mentre que el primer utilitza cinc sensors els quals s'introdueixen al control per les entrades analògiques i totes elles actuant com a independents, el segon model necessita utilitzar les vuit entrades analògiques que disposa la DS1104, sent quatre independents i quatre multiplexades, aquestes noves funcions disparen el temps de resposta i fan augmentar per tant el temps de mostreig.

En els resultats experimentals s'ha pogut apreciar un error en règim permanent de l'amplitud del corrent, aquests errors són provocats per imprecisions del model: linealitat de les bobines, soroll provinent de commutacions o imprecisió dels sensors tot i que han estat recalibrats.

Les línies d'investigació futura que poden derivar d'aquest projecte de cara a treballs futurs són:

- Esbrinar amb precisió la font d'aquest error en règim permanent i trobar la solució.
- Reducció al màxim del temps de mostreig del model amb sincronització a xarxa mitjançant la inclusió integral dels blocs en llenguatge C al bloc S-Function.
- Millora de hardware com a mesura per a la millora de senyal, una nova DSP amb millors prestacions o altres components amb major velocitat de resposta.
- Solucionar el mòdul de Enable, Reset i Trip, que no funciona correctament.

## 8. Referències.

- [1] Òscar García Benítez, Control predictiu d'un inversor de tres nivells programat amb llenguatge C. 2012.
- [2] S. Alepuz, Apunts d'Electrònica de potència. EUPMT, 2013.
- [3] S. Alepuz, S. Busquets-Mongue, J. Bordonau, P. Cortés, S. Kouro, *IEEE Energy Conversion Congress and Exposition (ECCE) - Control Method for Low Voltage Ride-Through Compliance of Grid-Connected NPC Converter Based Wind Power Systems Using Predictive Control*. IEEE, 2009.
- [4] [http://www.gte.us.es/~leopoldo/Store/tsp\\_16.pdf](http://www.gte.us.es/~leopoldo/Store/tsp_16.pdf)
- [5] R. Vargas, P. Cortés, U. Ammann, J. Rodríguez, J. Pontt, *IEEE Transactions on industrial electronics - Predictive Control of a Three Phase-Phase-Neutral-Point-Clamped Inverter*. IEEE, 2009.
- [6] P. Cortés, J. Rodríguez, C. Silva, A. Flores, *IEEE Transactions on industrial electronics - Delay Compensation in Model predictive Current Control of a Three-Phase Inverter*. IEEE, Vol. 59, Núm. 2, Febrer 2012.
- [7] S. Kouro, P. Cortés, R. Vargas, U. Ammann, J. Rodríguez, *IEEE Transactions on industrial electronics - Model Predictive Control – A Simple and Powerful Method to Control Power Converters*. Vol. 56, Núm 6, Juny 2009.
- [8] <http://c.learnthethehardway.org/book/>
- [9] <http://es.mathworks.com/help/matlab/>
- [10] dSpace, *DS1104 R&D Controller Board, Hardware Installation and Configuration*. Release 4.1, March 2004.
- [11] dSpace, *DS1104 R&D Controller Board, RTLib Reference*. Release 4.2, March 2005.