

Escola Universitària Politécnica de Mataró

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA**

ENGINYERIA TÈCNICA INDUSTRIAL: ESPECIALITAT ELECTRÒNICA INDUSTRIAL

CONTROL PREDICTIU D'UN INVERSOR DE TRES NIVELLS PROGRAMAT AMB LENGUATGE C

Memòria

OSCAR GARCIA BENITEZ

PONENT: SALVADOR ALEPUZ

TARDOR 2012



**TecnoCampus
Mataró-Maresme**

Dedicatòria

A tots els que van confiar en la meva feina i dedicació.

Agraïments

Al meu professor ponent Salvador Alepuz, sense la seva aportació la feina no hagués sigut la mateixa. I especialment, als meus pares.

Resum

A partir del control predictiu d'un inversor de tres nivells programat sobre Matlab-Simulink, s'ha traduït el programa a llenguatge de programació C per tal de millorar la seva velocitat d'execució. S'han ofert dos programes diferents amb la mateixa finalitat i s'ha demostrat, mitjançant les simulacions, el correcte funcionament de tots dos, amb una considerable millora del segon programa presentat respecte del primer.

Resumen

A partir del control predictivo de un inversor de tres niveles programado sobre Matlab-Simulink, se ha traducido el programa a lenguaje de programación C por tal de mejorar su velocidad de ejecución. Se han ofrecido dos programas diferentes con la misma finalidad y se ha demostrado, mediante las simulaciones, el correcto funcionamiento de los mismos, con una considerable mejora del segundo programa presentado respecto al primero.

Abstract

Starting from predictive control of a three-level inverter programmed on Matlab-Simulink, the same program has been translated into C programming language, in order to improve their execution speed. Two different programs with the same purpose have been obtained and their correct operation has been verified by simulation, with a significant improvement of the latter program with respect to the former one.

Índex.

Índex de figures.....	III
Índex de taules.....	V
1. Objectius.....	1
1.1. Propòsit.....	1
1.2. Finalitat.....	1
1.3. Objecte.....	1
1.4. Abast.....	1
2. Convertidor CC/CA de tres nivells.....	3
2.1. Introducció al convertidor CC/CA de tres nivells.....	3
2.2. Topologia del convertidor CC/CA de tres nivells.....	4
2.3. Generació dels senyals de commutació dels interruptors.....	9
2.3.1. Vectors d'estat.....	11
2.3.2. Estats de commutació.....	13
3. Control predictiu.....	17
3.1. Introducció al control predictiu.....	17
3.2. Model del sistema.....	18
3.3. Mètode del control predictiu del corrent.....	21
4. Programa amb llenguatge de programació C.....	23
4.1. Particularitats de la programació en C.....	23
4.2. Estructura del programa.....	25
4.2.1. Estructura del programa un pas.....	25
4.2.2. Estructura del programa dos passos.....	26

II

4.3. Programa detallat.....	28
4.3.1. Programa detallat un pas.....	28
4.3.2. Programa detallat dos passos.....	37
5. Resultats de les simulacions del programa amb llenguatge de programació C.....	47
5.1. Simulacions del programa un pas.....	47
5.2. Simulacions del programa dos passos.....	51
6. Conclusions.....	55
7. Referències.....	57

Índex de figures.

Fig. 2.1. Esquema del circuit d'un convertidor CC/CA de tres nivells amb topologia NPC.....	3
Fig. 2.2. Tensió de sortida V_{ao} en funció dels interruptors connectats.....	6
Fig. 2.3. Tensió de sortida V_{bo} en funció dels interruptors connectats.....	6
Fig. 2.4. Tensió de sortida V_{co} en funció dels interruptors connectats.....	7
Fig. 2.5. Model equivalent amb commutadors de tres posicions del convertidor de tres nivells amb topologia NPC.....	9
Fig. 2.6. Diagrama de vectors d'estat pel convertidor de 3 nivells.....	11
Fig. 3.1. Circuit inversor de 3 nivells amb topologia NPC, connectat a la xarxa mitjançant un filtre (L), i diagrama de control.....	18
Fig. 3.2. Esquema d'un circuit R-L.....	18
Fig. 3.3. Període de mostreig T_s	19
Fig. 4.1. Bloc del Predictive Control al Simulink amb l'Embedded MATLAB Function.....	24
Fig. 4.2. Bloc del Predictive Control al Simulink amb la S-Function (llenguatge C).....	24
Fig. 5.1. Corrents de referència i_a , i_b i i_c (A).....	47
Fig. 5.2. Corrents i_a , i_b i i_c (A).....	48
Fig. 5.3. Tensió de desequilibri v_o (V).....	48
Fig. 5.4. Corrents i_a , i_b i i_c (A).....	49
Fig. 5.5. Corrents i_a , i_b i i_c (A).....	50
Fig. 5.6. Tensió de desequilibri v_o (V).....	50
Fig. 5.7. Corrents de referència i_a , i_b i i_c (A).....	51
Fig. 5.8. Corrents i_a , i_b i i_c (A).....	52
Fig. 5.9. Tensió de desequilibri v_o (V).....	52
Fig. 5.10. Corrents i_a , i_b i i_c (A).....	53

IV

Fig. 5.11. Corrents i_a , i_b i i_c (A).....	54
Fig. 5.12. Tensió de desequilibri v_o (V).....	54

Índex de taules.

Taula 2.1. Interruptors a commutar per obtenir els tres nivells en la tensió de sortida V_{ao}	4
Taula 2.2. Interruptors a commutar per obtenir els tres nivells en la tensió de sortida V_{bo}	5
Taula 2.3. Interruptors a commutar per obtenir els tres nivells en la tensió de sortida V_{co}	5
Taula 2.4. Estats de commutació d'a	10
Taula 2.5. Estats de commutació de b.	10
Taula 2.6. Estats de commutació de c.	10

1. Objectius.

1.1. Propòsit.

Realització d'un programa sobre el Matlab-Simulink amb llenguatge de programació C per fer el control predictiu d'un inversor de tres nivells amb topologia NPC.

1.2. Finalitat.

Millorar la velocitat d'execució del programa sobre el control predictiu d'un inversor de tres nivells amb topologia NPC i preparar-lo per la seva implementació en el sistema experimental.

1.3. Objecte.

Dos programes amb llenguatge de programació C que fan el control predictiu d'un inversor de tres fases amb topologia NPC, per treballar a sobre del programa Matlab-Simulink.

1.4. Abast.

A partir del control predictiu d'un inversor de tres nivells programat sobre Matlab-Simulink, s'ha traduït el programa a llenguatge de programació C per tal de millorar la seva velocitat d'execució.

S'han creat dos programes i s'ha verificat, mitjançant simulacions, el seu correcte funcionament. Els dos programes s'han preparat per implementar-los en el sistema experimental.

2. Convertidor CC/CA de tres nivells.

2.1. Introducció al convertidor CC/CA de tres nivells.

Aquest projecte es centra en el convertidor CC/CA de tres nivells, un cas particular dins dels convertidors multinivell.

Els circuits convertidors CC/CA s'anomenen Inversors o Onduladors. Amb aquests convertidors es desitja tenir una tensió a la sortida tan sinusoidal com sigui possible, poder regular el valor eficaç i el de la freqüència de la tensió de sortida.[2]

En moltes aplicacions es té la necessitat de generar un sistema trifàsic de tensions alternes per alimentar càrregues trifàsiques. És en aquests casos on es necessita un inversor trifàsic.

Tot seguit es mostra el model equivalent del circuit d'un convertidor CC/CA de tres nivells amb topologia NPC.

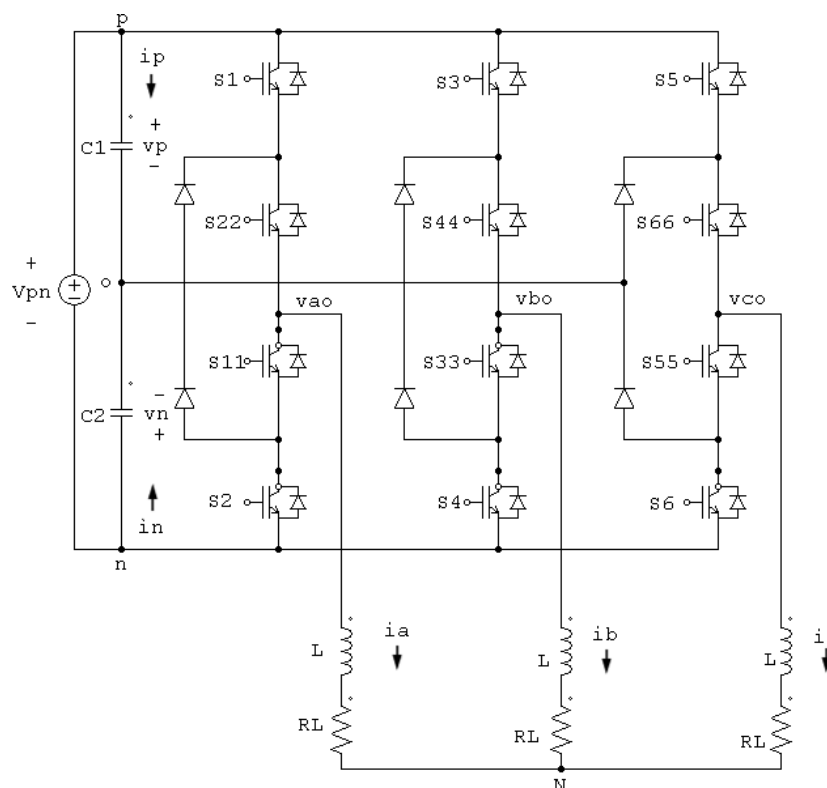


Fig. 2.1. Esquema del circuit d'un convertidor CC/CA de tres nivells amb topologia NPC.

2.2. Topologia del convertidor CC/CA de tres nivells.

Actualment, la investigació i el desenvolupament dels convertidors presenten diferents topologies. La topologia d'inversor trifàsic que s'ha fet servir és coneguda com a Neutral Point Clamped (NPC), topologia d'inversor de punt neutre.

El disseny de la topologia permet obtenir voltatges alterns de tres nivells a cada fase de sortida a partir d'una única font de tensió contínua, commutant en les seqüències adequades pels quatre transistors de cada branca.

En el circuit de la figura 2.1 la tensió del bus de contínua V_{pn} es divideix mitjançant dues capacitats, on el punt mig 'o' es defineix també com el punt neutre del bus de contínua. La tensió de sortida per a cada fase (V_{ao} , V_{bo} , V_{co}) pot agafar tres nivells diferents: $V_{pn}/2$, 0 i $-V_{pn}/2$, d'aquesta manera s'observa que el convertidor presenta nou estats diferents de commutació dels transistors.

A la taula 2.1, 2.2 i 2.3 es mostra l'accionament adequat dels interruptors per obtenir els tres nivells en la tensió de sortida de V_{ao} , V_{bo} i V_{co} .

Interruptors connectats	Tensió V_{ao}
{S1-S22}	$V_{pn}/2$
{S22-S11}	0
{S11-S2}	$-V_{pn}/2$

Taula 2.1. Interruptors a commutar per obtenir els tres nivells en la tensió de sortida V_{ao} .

Interruptors connectats	Tensió Vbo
{S3-S44}	$V_{pn}/2$
{S44-S33}	0
{S33-S4}	$-V_{pn}/2$

Taula 2.2. Interruptors a commutar per obtenir els tres nivells en la tensió de sortida Vbo.

Interruptors connectats	Tensió Vco
{S5-S66}	$V_{pn}/2$
{S66-S55}	0
{S55-S6}	$-V_{pn}/2$

Taula 2.3. Interruptors a commutar per obtenir els tres nivells en la tensió de sortida Vco.

A les figures 2.2, 2.3 i 2.4 es mostra el resultat il·lustrat de l'accionament adequat dels interruptors. A cada branca es troben dos parells complementaris, es defineix el parell d'interruptors de manera que si n'hi ha un tancat obliga a l'altre a estar obert. Per a aquest convertidor, són parells complementaris S1-S11 i S22-S2 per la fase Vao, S3-S33 i S44-S4 per la fase Vbo i S5-S55 i S66-S6 per la fase Vco.

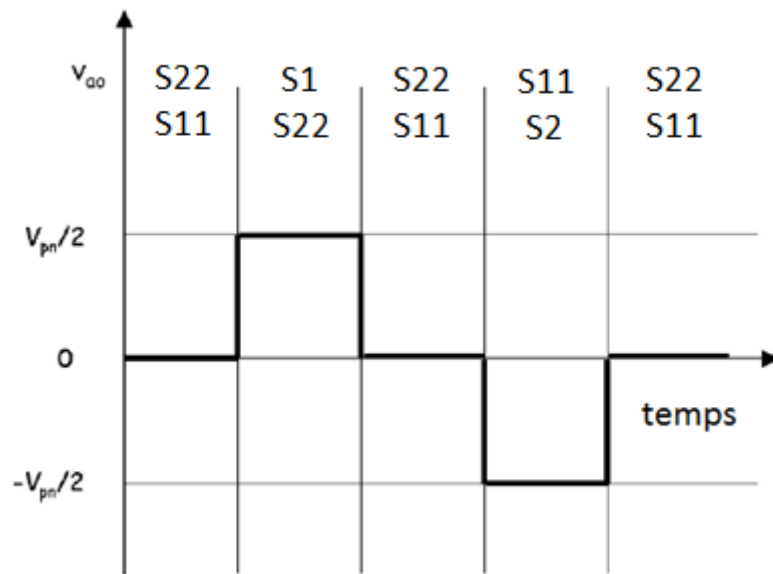


Fig. 2.2. Tensió de sortida V_{ao} en funció dels interruptors connectats.

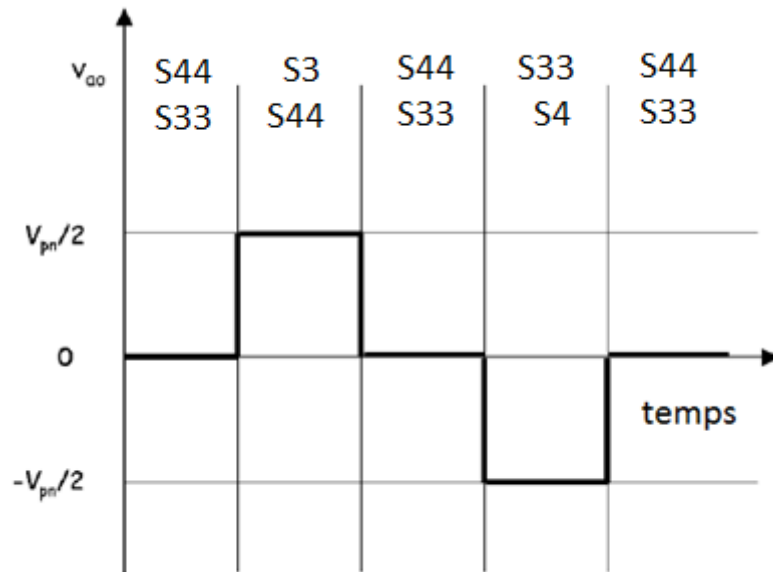


Fig. 2.3. Tensió de sortida V_{bo} en funció dels interruptors connectats.

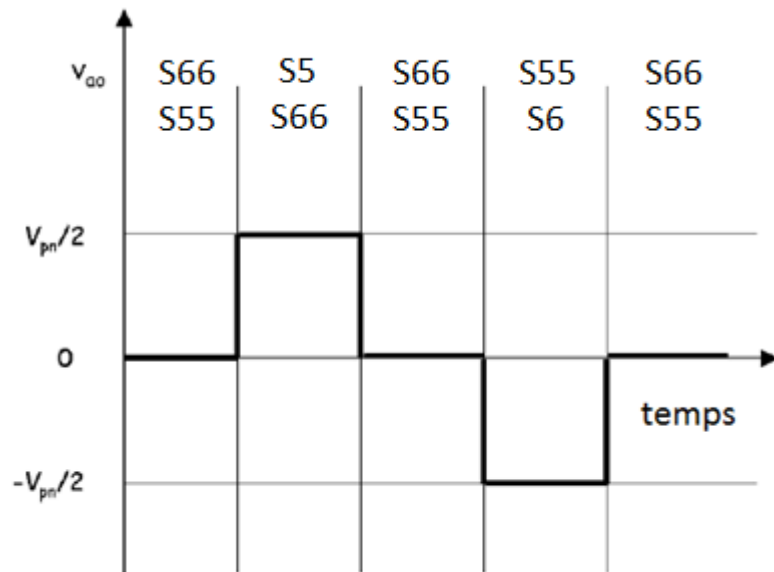


Fig. 2.4. Tensió de sortida V_{co} en funció dels interruptors connectats.

Els díodes connectats al punt mig del bus de contínua 'o' són l'element clau que diferencia aquest circuit d'un inversor convencional, fent la funció de fixar (*clamp*) les tensions de bloqueig dels interruptors a una fracció de la tensió del bus de contínua, a la tensió d'una capacitat $V_{pn}/2$. Els díodes de fixació també han de bloquejar la mateixa tensió, en aquest convertidor. D'altra banda, la tensió de sortida analitzada V_{ao} , V_{bo} i V_{co} és alterna, s'ha pogut comprovar. No obstant, si es considera com a tensió de sortida la existent entre la fase i el negatiu 'n', per exemple V_{an} , s'obtenen tres nivells de tensió positius (V_{pn} , $V_{pn}/2$, 0) i el convertidor treballa com a convertidor CC/CC. En una aplicació de contínua, el corrent de sortida és unidireccional i no és possible mantenir equilibrat el punt mig del bus de contínua si s'aplica a una sola branca, per això aquesta topologia no és viable per a la conversió CC/CC, a menys que s'apliquin a dues branques.

Els avantatges de la topologia d'aquest convertidor són:

- La tensió de bloqueig dels interruptors és la tensió d'una capacitat d'entrada $V_{pn}/2$.
- El nombre de capacitats requerides és petit en comparació amb altres topologies. Aquest punt és interessant ja que són els components reactius els que suposen un major cost en el convertidor.

- Es poden connectar directament a un bus de contínua, sense necessitat de crear altres busos addicionals.
- No requereixen transformadors.
- Canvia d'un estat a un altre accionant només un interruptor.

Per una altra banda, els inconvenients són:

- Els díodes de fixació requereixen una recuperació ràpida i són capaços de conduir el corrent nominal del convertidor, a més d'estar sotmesos a una recuperació inversa exigent. Tots aquests requisits han de tenir-se en compte en el disseny del convertidor. Tanmateix, els interruptors interns de cada branca poden arribar a suportar pics de tensió en el moment de la commutació, provocades per les inductàncies paràsites.
- És necessari que les tensions de les capacitats es mantinguin equilibrades en qualsevol punt de treball, complicant el sistema de control del convertidor. L'equilibrat de les capacitats es dificulta conforme s'incrementa el número de nivells, fins i tot pot ser impossible en algunes condicions d'operació.

Un cop s'han vist els avantatges i els inconvenients dels convertidors de tres nivells, s'observa que la topologia reuneix tots els avantatges que s'han esmentat prèviament i no presenten aquells inconvenients que apareixen quan es tenen més de tres nivells. Per passar de dos a tres nivells, només és necessari afegir dos díodes de fixació per fase, a més de presentar un disseny mecànic simple, a diferència dels convertidors de més nivells. Per aquesta raó, la topologia NPC ha sigut la més extensament estudiada i aplicada. Però, es poden trobar muntatges de diversos nivells funcionant comercialment.[1]

2.3. Generació dels senyals de commutació dels interruptors.

El convertidor CC/CA de tres nivells presenta 27 possibles estats de commutació. A la Fig. 2.5. es mostra un model equivalent del circuit del convertidor on els transistors es simplifiquen per commutadors de 3 posicions, per així ajudar a la comprensió del seu funcionament. Cadascuna de les tres fases de sortida es poden connectar a un dels tres nivells del bus de contínua en cada instant.

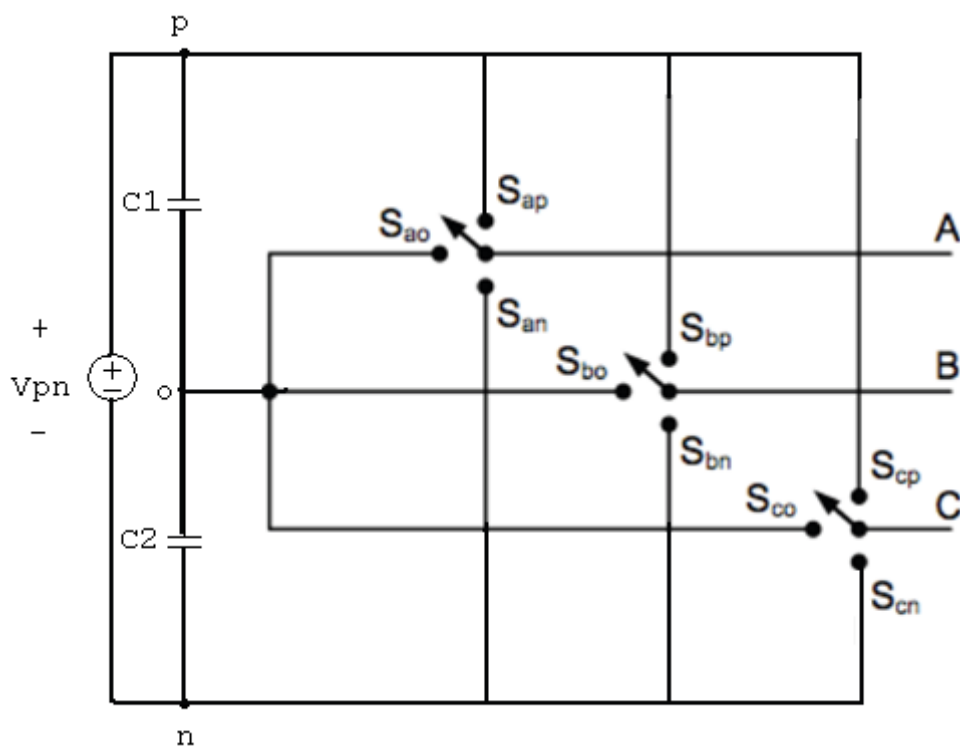


Fig. 2.5. Model equivalent amb commutadors de tres posicions del convertidor de tres nivells amb topologia NPC.

A la Taula 2.4, 2.5 i 2.6 es mostren els estats possibles de commutació per a les tres fases (a, b i c) del nostre convertidor.

	a→p	a→0	a→n
Sap	1	0	0
Sa0	0	1	0
San	0	0	1

Taula 2.4. Estats de commutació d'a.

	b→p	b→0	b→n
Sbp	1	0	0
Sb0	0	1	0
Sbn	0	0	1

Taula 2.5. Estats de commutació de b.

	c→p	c→0	c→n
Scp	1	0	0
Sc0	0	1	0
Scn	0	0	1

Taula 2.6. Estats de commutació de c.

2.3.1. Vectors d'estat.

Aquest apartat està dedicat a l'obtenció dels 19 vectors de tensió del nostre convertidor.

A la Fig. 2.6. es mostra el diagrama de vectors d'estat per al convertidor de tres nivells, amb 27 estats possibles de commutació expressats per 19 vectors d'estats diferents, juntament amb un possible vector de modulació. El diagrama de vectors d'estat és universal per als convertidors de tres nivells amb topologia NPC.

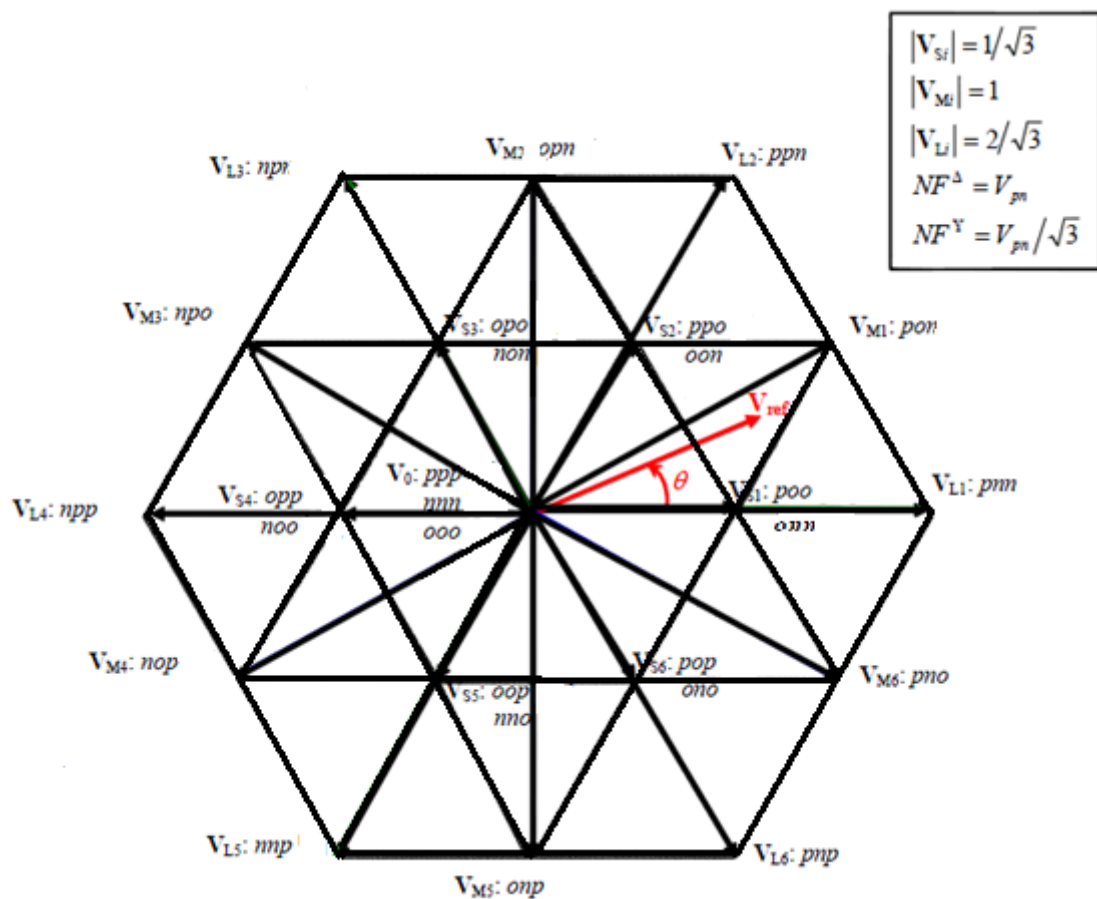
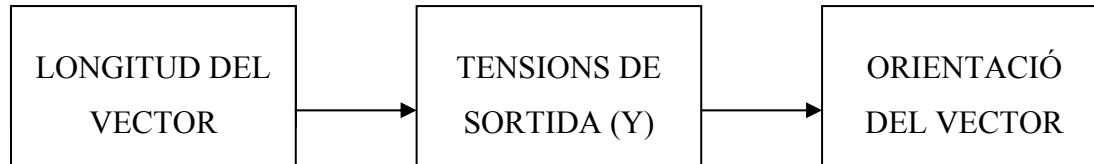


Fig. 2.6. Diagrama de vectors d'estat pel convertidor de 3 nivells.

Com es pot observar a la Fig. 2.6., hi ha un quadre al costat del diagrama de vectors on s'explica quins valors s'han d'utilitzar segons la mida dels vectors (V_S , V_M i V_L) i també en funció de la connexió que farem servir, si en triangle o en estrella (NF^Δ i NF^Y). En el nostre cas serà en estrella. Amb tot això, i tenint en compte l'orientació dels angles de cada vector, ja es podran obtenir els 19 vectors.



$$\text{Vector 1: } \mathbf{V}_0 = 0 + j0 \quad (2.1)$$

$$\text{Vector 2: } \mathbf{V}_{S1} = \frac{1}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 0^\circ + j \sin 0^\circ) = \frac{V_{pn}}{3} \cdot (1 + j0) \quad (2.2)$$

$$\text{Vector 3: } \mathbf{V}_{S2} = \frac{1}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 60^\circ + j \sin 60^\circ) = \frac{V_{pn}}{3} \cdot (0.5 + j0.866) \quad (2.3)$$

$$\text{Vector 4: } \mathbf{V}_{S3} = \frac{1}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 120^\circ + j \sin 120^\circ) = \frac{V_{pn}}{3} \cdot (-0.5 + j0.866) \quad (2.4)$$

$$\text{Vector 5: } \mathbf{V}_{S4} = \frac{1}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 180^\circ + j \sin 180^\circ) = \frac{V_{pn}}{3} \cdot (-1 + j0) \quad (2.5)$$

$$\text{Vector 6: } \mathbf{V}_{S5} = \frac{1}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 240^\circ + j \sin 240^\circ) = \frac{V_{pn}}{3} \cdot (-0.5 - j0.866) \quad (2.6)$$

$$\text{Vector 7: } \mathbf{V}_{S6} = \frac{1}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 300^\circ + j \sin 300^\circ) = \frac{V_{pn}}{3} \cdot (0.5 - j0.866) \quad (2.7)$$

$$\text{Vector 8: } \mathbf{V}_{L1} = \frac{2}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 0^\circ + j \sin 0^\circ) = \frac{2 \cdot V_{pn}}{3} \cdot (1 + j0) \quad (2.8)$$

$$\text{Vector 9: } \mathbf{V}_{M1} = 1 \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 30^\circ + j \sin 30^\circ) = \frac{V_{pn}}{\sqrt{3}} \cdot (0.866 + j0.5) \quad (2.9)$$

$$\text{Vector 10: } \mathbf{V}_{L2} = \frac{2}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 60^\circ + j \sin 60^\circ) = \frac{2 \cdot V_{pn}}{3} \cdot (0.5 + j0.866) \quad (2.10)$$

$$\text{Vector 11: } \mathbf{V}_{M2} = 1 \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 90^\circ + j \sin 90^\circ) = \frac{V_{pn}}{\sqrt{3}} \cdot (0 + j) \quad (2.11)$$

$$\text{Vector 12: } \mathbf{V}_{L3} = \frac{2}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 120^\circ + j \sin 120^\circ) = \frac{2 \cdot V_{pn}}{3} \cdot (-0.5 + j0.866) \quad (2.12)$$

$$\text{Vector 13: } \mathbf{V}_{M3} = 1 \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 150^\circ + j \sin 150^\circ) = \frac{V_{pn}}{\sqrt{3}} \cdot (-0.866 + j0.5) \quad (2.13)$$

$$\text{Vector 14: } \mathbf{V}_{L4} = \frac{2}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 180^\circ + j \sin 180^\circ) = \frac{2 \cdot V_{pn}}{3} \cdot (-1 + j0) \quad (2.14)$$

$$\text{Vector 15: } \mathbf{V}_{M4} = 1 \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 210^\circ + j \sin 210^\circ) = \frac{V_{pn}}{\sqrt{3}} \cdot (-0.866 - j0.5) \quad (2.15)$$

$$\text{Vector16: } \mathbf{V}_{L5} = \frac{2}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 240^\circ + j \sin 240^\circ) = \frac{2 \cdot V_{pn}}{3} \cdot (-0.5 - j0.866) \quad (2.16)$$

$$\text{Vector 17: } \mathbf{V}_{M5} = 1 \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 270^\circ + j \sin 270^\circ) = \frac{V_{pn}}{\sqrt{3}} \cdot (0 - j) \quad (2.17)$$

$$\text{Vector18: } \mathbf{V}_{L6} = \frac{2}{\sqrt{3}} \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 300^\circ + j \sin 300^\circ) = \frac{2 \cdot V_{pn}}{3} \cdot (0.5 - j0.866) \quad (2.18)$$

$$\text{Vector19: } \mathbf{V}_{M6} = 1 \cdot \frac{V_{pn}}{\sqrt{3}} \cdot (\cos 330^\circ + j \sin 330^\circ) = \frac{V_{pn}}{\sqrt{3}} \cdot (0.866 - j0.5) \quad (2.19)$$

2.3.2. Estats de commutació.

Aquest apartat està dedicat a l'obtenció dels 27 possibles estats de commutació del nostre convertidor.

Tal com s'ha mostrat a la Fig. 2.7. en el diagrama de vectors d'estat, s'observen els 27 estats possibles de commutació expressats pels 19 vectors d'estat.

Cada punt del bus de contínua (p, 0 i n) ens indica a on està connectada cada fase (a, b i c), i quin corrent (ia, ib i ic) és el que s'obté per a cada commutació.

Al vector V_{S1} es troben dos estats possibles de commutació:

$$\text{Estat 1: } \quad \mathbf{p00} \quad \rightarrow \quad ip = ia \quad i0 = ib+ic \quad in = 0$$

$$\text{Estat 2: } \quad \mathbf{onn} \quad \rightarrow \quad ip = 0 \quad i0 = ia \quad in = ib+ic$$

Al vector V_{S2} es troben dos estats possibles de commutació:

$$\text{Estat 3: } \quad \mathbf{ppo} \quad \rightarrow \quad ip = ia+ib \quad i0 = ic \quad in = 0$$

$$\text{Estat 4: } \quad \mathbf{oon} \quad \rightarrow \quad ip = 0 \quad i0 = ia+ib \quad in = ic$$

Al vector V_{S3} es troben dos estats possibles de commutació:

$$\text{Estat 5: } \quad \mathbf{opo} \quad \rightarrow \quad ip = ib \quad i0 = ia+ic \quad in = 0$$

$$\text{Estat 6: } \quad \mathbf{non} \quad \rightarrow \quad ip = 0 \quad i0 = ib \quad in = ia+ic$$

Al vector V_{S4} es troben dos estats possibles de commutació:

$$\text{Estat 7: } \quad \mathbf{opp} \quad \rightarrow \quad ip = ib+ic \quad i0 = ia \quad in = 0$$

$$\text{Estat 8: } \quad \mathbf{noo} \quad \rightarrow \quad ip = 0 \quad i0 = ib+ic \quad in = ia$$

Al vector V_{S5} es troben dos estats possibles de commutació:

$$\text{Estat 9: } \quad \mathbf{oop} \quad \rightarrow \quad ip = ic \quad i0 = ia+ib \quad in = 0$$

$$\text{Estat 10: } \quad \mathbf{nno} \quad \rightarrow \quad ip = 0 \quad i0 = ic \quad in = ia+ib$$

Al vector V_{S6} es troben dos estats possibles de commutació:

$$\text{Estat 11: } \quad \mathbf{pop} \quad \rightarrow \quad ip = ia+ic \quad i0 = ib \quad in = 0$$

$$\text{Estat 12: } \quad \mathbf{ono} \quad \rightarrow \quad ip = 0 \quad i0 = ia+ic \quad in = ib$$

Al vector V_{L1} es troba un estat possible de commutació:

$$\text{Estat 13: } \quad \mathbf{pnn} \quad \rightarrow \quad ip = ia \quad i0 = 0 \quad in = ib+ic$$

Al vector V_{M1} es troba un estat possible de commutació:

$$\text{Estat 14: } \quad \mathbf{pon} \quad \rightarrow \quad ip = ia \quad i0 = ib \quad in = ic$$

Al vector V_{L2} es troba un estat possible de commutació:

$$\text{Estat 15: } \quad \mathbf{ppn} \quad \rightarrow \quad ip = ia+ib \quad i0 = 0 \quad in = ic$$

Al vector V_{M2} es troba un estat possible de commutació:

$$\text{Estat 16: } \quad \mathbf{opn} \quad \rightarrow \quad ip = ib \quad i0 = ia \quad in = ic$$

Al vector V_{L3} es troba un estat possible de commutació:

$$\text{Estat 17: } \quad \mathbf{npn} \quad \rightarrow \quad ip = ib \quad i0 = 0 \quad in = ia+ic$$

Al vector V_{M3} es troba un estat possible de commutació:

$$\text{Estat 18: } \quad \mathbf{npo} \quad \rightarrow \quad ip = ib \quad i0 = ic \quad in = ia$$

Al vector V_{L4} es troba un estat possible de commutació:

$$\text{Estat 19: } \quad \mathbf{npp} \quad \rightarrow \quad ip = ib+ic \quad i0 = 0 \quad in = ia$$

Al vector V_{M4} es troba un estat possible de commutació:

$$\text{Estat 20: } \quad \mathbf{nop} \quad \rightarrow \quad ip = ic \quad i0 = ib \quad in = ia$$

Al vector V_{L5} es troba un estat possible de commutació:

$$\text{Estat 21: } \quad \mathbf{nnp} \quad \rightarrow \quad ip = ic \quad i0 = 0 \quad in = ia+ib$$

Al vector V_{M5} es troba un estat possible de commutació:

$$\text{Estat 22: } \quad \mathbf{onp} \quad \rightarrow \quad ip = ic \quad i0 = ia \quad in = ib$$

Al vector V_{L6} es troba un estat possible de commutació:

$$\text{Estat 23: } \quad \mathbf{pnp} \quad \rightarrow \quad ip = ia+ic \quad i0 = 0 \quad in = ib$$

Al vector V_{M6} es troba un estat possible de commutació:

$$\text{Estat 24: } \quad \mathbf{pno} \quad \rightarrow \quad ip = ia \quad i0 = ic \quad in = ib$$

Al vector V_0 es troben tres estats possibles de commutació:

Estat 25: **ppp** → $i_p = 0$ $i_0 = 0$ $i_n = 0$

Estat 26: **ooo** → $i_p = 0$ $i_0 = 0$ $i_n = 0$

Estat 27: **nnn** → $i_p = 0$ $i_0 = 0$ $i_n = 0$

Aquests corrents d'aquests tres últims estats donen 0 perquè el neutre està aïllat, no té connexió.

3. Control predictiu.

3.1. Introducció al control predictiu.

El control predictiu és una teoria de control que es va desenvolupar a finals dels 70. La variació d'aquest tipus d'estratègia de control permet realitzar aplicacions de control en convertidors de potència. El control predictiu s'utilitza en control de corrent, en la correcció del factor de potència i en filtres actius. Tots aquests controls són considerats models lineals i s'utilitzen en tècniques de modulació per a la generació de voltatge.[3][5]

La idea bàsica del control predictiu és considerar l'inversor de tres nivells com un sistema lineal, aquesta consideració és contrària a la seva naturalesa discreta i al seu procés de control.

El control predictiu és una estratègia de control que consisteix en generar prediccions a partir d'un model del sistema. Les prediccions realitzades en el model del sistema són avaluades a partir d'una funció de qualitat. Aquesta funció de qualitat minimitza l'aplicació dels diferents estats de l'inversor de tres nivells, tenint en compte la seva naturalesa discreta.

3.2. Model del sistema.

Tal com es veu a la Fig. 3.1., per a obtenir la minimització de la funció de qualitat s'haurà de controlar la tensió als condensadors (V_p i V_n) i els corrents a les fases (i_a , i_b i i_c).

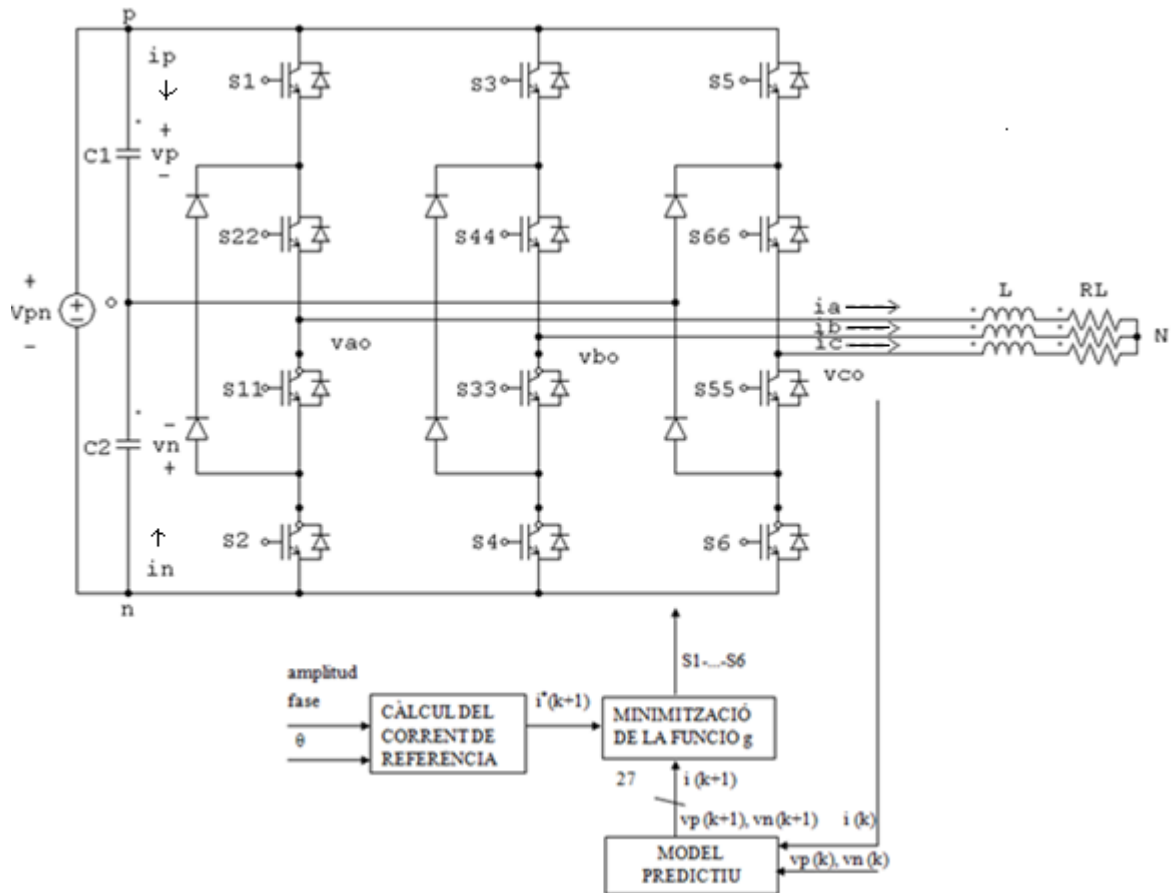


Fig. 3.1. Circuit inversor de 3 nivells amb topologia NPC, connectat a la xarxa mitjançant un filtre (L), i diagrama de control.

El model de l'algorisme del controlador predictiu prové del circuit que es mostra a la Fig. 3.2.

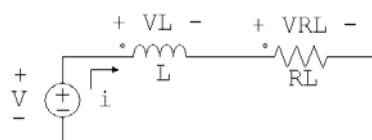


Fig. 3.2. Esquema d'un circuit R-L.

Aplicant la segona llei de Kirchoff al circuit:

$$v = v_L + v_{RL} \rightarrow v = L \cdot \frac{di}{dt} + R_L \cdot i \rightarrow L \cdot \frac{di}{dt} = v - R_L \cdot i$$

$$L \cdot \frac{di(t)}{dt} = v(t) - R_L \cdot i(t) \quad (3.1)$$

Finalment, tal com s'observa a (3.1), s'obté l'algorisme del controlador predictiu del nostre model.

Els càlculs dels vectors de tensió i el corrent estan definits per les equacions (3.2) i (3.3), on a és igual a $e^{j(2\pi/3)}$.

$$v = \frac{2}{3} \cdot (v_{ao} + a \cdot v_{bo} + a^2 \cdot v_{co}) \quad (3.2)$$

$$i = \frac{2}{3} \cdot (i_a + a \cdot i_b + a^2 \cdot i_c) \quad (3.3)$$

El vector de tensió generat per l'inversor NPC és seleccionat entre els 19 vectors diferents obtinguts a partir dels 27 estats de commutació diferents.

La forma derivada de l'aplicació d'un període de mostreig T_s és aproximada per l'expressió que s'observa a (3.4).

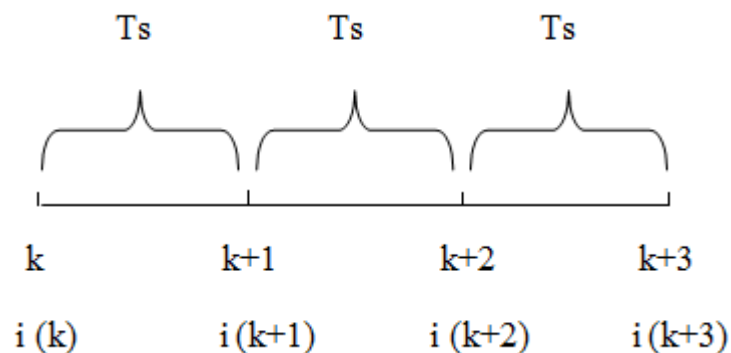


Fig. 3.3. Període de mostreig T_s .

$$\frac{di(t)}{dt} = \frac{\Delta i}{\Delta t} = \frac{\Delta i}{T_s} = \frac{i(k+1) - i(k)}{T_s} \quad \rightarrow \quad \frac{di(t)}{dt} = \frac{i(k+1) - i(k)}{T_s} \quad (3.4)$$

Substituint aquesta expressió en l'algoritme de control predictiu, es pot comprovar a (3.5) la relació que hi ha entre les variables a temps discret.

$$\begin{aligned} L \cdot \frac{i(k+1) - i(k)}{T_s} &= v(t) - R_L \cdot i(t) \rightarrow \\ \rightarrow i(k+1) &= \frac{T_s}{R_L \cdot T_s + L} \cdot \left[\frac{L}{T_s} \cdot i(k) + v(k) \right] \end{aligned} \quad (3.5)$$

L'equació anterior s'utilitza per obtenir prediccions per al futur valor de la línia del corrent $i(k+1)$, considerant tots els vectors de tensió possibles $v(k)$ generats per l'inversor i el vector de corrent de la línia mesurada $i(k)$.

Finalment, el model del sistema es completa amb les equacions del circuit intermedi. A (3.6) i (3.7) es pot comprovar la tensió dels condensadors del circuit intermedi.

$$v_p(k+1) = v_p(k) + \frac{1}{C} \cdot i_p(k) \cdot T_s \quad (3.6)$$

$$v_n(k+1) = v_n(k) + \frac{1}{C} \cdot i_n(k) \cdot T_s \quad (3.7)$$

Els corrents que passen a través dels condensadors (i_p i i_n) es calculen utilitzant els corrents de línia i l'estat de commutació actual, per tant, no es necessiten mesures addicionals. Aquestes equacions s'utilitzen per obtenir prediccions per al valor futur de la tensió dels condensadors en funció del valor del seu corrent i el voltatge present.

3.3. Mètode del control predictiu del corrent.

El futur valor de la línia de corrent $i(k+1)$, i la tensió als condensadors $v_p(k+1)$ i $v_n(k+1)$ es prediuen pels 27 estats possibles de commutació generades pel convertidor, utilitzant el model predictiu de les equacions del sistema. Per aquest motiu cal mesurar el corrent de línia $i(k)$ i les tensions dels condensadors del circuit intermedi $v_p(k)$ i $v_n(k)$.

Després d'obtenir les prediccions, una funció g de qualitat avalua cada estat de commutació. L'estat de commutació que minimitza g es selecciona i s'aplica durant el següent període de mostreig.

A (3.8) es mostra la funció de qualitat g proposada.

$$g = f(i^*(k+1), i(k+1)) + h(v_{p,n}(k+1)) \quad (3.8)$$

El primer terme $f(i^*(k+1), i(k+1))$ en la funció de qualitat g està dedicat a aconseguir un seguiment de la referència, quantificant la diferència entre el corrent de referència $i^*(k+1)$ i la predicció actual $i(k+1)$ en el següent temps de mostreig, per a una commutació de determinat estat, definit en el marc $\alpha\beta$.

Es pot observar a (3.9) la composició de la funció f , o seguiment de costos.

$$f(i^*(k+1), i(k+1)) = |i_\alpha^*(k+1) - i_\alpha(k+1)| + |i_\beta^*(k+1) - i_\beta(k+1)| \quad (3.9)$$

L'objectiu del segon terme $h(v_{p,n}(k+1))$ en la funció de qualitat g (3.8) és per prendre avantatge en la situació de redundància d'un inversor de tres nivells, des del moment que la funció f de seguiment de costos depèn només del vector de tensió seleccionat. La proposta de composició d'aquest terme es mostra a (3.10).

$$h(v_{p,n}(k+1)) = \lambda_{DC} |v_p(k+1) + v_n(k+1)| \quad (3.10)$$

Aquest segon terme que s'ha observat a (3.10) en la funció de qualitat (3.8) és proporcional a la diferència absoluta entre ambdós circuits intermedis de la predicció de la tensió del condensador. Preferiblement, l'estat de commutació redundant de l'inversor NPC generarà menor diferència.

El factor de ponderació λ_{DC} ha de gestionar la relació entre els termes dedicats a la referència i l'equilibri de tensió dins de la funció de qualitat g . Un valor gran de λ_{DC} implica més prioritat per a l'objectiu.

Tal com es veu a (3.9), és evident que l'estratègia del control predictiu requereix una estimació del futur en el vector de corrent de referència $i^*(k+1)$. Depenent del temps de mostreig aplicat i el computat, l'estimació es pot aconseguir per una extrapolació de segon ordre com s'observa a (3.11) o per un temps de mostreig suficientment petit i també per guardar esforços computacionals, és possible considerant $i^*(k+1) = i^*(k)$, de manera que no seria necessària l'extrapolació.

$$i^*(k+1) = 3 \cdot i^*(k) - 3 \cdot i^*(k-1) - 3 \cdot i^*(k-2) \quad (3.11)$$

El vector de corrent de referència i^* es calcula a partir del corrent de referència, la fase de referència i l'angle de la xarxa real (θ).

Al programa dos passos, a diferència del programa un pas, es podrà observar que a l'hora de fer el càlcul de la predicció dels corrents a les fases i les tensions en els condensadors, es requereix una estimació dels futurs vectors de corrent per $i(k+1)$ i $i(k+2)$, tal com s'observa a (3.12) i (3.13).

La funció de cost del programa dos passos s'avalua per la predicció de corrent a l'instant $k+2$, i l'estat de commutació que minimitza aquesta funció de cost és seleccionada i emmagatzemada per al següent instant.[4]

$$i(k+1) = \left(1 - \frac{R_L \cdot T_s}{L}\right) \cdot i(k) + \frac{T_s}{L} \cdot v(k) \quad (3.12)$$

$$i(k+2) = \left(1 - \frac{R_L \cdot T_s}{L}\right) \cdot i(k+1) + \frac{T_s}{L} \cdot v(k+1) \quad (3.13)$$

4. Programa amb llenguatge de programació C.

4.1. Particularitats de la programació en C.

El primer que s'ha hagut de fer és crear un bloc S-Function, per poder fer el programa en llenguatge C, i substituir-lo dins del Predictive Control al bloc de l'Embedded MATLAB Function del programa ja existent, tal com es veu a (4.1) i (4.2).

La primera particularitat que s'observa és que per poder invocar el programa en llenguatge C des de Matlab s'ha de crear una arxiu en C (programa.c) amb el programa que s'utilitzarà i fer servir la funció "mex" per a poder invocar i compilar el programa al Matlab (>> mex programa.c).

Una altra particularitat important és el tractament dels nombres complexos. Amb llenguatge C s'ha hagut de separar en dues parts els càlculs amb termes complexos, d'una banda la part real i de l'altra, la part imaginària ($z_1 = a$ i $z_2 = b$). En canvi, el programa de l'Embedded els tracta amb normalitat ($z = a + b \cdot j$). Com es pot observar a (4.1) i (4.2) al bloc amb el programa amb llenguatge C hi ha 8 entrades, en canvi a l'Embedded n'hi ha 5, per cada entrada amb nombres complexos s'han hagut de posar dues entrades (una per la part real i l'altra per la part imaginària).

Una altra particularitat important és que en llenguatge C les operacions amb números reals s'han de fer posant el punt, és a dir, si es posa, per exemple, 3, n fa el càlcul correcte, cal posar 3.0.

Per fer el càlcul amb valor absolut amb llenguatge C s'ha hagut d'utilitzar la funció "fabs" en comptes d'"abs", com a l'Embedded Matlab Function.

L'última particularitat que s'observa amb la programació en llenguatge C, es troba a l'hora de declarar les variables globals, són variables persistents i s'han de declarar fora de la rutina principal del programa, es declararan immediatament després de les llibreries, com s'observarà més endavant, on s'explica més detalladament l'estructura del programa.

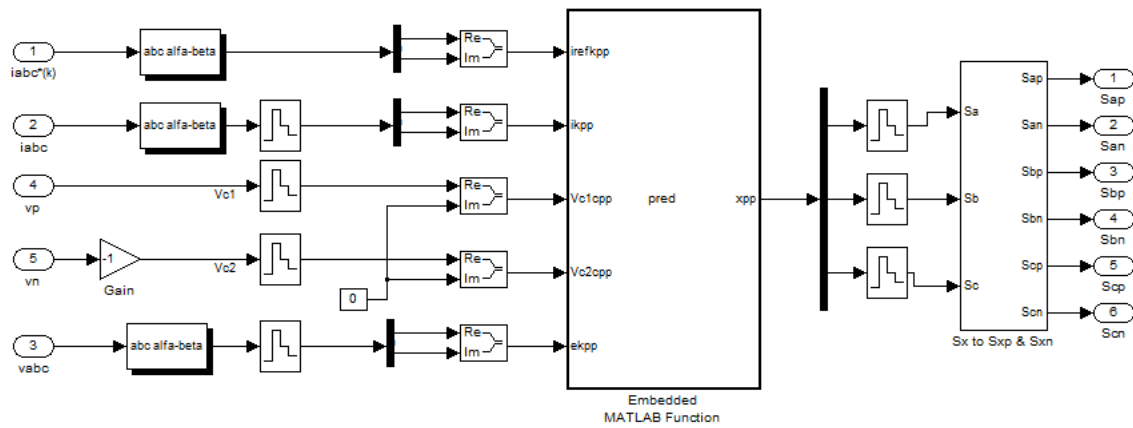


Fig. 4.1. Bloc del Predictive Control al Simulink amb l'Embedded MATLAB Function.

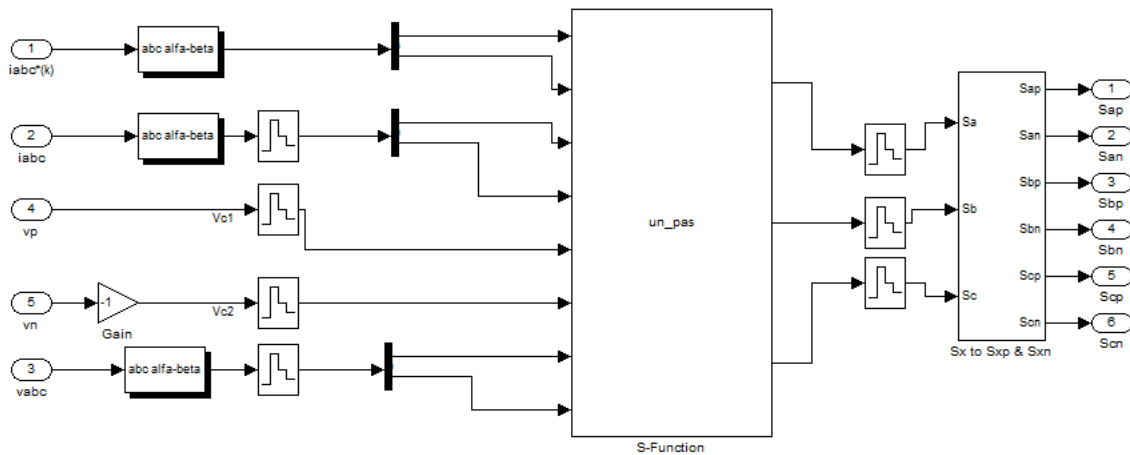


Fig. 4.2. Bloc del Predictive Control al Simulink amb la S-Function (llenguatge C).

4.2. Estructura del programa.

4.2.1. Estructura del programa un pas.

Per programar el bloc S-Function en llenguatge C s'ha d'utilitzar una capçalera on es defineix el nom de la funció, el nom de l'arxiu i les llibreries necessàries. Tot seguit, es defineixen les entrades i sortides amb una funció pròpia del Matlab.

Un cop s'han definit totes les entrades i sortides del nostre sistema, la següent tasca del programa és definir el temps de mostreig, amb una funció pròpia del Matlab per definir el temps de mostreig.

Funció principal del bloc

Es començarà la funció principal del bloc amb una funció pròpia del sistema com es podrà observar més endavant.

Es defineixen les variables constants, aquestes variables mantindran durant tot el programa el mateix valor. Tot seguit, es declaren totes les variables.

Es defineixen els 19 vectors de tensió possibles del nostre convertidor, i els 27 estats possibles de commutació.

Després, s'introdueix l'adquisició de dades d'entrades i sortides mitjançant punters, cosa necessària al nostre programa per les 8 entrades i les 3 sortides.

Tot seguit, s'introdueix l'adjudicació de les variables per a cada punter, a les 8 entrades dels nostre sistema se li adjudiquen els 8 punters d'entrades de dades.

Funció

El següent pas és introduir les variables que necessiten ser inicialitzades.

Per evitar l'extrapolació del corrent de referència, s'ha simplificat aquest pas fent una igualació del corrent de referència $i^*(k)$ i el corrent en $i^*(k+1)$.

És el moment de l'avaluació i l'optimització de g . Es recorren els 19 vectors de tensió amb els seus estats redundants corresponents. Com ja s'ha comprovat en el capítol anterior hi ha vectors que tenen diferents estats.

El següent pas és el càlcul de la predicció del corrent i dels corrents a les fases a partir dels 19 vectors. Es recorren tots els estats possibles de commutació i es fa el càlcul de corrent i de tensió dels condensadors del nostre convertidor.

Seguidament es fa l'avaluació de la funció de qualitat. Un cop ha recorregut tots els vectors i estats possibles, el programa es queda amb el valor més òptim (vector òptim, estat òptim i error mínim).

La sortida del nostre sistema és l'estat de commutació que minimitza g , les sortides es treuen per les variables dels punters que anteriorment s'han adjudicat.

Finalment, s'utilitza una funció pròpia del Matlab per tancar i finalitzar el nostre programa.

4.2.2. Estructura del programa dos passos.

Tal com s'ha explicat anteriorment, per programar el bloc S-Function en llenguatge C s'ha d'utilitzar una capçalera on es defineix el nom de la funció, el nom de l'arxiu i les llibreries necessàries.

A diferència de l'altre programa, com que en aquest es faran servir variables globals, les haurem de definir immediatament després de les llibreries. Tot seguit, es defineixen les entrades i sortides amb una funció pròpia del Matlab.

Un cop s'han definit totes les entrades i sortides del nostre sistema, la següent tasca del programa és definir el temps de mostreig, amb una funció pròpia del Matlab per definir el temps de mostreig.

Funció principal del bloc

Es començarà la funció principal del bloc amb una funció pròpia dels sistema com es podrà observar més endavant.

Es defineixen les variables constants, aquestes variables mantindran durant tot el programa el mateix valor. Tot seguit, es declaren totes les variables que es faran servir en el nostre programa.

A diferència de l'altre programa, no caldrà definir els 19 vectors de tensió possibles del nostre convertidor i els 27 estats possibles de commutació.

Després de declarar les variables, s'introdueix l'adquisició de dades d'entrades i sortides mitjançant punters, cosa necessària al nostre programa per les 8 entrades i les 3 sortides.

Tot seguit, s'introdueix l'adjudicació de les variables per a cada punter, a les 8 entrades dels nostre sistema li adjudiquem els 8 punters d'entrades de dades.

Funció

El següent pas és introduir les variables que necessiten ser inicialitzades.

Per evitar l'extrapolació del corrent de referència, com en el programa anterior, s'ha simplificat aquest pas fent una igualació del corrent de referència $i^*(k)$ i el corrent en $i^*(k + 1)$.

El següent pas del programa és la realització del càlcul de la predicció del corrent per $k + 1$, i dels corrents a les fases a l'instant $k + 1$. I tot seguit, es fa el càlcul de la predicció de les tensions als circuits intermedis per $k + 1$.

És el moment de l'avaluació i l'optimització de g sobre els 27 estats de commutació possibles. Es recorren els 19 vectors de tensió amb els seus estats redundants corresponents.

Tot seguit, es torna a fer el càlcul de la predicció del corrent i dels corrents a les fases, però en aquest cas per l'instant $k + 2$. I com anteriorment, es fa el càlcul de la predicció de les tensions als circuits intermedis per l'instant $k + 2$.

Seguidament, es fa l'avaluació de la funció de qualitat. Un cop ha recorregut tots els vectors i estats possibles, el programa es queda amb el valor més òptim (vector òptim, estat òptim i error mínim).

En el següent i últim pas, el nostre programa extreu, mitjançant variables globals que ja han sigut declarades anteriorment, l'estat de commutació per al següent període de commutació del nostre convertidor.

La sortida del nostre sistema és l'estat de commutació que minimitza g , les sortides es treuen per les variables dels punters que anteriorment s'han adjudicat.

Finalment, s'utilitza una funció pròpia del Matlab per tancar i finalitzar el nostre programa.

4.3. Programa detallat.

4.3.1. Programa detallat un pas.

```
/* Capçalera */
/* Nom de l'arxiu */
#define S_FUNCTION_NAME un_pas
#define S_FUNCTION_LEVEL 2
/* Llibreries a utilitzar */
#include "simstruc.h"
#include <math.h>
#include <assert.h>
#include <stdlib.h>
/* Definició d'entrades i sortides */
static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, 0);
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount (S)) {
        return;
    }
    /* 8 entrades */
    if (!ssSetNumInputPorts(S,8)) return;
    {int_T i;
        for (i=0; i<8; i++)
```

```

    {
        ssSetInputPortWidth(S, i, 1);

        ssSetInputPortDirectFeedThrough (S, i, 1);
    }
}

/* 3 sortides */
if (!ssSetNumOutputPorts(S, 3)) return;
{ int_T i;
    for (i=0; i<3; i++)
    {
        ssSetOutputPortWidth(S, i, 1);
    }
}

ssSetNumSampleTimes(S, 1);

ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE
SS_OPTION_USE_TLC_WITH_ACCELERATOR |SS_OPTION_PLACE_ASAP);
}

/* Inicialització del temps de mostreig */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime (S, 0, 100.0e-6);
    ssSetOffsetTime (S, 0, 0.0);
}

/* Funció principal del bloc */
static void mdlOutputs(SimStruct *S, int_T tid)
{
    int_T width = ssGetOutputPortWidth(S,0);

    /* Constants */

    /* Tensió */

    real_T Vdc=100.0;

    /* Filtre */

```

```
real_T L=5.0e-3;

/* Resistència de la bobina */

real_T R=10.0;

/* Temps de mostreig */

real_T Ts=100.0e-6;

/* Condensador superior */

real_T C1=750.0e-6;

/* Condensador inferior */

real_T C2=750.0e-6;

/* Desequilibri de pes */

real_T lambdaDC=1.0;

/* Declaració de variables*/

/* Entrades */

real_T irefkre;

real_T irefkim;

real_T ikre;

real_T ikim;

real_T Vc1;

real_T Vc2;

real_T ekre;

real_T ekim;

/* Extrapolació */

real_T irefk1re;

real_T irefk1im;

/* Predicció de corrent i(k+1) */

real_T ik1re;

real_T ik1im;

/* Corrents a cada fase a, b i c */

real_T ia;

real_T ib;

real_T ic;
```

```
/* Corrents a cada branca p, o i n */
real_T ip;
real_T i0;
real_T in;

/* Corrent als condensadors */
real_T ic1;
real_T ic2;

/* Tensions als condensadors */
real_T Vc11;
real_T Vc21;

/* Funció de cost */
real_T g;

/* Error òptim */
real_T gop;

/* Contador */
int_T contador;

/* Estats redundants */
int_T n;

/* Vector òptim */
int_T iop;

/* Estat òptim */
int_T xop;

/* Vectors de tensió */
float SV[19][2]=
    {
        {0.0,0.0}, /*V0*/
        {33.3,0.0}, /*V1*/
        {16.67,28.87}, /*V2*/
        {-16.67,28.87}, /*V3*/
        {-33.33,0.0}, /*V4*/
```

```
{-16.67,-28.87}, /*V5*/
{16.67,-28.87}, /*V6*/
{66.67,0.0}, /*V7*/
{49.99,28.87}, /*V8*/
{33.33,57.73}, /*V9*/
{0.0,57.73}, /*V10*/
{-33.33,57.73}, /*V11*/
{-49.99,28.87}, /*V12*/
{-66.67,0.0}, /*V13*/
{-49.99,-28.87}, /*V14*/
{-33.33,-57.73}, /*V15*/
{0.0,-57.73}, /*V16*/
{33.33,-57.73}, /*V17*/
{49.99,-28.87}, /*V18*/
};

/* Estats */
int ST[27][3]=
{
{-1,-1,-1}, /*estat_0*/
{0,0,0}, /*estat_1*/
{1,1,1}, /*estat_2*/
{1,0,0}, /*estat_3*/
{0,-1,-1}, /*estat_4*/
{1,1,0}, /*estat_5*/
{0,0,-1}, /*estat_6*/
{0,1,0}, /*estat_7*/
{-1,0,-1}, /*estat_8*/
{0,1,1}, /*estat_9*/
{-1,0,0}, /*estat_10*/
{0,0,1}, /*estat_11*/
{-1,-1,0}, /*estat_12*/
```

```
{1,0,1}, /*estat_13*/  
{0,-1,0}, /*estat_14*/  
{1,-1,-1}, /*estat_15*/  
{1,0,-1}, /*estat_16*/  
{1,1,-1}, /*estat_17*/  
{0,1,-1}, /*estat_18*/  
{-1,1,-1}, /*estat_19*/  
{-1,1,0}, /*estat_20*/  
{-1,1,1}, /*estat_21*/  
{-1,0,1}, /*estat_22*/  
{-1,-1,1}, /*estat_23*/  
{0,-1,1}, /*estat_24*/  
{1,-1,1}, /*estat_25*/  
{1,-1,0}, /*estat_26*/  
};
```

```
/* Adquisició de dades d'entrades mitjançant punters */
```

```
InputRealPtrsType uPtrs0 = ssGetInputPortRealSignalPtrs(S, 0);  
InputRealPtrsType uPtrs1 = ssGetInputPortRealSignalPtrs(S, 1);  
InputRealPtrsType uPtrs2 = ssGetInputPortRealSignalPtrs(S, 2);  
InputRealPtrsType uPtrs3 = ssGetInputPortRealSignalPtrs(S, 3);  
InputRealPtrsType uPtrs4 = ssGetInputPortRealSignalPtrs(S, 4);  
InputRealPtrsType uPtrs5 = ssGetInputPortRealSignalPtrs(S, 5);  
InputRealPtrsType uPtrs6 = ssGetInputPortRealSignalPtrs(S, 6);  
InputRealPtrsType uPtrs7 = ssGetInputPortRealSignalPtrs(S, 7);
```

```
/* Adquisició de dades de sortides mitjançant punters */
```

```
real_T *y0 = ssGetOutputPortRealSignal(S,0);  
real_T *y1 = ssGetOutputPortRealSignal(S,1);  
real_T *y2 = ssGetOutputPortRealSignal(S,2);
```

```
/* Adjudicació de les variables de dades d'entrades a cada punter */
```

```
irefkr= *uPtrs0[0];  
irefkim= *uPtrs1[0];
```

```

ikre= *uPtrs2[0];

ikim= *uPtrs3[0];

Vc1= *uPtrs4[0];

Vc2= *uPtrs5[0];

ekre= *uPtrs6[0];

ekim= *uPtrs7[0];

/* Funció */

/* Inicialització */

contador=0;

gop=10.0e+30;

/* Referència d'extrapolació(simplificació) */

irefk1re=irefkre;

irefk1im=irefkim;

/* Avaluació i optimització de g */

{int_T i;

for (i=0; i<19; i++){

    if (i==0){

        n=3;}

    if ((i<7)&&(i>0)){

        n=2;}

    if (i>=7){

        n=1;}

/* Predicció de corrents */

ik1re=(L/(R*Ts+L))*ikre+(Ts/(R*Ts+L))*(SV[i][0])-(Ts/(R*Ts+L))*ekre;

ik1im=(L/(R*Ts+L))*ikim+(Ts/(R*Ts+L))*(SV[i][1])-(Ts/(R*Ts+L))*ekim;

/* Predicció dels corrents a les fases */

ia=ik1re;

ib=-0.5*ik1re+(sqrt(3.0)/2.0)*ik1im;

ic=-0.5*ik1re-(sqrt(3.0)/2.0)*ik1im;

{int_T j;

for (j=0; j<n; j++){

```



```
if (i==0){ /*V0*/
    i0=0;ip=0;in=0;}
if (contador==3){ /*V1*/
    ip=ia;i0=ic+ib;in=0;}
if (contador==4){ /*V1*/
    i0=ia;in=ib+ic;ip=0;}
if (contador==5){ /*V2*/
    ip=ia+ib;i0=ic;in=0;}
if (contador==6){ /*V2*/
    ip=0;i0=ia+ib;in=ic;}
if (contador==7){ /*V3*/
    ip=ib;i0=ia+ic;in=0;}
if (contador==8){ /*V3*/
    ip=0;i0=ib;in=ia+ic;}
if (contador==9){ /*V4*/
    ip=ib+ic;i0=ia;in=0;}
if (contador==10){ /*V4*/
    ip=0;i0=ib+ic;in=ia;}
if (contador==11){ /*V5*/
    ip=ic;i0=ia+ib;in=0;}
if (contador==12){ /*V5*/
    ip=0;i0=ic;in=ia+ib;}
if (contador==13){ /*V6*/
    ip=ia+ic;i0=ib;in=0;}
if (contador==14){ /*V6*/
    ip=0;i0=ia+ic;in=ib;}
if (contador==15){ /*V7*/
    ip=ia;i0=0;in=ib+ic;}
if (contador==16){ /*V8*/
    ip=ia;i0=ib;in=ic;}
if (contador==17){ /*V9*/
```

```

    ip=ia+ib;i0=0;in=ic;}

if (contador==18){          /*V10*/
    ip=ib;i0=ia;in=ic;}

if (contador==19){          /*V11*/
    ip=ib;i0=0;in=ia+ic;}

if (contador==20){          /*V12*/
    ip=ib;i0=ic;in=ia;}

if (contador==21){          /*V13*/
    ip=ib+ic;i0=0;in=ia;}

if (contador==22){          /*V14*/
    ip=ic;i0=ib;in=ia;}

if (contador==23){          /*V15*/
    ip=ic;i0=0;in=ia+ib;}

if (contador==24){          /*V16*/
    ip=ic;i0=ia;in=ib;}

if (contador==25){          /*V17*/
    ip=ia+ic;i0=0;in=ib;}

if (contador==26){          /*V18*/
    ip=ia;i0=ic;in=ib;}

    ic1=-ip;
    ic2=in;

    Vc11=Vc1+(1.0/C1)*ic1*Ts;
    Vc21=Vc2+(1.0/C2)*ic2*Ts;

/* Avaluació de la funció de qualitat */
g=fabs((irefk1re)-(ik1re))+fabs((irefk1im)-(ik1im))+lambdaDC*fabs(Vc11-Vc21);

    if (g<gop){
        iop=i;
        xop=contador;
        gop=g;
    }

    contador=contador+1;

```

```
    }
  }
}
}
/* La sortida és l'estat de commutació que minimitza g */
*y0=ST[xop][0];
*y1=ST[xop][1];
*y2=ST[xop][2];
}
/*Funció principal del bloc finalitzada*/
/* Finalització del programa */
static void mdlTerminate(SimStruct *S)
{
}

#ifdef MATLAB_MEX_FILE
#include "simulink.c"
#else
#include "cg_sfun.h"
#endif
```

4.3.2. Programa detallat dos passos.

```
/* Capçalera */
/* Nom de l'arxiu */
#define S_FUNCTION_NAME dos_passos
#define S_FUNCTION_LEVEL 2
/* Llibreries a utilitzar */
#include "simstruc.h"
#include <math.h>
#include <assert.h>
```

```
#include <stdlib.h>

/* Variables globals (memòria) */

/* Tensions als condensadors */

real_T vxa;

real_T vxb;

real_T vxm1a;

real_T vxm1b;

/* Estats de commutació */

int_T Sm1a;

int_T Sm1b;

    int_T Sm1c;

    int_T Saop;

    int_T Sbop;

    int_T Scop;

/* Definició d'entrades i sortides */

static void mdlInitializeSizes(SimStruct *S)

{

    ssSetNumSFcnParams(S, 0);

    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount (S)) {

        return;

    }

    /* 8 entrades */

    if (!ssSetNumInputPorts(S,8)) return;

    {int_T i;

        for (i=0; i<8; i++)

        {

            ssSetInputPortWidth(S, i, 1);

            ssSetInputPortDirectFeedThrough (S, i, 1);

        }

    }

    /* 3 sortides */
```

```
if (!ssSetNumOutputPorts(S, 3)) return;

{ int_T i;

  for (i=0; i<3; i++)

  {

    ssSetOutputPortWidth(S, i, 1);

  }

}

ssSetNumSampleTimes(S, 1);

ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE
SS_OPTION_USE_TLC_WITH_ACCELERATOR |SS_OPTION_PLACE_ASAP);

}

/* Inicialització del temps de mostreig */
static void mdlInitializeSampleTimes(SimStruct *S)
{

  ssSetSampleTime (S, 0, 100.0e-6);

  ssSetOffsetTime (S, 0, 0.0);

}

/* Funció principal del bloc */
static void mdlOutputs(SimStruct *S, int_T tid)
{

  int_T width = ssGetOutputPortWidth(S,0);

  /* Constants */

  /* Tensió */

  real_T Vdc=100.0;

  /* Filtre */

  real_T L=5.0e-3;

  /* Resistència de la bobina */

  real_T R=10.0;

  /* Temps de mostreig */

  real_T Ts=100.0e-6;

  /* Condensador superior */
```

```
real_T C1=750.0e-6;

/* Condensador inferior */

real_T C2=750.0e-6;

/* Desequilibri de pes */

real_T lambdaDC=1.0;

/* Declaració de variables*/

/* Entrades */

real_T irefka;

real_T irefkb;

real_T ika;

    real_T ikb;

    real_T vpk;

    real_T vnk;

    real_T eka;

    real_T ekb;

/* Extrapolació */

real_T irefk1a;

real_T irefk1b;

/* Predicció de corrent a l'instant k+1 */

real_T ik1a;

real_T ik1b;

/* Predicció dels corrents a les fases per k+1 */

real_T iak1;

real_T ibk1;

real_T ick1;

real_T iok1;

/* Tensions als condensadors a l'instant k+1 */

real_T vpk1;

real_T vnk1;

/* Vectors de tensió */

real_T va;
```

```
real_T vb;

real_T vc;

/* Predicció de corrent a l'instant k+2 */

real_T ik2a;

real_T ik2b;

/* Predicció dels corrents a les fases per k+2 */

real_T iak2;

real_T ibk2;

real_T ick2;

real_T iok2;

/* Tensions als condensadors a l'instant k+2 */

real_T vpk2;

real_T vnk2;

/* Funció de cost*/

real_T g;

/* Error òptim*/

real_T gop;

/* Estats de commutació */

real_T iok1a;

real_T iok1b;

real_T iok1c;

real_T iok2a;

real_T iok2b;

real_T iok2c;

/* Adquisició de dades d'entrades mitjançant punters */

InputRealPtrsType uPtrs0 = ssGetInputPortRealSignalPtrs(S, 0);

InputRealPtrsType uPtrs1 = ssGetInputPortRealSignalPtrs(S, 1);

InputRealPtrsType uPtrs2 = ssGetInputPortRealSignalPtrs(S, 2);

InputRealPtrsType uPtrs3 = ssGetInputPortRealSignalPtrs(S, 3);

InputRealPtrsType uPtrs4 = ssGetInputPortRealSignalPtrs(S, 4);

InputRealPtrsType uPtrs5 = ssGetInputPortRealSignalPtrs(S, 5);
```

```
InputRealPtrsType uPtrs6 = ssGetInputPortRealSignalPtrs(S, 6);
```

```
InputRealPtrsType uPtrs7 = ssGetInputPortRealSignalPtrs(S, 7);
```

```
/* Adquisició de dades de sortides mitjançant punters */
```

```
real_T *y0 = ssGetOutputPortRealSignal(S,0);
```

```
real_T *y1 = ssGetOutputPortRealSignal(S,1);
```

```
real_T *y2 = ssGetOutputPortRealSignal(S,2);
```

```
/* Adjudicació de les variables de dades d'entrades a cada punter */
```

```
irefka= *uPtrs0[0];
```

```
irefkb= *uPtrs1[0];
```

```
ika= *uPtrs2[0];
```

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

```
vpk= *uPtrs4[0];
```

```
vnk= *uPtrs5[0];
```

```
eka= *uPtrs6[0];
```

```
ekb= *uPtrs7[0];
```

```
/* Funció */
```

```
/* Inicialització */
```

```
gop=10.0e+30;
```

```
/* Corrent de referència (simplificació de l'extrapolació) */
```

```
irefk1a=irefka;
```

```
irefk1b=irefkb;
```

```
/* Predicció de corrent per k+1 */
```

```
ik1a=(1.0-(R*Ts/L))*ika+(Ts/L)*(vxm1a-eka);
```

```
ik1b=(1.0-(R*Ts/L))*ikb+(Ts/L)*(vxm1b-ekb);
```

```
/* Predicció dels corrents a les fases per k+1 */
```

```
iak1=ik1a;
```

```
ibk1=-0.5*ik1a+(sqrt(3.0)/2.0)*ik1b;
```

```
ick1=-0.5*ik1a-(sqrt(3.0)/2.0)*ik1b;
```

```
/* Predicció als circuits intermitjos per k+2 */
```

```
iok1a=0;
```

```
iok1b=0;
```



```

iok1c=0;

if(Sm1a==0){
    iok1a=iak1;}

if(Sm1b==0){
    iok1b=ibk1;}

if(Sm1c==0){
    iok1c=ick1;}

iok1=(iok1a+iok1b+iok1c);

vpk1=vpk+(1.0/(2.0*C1))*iok1*Ts;

vnk1=vnk+(1.0/(2.0*C2))*iok1*Ts;

/* Avaluació i optimització de g sobre els 27 estats de commutació */

{int_T Sa;
for (Sa=-1; Sa<2; Sa++){
    {int_T Sb;
for (Sb=-1; Sb<2; Sb++){
        {int_T Sc;
for (Sc=-1; Sc<2; Sc++){
            /* Vectors */
            va=Sa*Vdc;
            vb=Sb*Vdc;
            vc=Sc*Vdc;

            /* vx=(valpha)+(vbeta)*/
            vxa=(va-0.5*vb-0.5*vc)/3.0;
            vxb=((sqrt(3.0)/2.0)*vb-(sqrt(3.0)/2.0)*vc)/3.0;

            /* Predicció del corrent per k+2 */
            ik2a=(1.0-(R*Ts/L))*ik1a+(Ts/L)*(vxa-eka);
            ik2b=(1.0-(R*Ts/L))*ik1b+(Ts/L)*(vxb-ekb);

/* Predicció dels corrents a les fases per k+2 */
            iak2=ik2a;

            ibk2=-0.5*ik2a+(sqrt(3.0)/2.0)*ik2b;

            ick2=-0.5*ik2a-(sqrt(3.0)/2.0)*ik2b;

```



```
Sm1a=Saop;

Sm1b=Sbop;

Sm1c=Scop;

/* La sortida és l'estat de commutació que minimitza g */

*y0=Saop;

*y1=Sbop;

*y2=Scop;

}

/* Funció principal del bloc finalitzada */

/* Finalització del programa */

static void mdlTerminate(SimStruct *S)

{

}

#ifdef MATLAB_MEX_FILE

#include "simulink.c"

#else

#include "cg_sfun.h"

#endif
```


5. Resultats de les simulacions del programa amb llenguatge de programació C.

5.1. Simulacions del programa un pas.

A continuació es mostren les formes d'ona de les simulacions en funció del temps dels senyals que s'han considerat rellevants pel nostre circuit. A la Fig. 5.2. es mostra el seguiment dels corrents a les fases ia, ib i ic (A). A la Fig. 5.3. es mostra el comportament de la tensió de desequilibri vo (V).

A la Fig. 5.1. s'observen les formes d'ona dels corrents de referència de les tres fases del nostre convertidor.

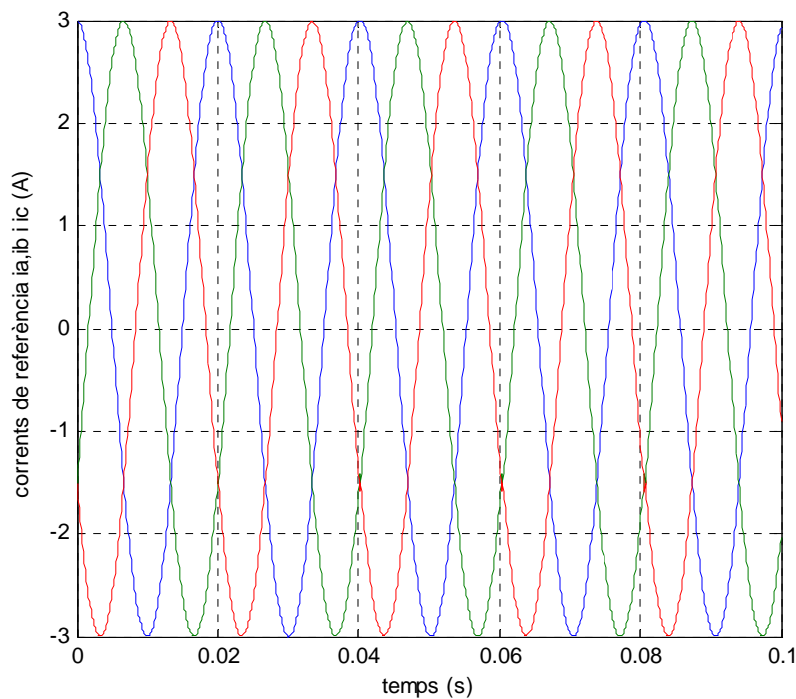
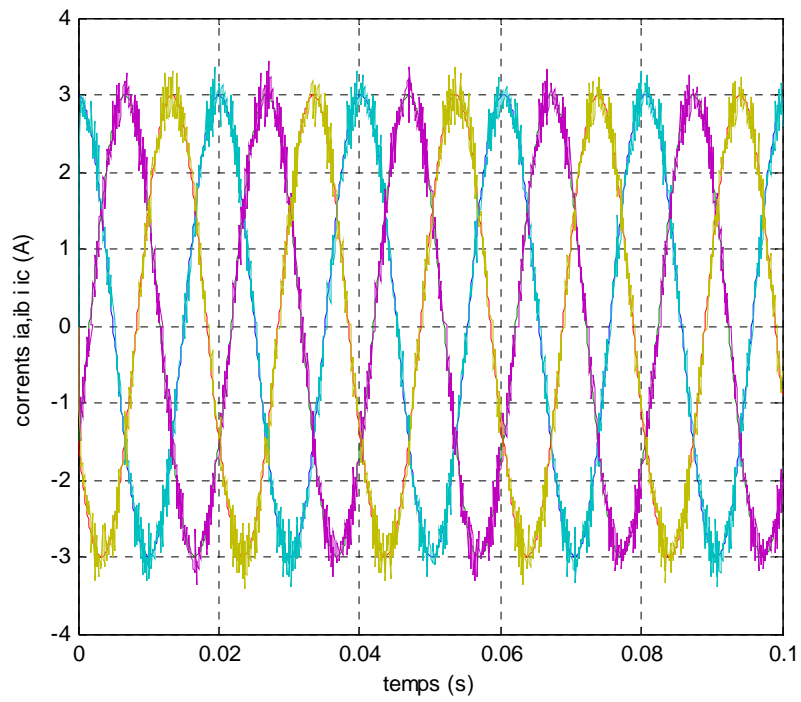
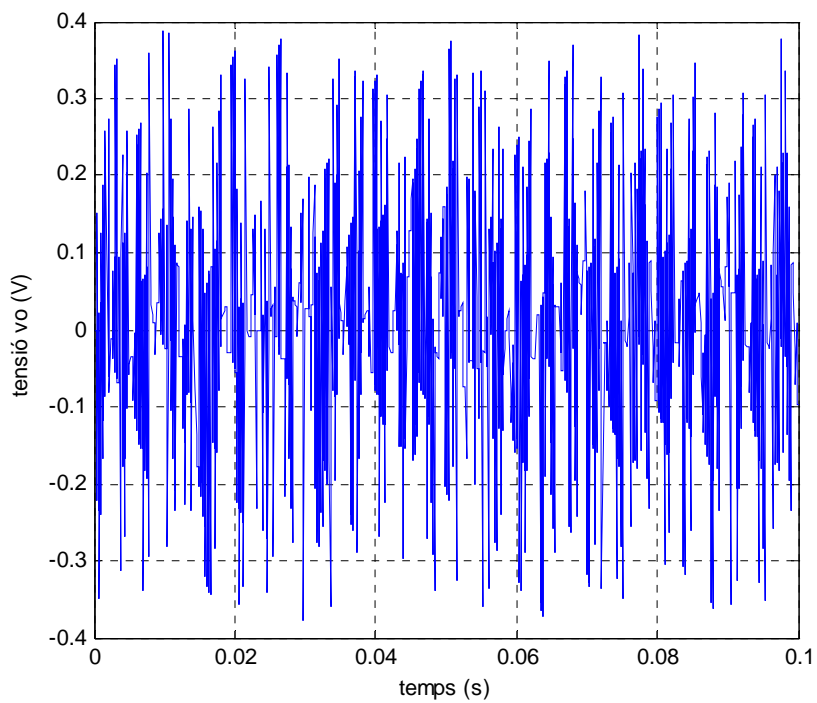


Fig. 5.1. Corrents de referència ia, ib i ic (A).

Fig. 5.2. Corrents i_a , i_b i i_c (A).Fig. 5.3. Tensió de desequilibri v_o (V).

Com es pot comprovar a la Fig. 5.2. el nostre programa fa un seguiment correcte dels corrents a les fases del nostre convertidor.

El comportament de la tensió de desequilibri no es veu afectada, no altera la seva evolució tal com es mostra a la Fig. 5.3. El bus de contínua es manté equilibrat en tot moment. En tractar-se d'una topologia NPC, s'ha de controlar l'equilibri de les capacitats del bus de contínua (v_o).

A la Fig. 5.4. , Fig. 5.5. i 5.6 s'observa com s'ha provocat un transitori entre 30 i 70 ms, i el control predictiu del nostre programa segueix fent un seguiment correcte de les fases del nostre convertidor. Com s'observa a la Fig. 5.5. s'ha fet un zoom en el moment que s'ha provocat el transitori, per comprovar amb més exactitud el seguiment correcte dels corrents a les tres fases del nostre convertidor.

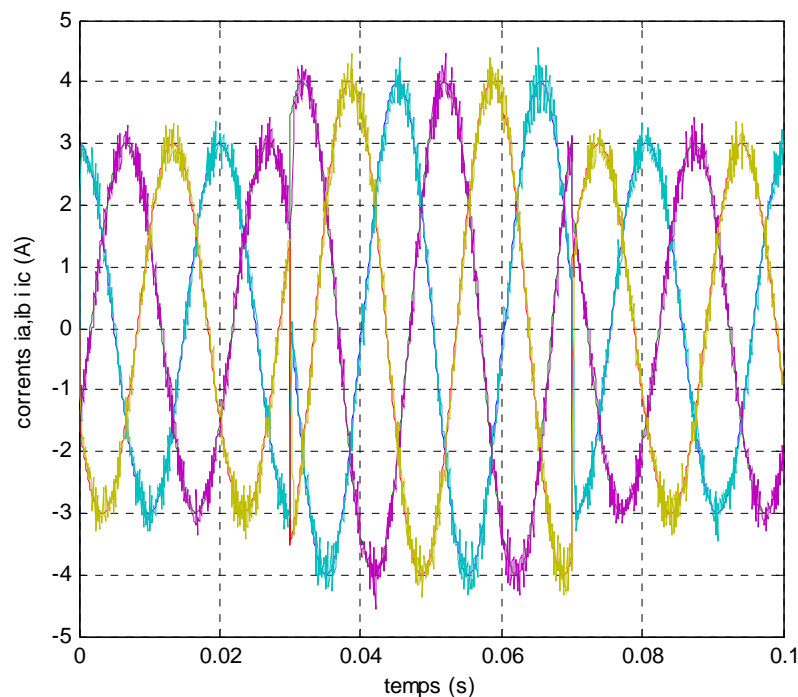
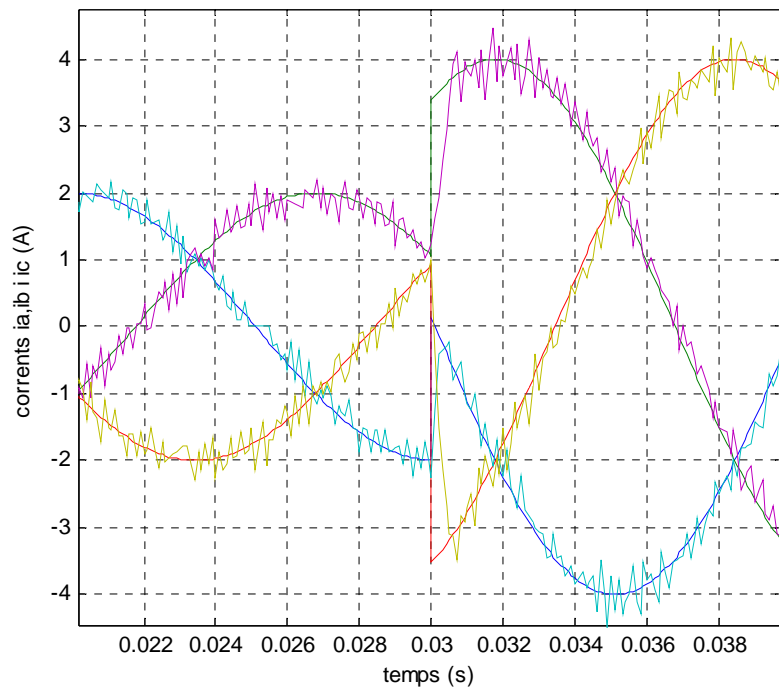
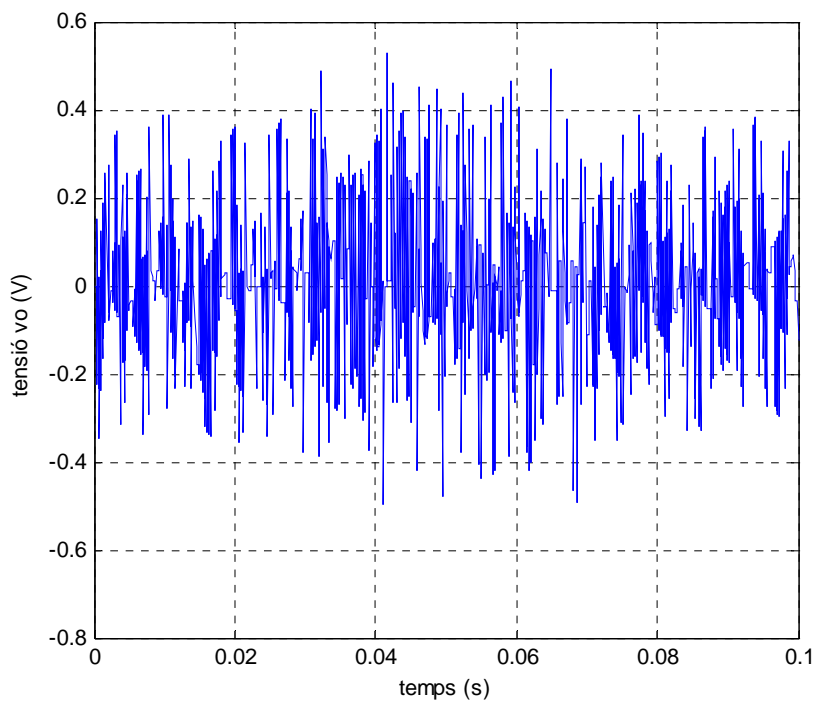


Fig. 5.4. Corrents ia, ib i ic (A).

Fig. 5.5. Corrents i_a , i_b i i_c (A).Fig. 5.6. Tensió de desequilibri v_o (V).

5.2. Simulacions del programa dos passos.

A continuació es mostren les formes d'ona de les simulacions en funció del temps dels senyals que s'han considerat rellevants pel nostre circuit. A la Fig. 5.8. es mostra el seguiment dels corrents a les fases ia, ib i ic (A). A la Fig. 5.9. es mostra el comportament de la tensió de desequilibri vo (V).

A la Fig. 5.7. s'observen les formes d'ona dels corrents de referència de les tres fases del nostre convertidor.

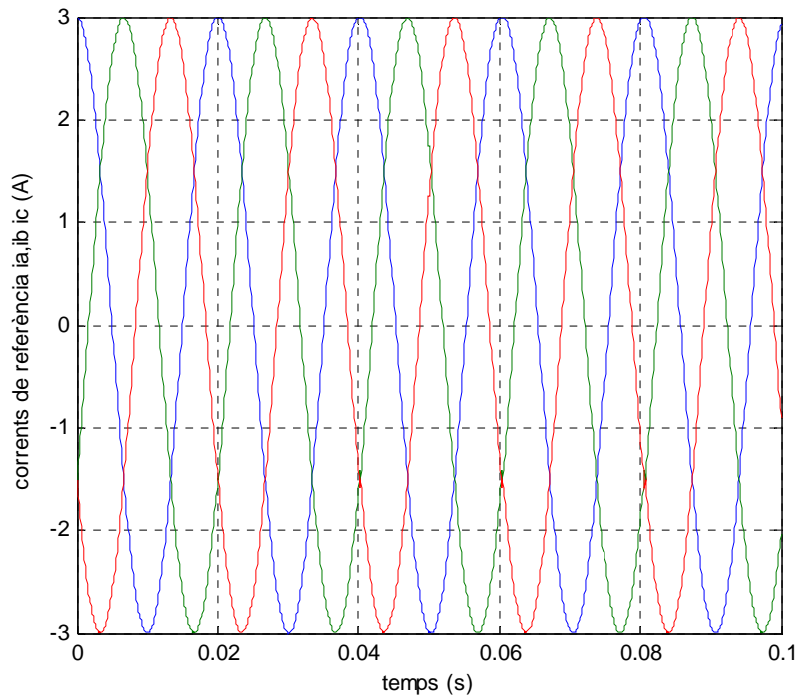
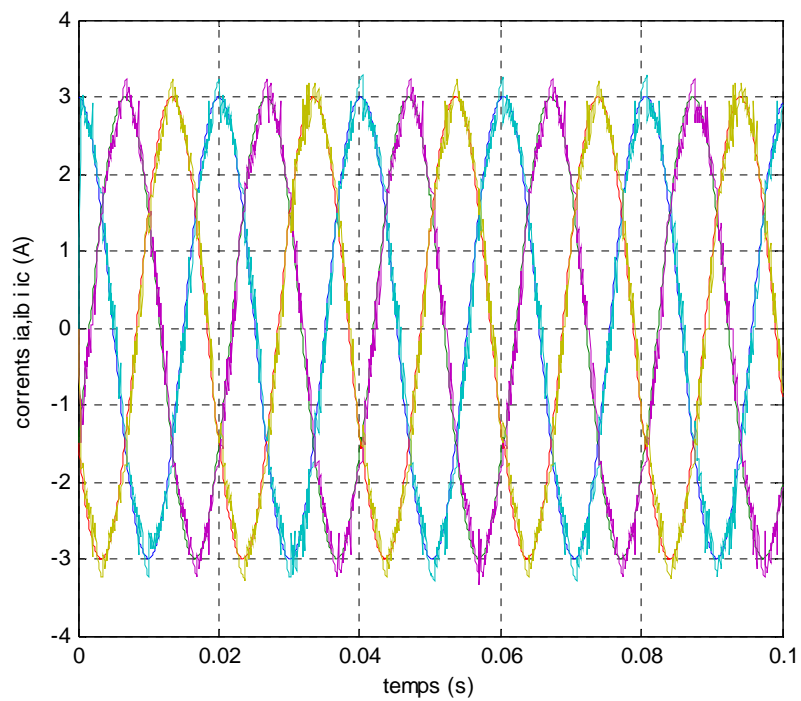
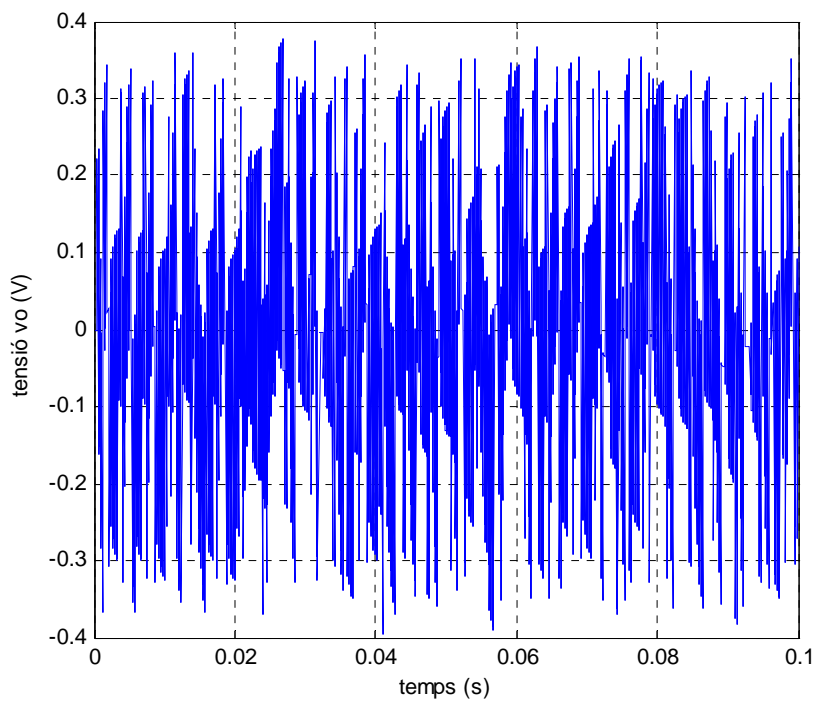


Fig. 5.7. Corrents de referència ia, ib i ic (A).

Fig. 5.8. Corrents i_a , i_b i i_c (A).Fig. 5.9. Tensió de desequilibri v_o (V).

Com es pot comprovar a la Fig. 5.8. el nostre programa fa un seguiment correcte dels corrents a les fases del nostre convertidor.

El comportament de la tensió de desequilibri no es veu afectada, no altera la seva evolució tal com es mostra a la Fig. 5.9. El bus de contínua es manté equilibrat en tot moment. En tractar-se d'una topologia NPC, s'ha de controlar l'equilibri de les capacitats del bus de contínua (v_o).

A la Fig. 5.10., Fig. 5.11. i Fig. 5.12. s'observa com s'ha provocat un transitori entre 30 i 70 ms, i el control predictiu del nostre programa segueix fent un seguiment correcte de les fases del nostre convertidor. Com s'observa a la Fig. 5.11. s'ha fet un zoom en el moment en què s'ha provocat el transitori, per comprovar amb més exactitud el seguiment correcte dels corrents a les tres fases del nostre convertidor. Clarament, es pot observar que l'esmoreïment és menor que el del programa un pas, per això obtenim una gran millora.

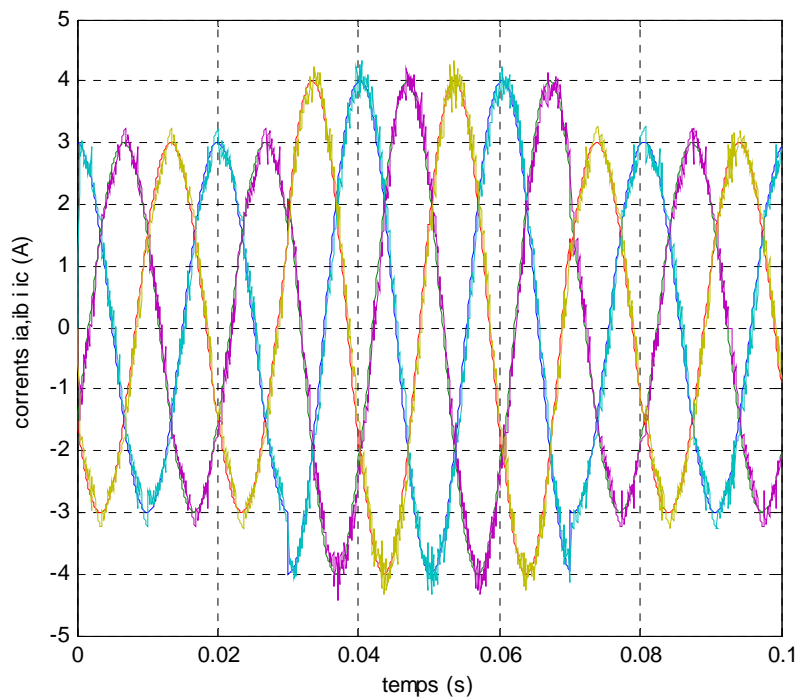
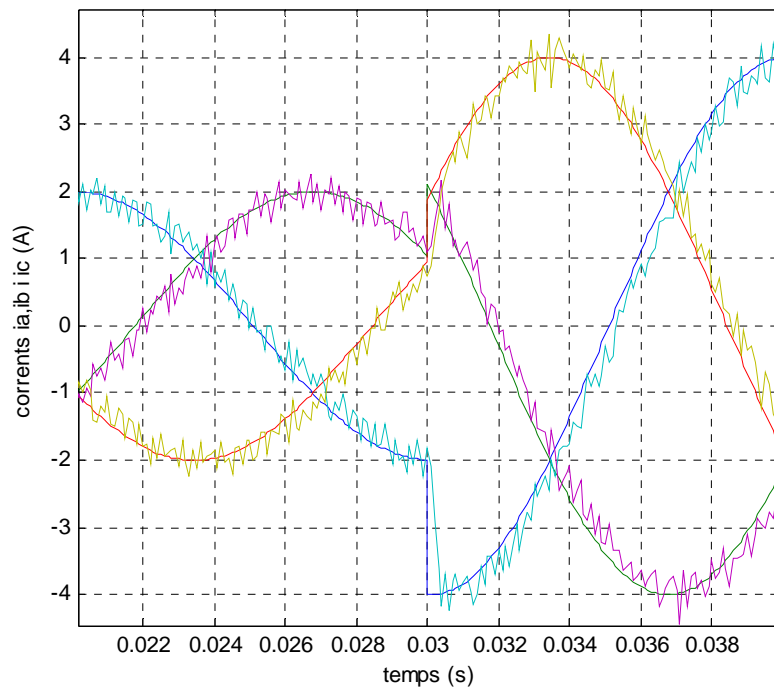
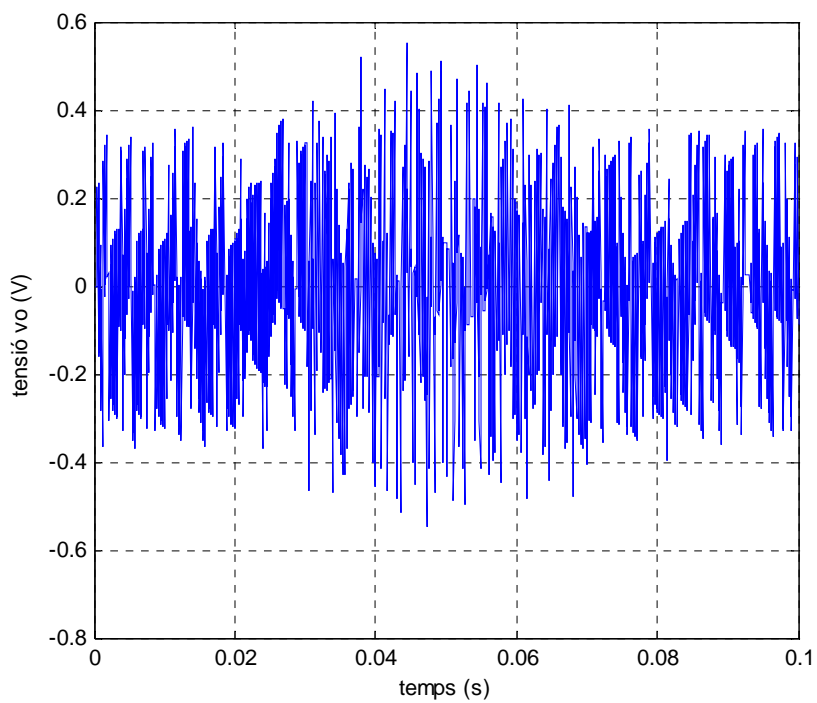


Fig. 5.10. Corrents i_a , i_b i i_c (A).

Fig. 5.11. Corrents i_a , i_b i i_c (A).Fig. 5.12. Tensió de desequilibri v_o (V).

6. Conclusions.

En el present projecte s'han realitzat dos programes en llenguatge C que fan el control predictiu d'un inversor trifàsic amb topologia NPC. La conclusió principal a la qual s'ha arribat és el correcte funcionament dels dos programes, tenint en compte els resultats obtinguts a les simulacions.

Després d'observar les simulacions dels dos programes (un pas i dos passos), s'observa i es dedueix que el programa dos passos millora el programa un pas. Fa un millor seguiment dels corrents del convertidor i de la tensió de desequilibri, que són els senyals rellevants pel control predictiu de l'inversor trifàsic amb topologia NPC. Tal com s'ha pogut observar, l'esmoreïment del programa dos passos és menor que el del programa un pas.

Els dos programes s'han preparat per implementar-los en el sistema experimental, i poder augmentar la velocitat d'execució en el control predictiu.

La realització d'aquest projecte ha aportat els següents aspectes a la formació acadèmica del projectista:

- S'ha aprofundit en l'ús del programa Matlab mitjançant l'entorn gràfic Simulink.
- S'ha après a fer servir el bloc S-Function de la llibreria del Simulink, per poder programar en llenguatge de programació C.
- També s'han après particularitats del llenguatge de programació C que s'han de tenir en compte a l'hora de programar.
- S'ha estudiat el convertidor de 3 nivells amb topologia NPC i el seu control predictiu corresponent.

7. Referències.

- [1] ALEPUZ MENÉNDEZ, Salvador. *Aportación al Control del Convertidor CC/CA de Tres Niveles*. Barcelona: Universitat Politècnica de Catalunya, 2004.
- [2] ALEPUZ MENÉNDEZ, Salvador. *Apunts convertidors CC/CA*. Barcelona: [s. n.], 2007.
- [3] ALEPUZ, S.; BUSQUETS-MONGE, J.; BORDONAU, J.; CORTÉS, P.; RODRÍGUEZ, J.; VARGAS, R. *Predictive Current Control of Grid-Connected Neutral-Point-Clamped Converters to meet Low Voltage Ride-Through Requirements*. Rodes, 2008, p. 2423-2428.
- [4] CORTÉS, P. *Delay compensation in model predictive current control of a three-phase inverter*. Xile: Universidad Técnica Federico Santa María, 2012.
- [5] RODRÍGUEZ, J.; CORTÉS, P. *Model predictive control: a simple and powerful method to control power converters*. Xile: Universidad Técnica Federico Santa María, 2011.