

國立交通大學

電機學院 電機與控制學程

碩士論文

無橋式功因校正昇壓型整流器
之無電流感測控制

Current Sensorless Control for
Bridgeless PFC Boost Rectifier

研 究 生：鍾肇強

指 導 教 授：陳鴻祺博士

中 華 民 國 一 百 零 一 年 七 月

無橋式功因校正昇壓型整流器

之無電流感測控制

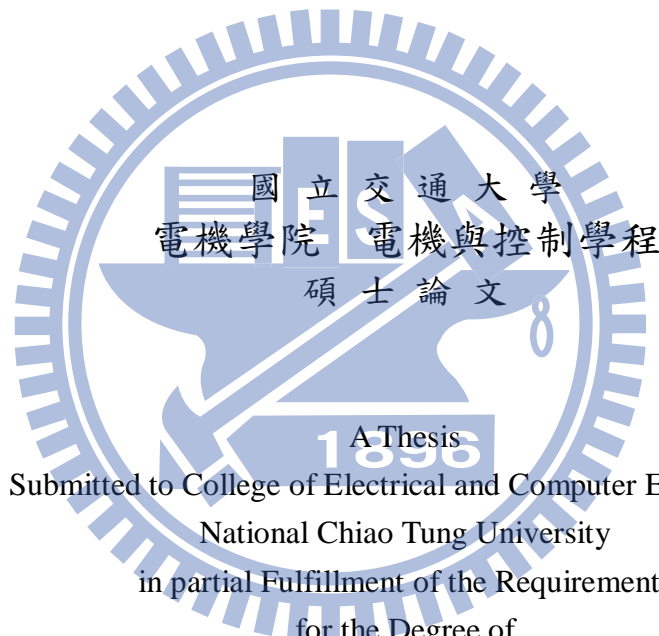
Current Sensorless Control for Bridgeless PFC Boost Rectifier

研 究 生：鍾肇強

Student : Chao-Chiang Chung

指 導 教 授：陳鴻祺

Advisor : Hung-Chi Chen



Submitted to College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master of Science

in

Electrical and Control Engineering

July 2012

Hsinchu, Taiwan, Republic of China

中華民國 一 百 零 一 年 七 月

無橋式功因校正昇壓型整流器 之無電流感測控制

學生：鍾肇強

指導教授：陳鴻祺 博士

國立交通大學 電機學院 電機與控制學程 碩士班

摘 要

本篇論文是採用無橋式功率校正昇壓型整流器，實現無電流感測迴路架構。在控制方法上，不同於傳統多迴路電流感測控制必須偵測輸入電壓、輸出電壓以及電感電流，本論文提出一只感測輸入電壓及輸出電壓的控制，在不迴授輸入電流下，可以不用考慮電流取樣及偵測，減少控制架構的複雜度，並利用演算的方式控制輸入電流使其達到功因校正且在直流鏈亦能維持一穩定直流的電壓，並仍能達到國際間對電流諧波的所設立的規範。

無電流感測控制架構控制的目的為控制電感電流為弦波函數，達到輸入電流為弦波且同相，且當輸入電壓高達400Hz，依然可以達到功率因素校正的效果。

Current Sensorless Control for Bridgeless PFC Boost Rectifier

Student : Chao-Chiang Chung

Advisor : Hung-Chi Chen

Degree Program of Electrical and Computer Engineering

National Chiao Tung University

Abstract

The conventional multi-loop control senses the input voltage, the output voltage and the inductor current, in order to yield in-phase sinusoidal current and regulate the output voltage. In this paper, the current sensorless control for bridgeless PFC boost rectifier is proposed where only the input voltage and the output voltage are sensed. The current sensorless control has one voltage loop, which simplifies the control structure, and reduce the number of sensors.

In this thesis, the input current waveform are analyzed and the circuit behavior is modeled with consideration of both the inductor resistance and the switch conduction voltage. From the simulation and experiment results, the measured sinusoidal current is in-phase with the input voltage evenwhen the line frequency is high to 400 Hz.

誌 謝

本論文得以順利完成首先感謝我的指導教授陳鴻棋老師，在撰寫本論文中，經常給予我方向和細心的指導，讓我能夠順利的進行研究和論文的修改，在此對老師獻上由衷的感謝。再者，感謝口試委員胡竹生博士和廖益宏博士給我論文提出寶貴的意見，使論文更加的完整。

在專班研究所修課及實驗時，每當來實驗室做實驗時都是戰戰兢兢，但實驗室裡的氣氛相當的融洽與和諧，讓我的緊張的心情得到了紓緩，感謝學姊如璇，同學任浩、暉舜等，給我於修課上面問題得以討論和解答；也感謝學弟宏和、威諭、振宇、智豪、文仁等等，在實驗上面的方向指導和幫助。再者，感謝我專班生活認識的同學和朋友，總是給予熱情得鼓勵和打氣。

最後感謝家人的支持，不論我做任何決定都給予鼓勵，並讓我專班研究所的生活可以無後顧之憂的完成學業。



目錄

摘要	I
Abstract	II
誌謝	III
第一章 緒論	1
1.1 研究動機	1
1.2 諧波電流	2
1.3 諧波規範	3
1.4 本文架構	5
第二章 無橋式 PFC 電路	6
2.1 無橋式 PFC 電路介紹	6
2.2 無橋式 PFC 的工作原理及特點	7
2.3 傳統昇壓型 PFC 和無橋式 PFC 效率簡析	8
2.4 無橋式 PFC 電路的變化型	10
第三章 無橋式 PFC 之無電流感測控制模型	11
3.1 電路模型假設	11
3.2 開關訊號產生方式	12
3.3 電路推導	13
3.3.1 輸入電壓正半週期時($sign(v_s) > 0$)	13
3.3.2 輸入電壓負半週期時($sign(v_s) < 0$)	14
3.3.3 等效模型	15
3.4 無電流感測控制架構	16
3.5 無橋式 PFC 電路之無電流感測控制推導	18
3.6 無電流感測控制的實現	19
第四章 模擬驗證	20
4.1 模擬電路及元件參數	21
4.2 無電流感測控制之模擬	22
4.2.1 穩態模擬	23

4.2.2 暫態模擬	34
第五章 電路實作	36
5.1 現場可規劃邏輯陣列	36
5.2 實作電路組成.....	38
5.2.1 數位類比轉換電路(A/D Circuit & D/A Circuit).....	40
5.2.2 開關驅動電路(Gate Driving Circuit).....	41
5.3 控制器合成.....	42
5.4 無電流感測之實作	43
5.4.1 頻率 60Hz 輸入電壓下之穩態實作	44
5.4.2 頻率 400Hz 輸入電壓下之穩態實作	46
5.4.3 頻率 60Hz 輸入電壓下之暫態實作	48
5.4.4 頻率 400Hz 輸入電壓下之暫態實作	49
5.5 諧波規範比較.....	50
5.6 實驗改善方向.....	57
第六章 結論	58
參考文獻	59



圖目錄

圖 1.1 傳統橋式整流的轉換器和帶有 PFC 的變換器比較.....	1
(a)傳統橋式整流的轉換器	(b)帶有 PFC 的轉換器..... 1
圖 1.2 電流失真.....	2
圖 2.1 PFC 電路圖.....	7
(a)傳統昇壓型 PFC 電路圖	(b) 無橋式 PFC 電路圖..... 7
(c)全橋式 PFC 電路圖	(d) Totem-Pole PFC 電路圖..... 7
圖 2.2 無橋式 PFC 電路動作狀態.....	8
(a)輸入電壓正半週期	(b)輸入電壓負半週期..... 8
圖 2.3 正半週期內二種電路架構之電流路徑.....	9
(a)傳統昇壓型 PFC	(b)無橋式 PFC..... 9
圖 2.4 增加二個二極體之無橋式 PFC 電路.....	10
圖 3.1 無橋式 PFC 主要電路及控制方塊圖.....	11
圖 3.2 開關訊號邏輯圖.....	12
圖 3.3 輸入電壓為正半週期時電路示意圖.....	13
(a)PWM 訊號 $d(t)$ 為 High	(b) PWM 訊號 $d(t)$ 為 Low..... 13
圖 3.4 輸入電壓為負半週期時電路示意圖.....	14
(a)PWM 訊號 $d(t)$ 為 High	(b) PWM 訊號 $d(t)$ 為 Low..... 14
圖 3.5 等效電路圖.....	15
圖 3.6 無電流感測控制電路方塊圖.....	16
圖 3.7 控制迴路參考波形.....	17
圖 4.1 PSIM 功能模組概要圖.....	20
圖 4.2 PSIM 內無橋式 PFC 無電流感測電路模擬圖.....	22
圖 4.3 60Hz 輸入電壓下穩態響應及控制器內部訊號.....	24
圖 4.4 400Hz 輸入電壓下穩態響應及控制器內部訊號.....	25
圖 4.5 200W 輸出 60Hz 模擬輸入電壓及輸入電流波形.....	26
圖 4.6 400W 輸出 60Hz 模擬輸入電壓及輸入電流波形.....	26
圖 4.7 600W 輸出 60Hz 模擬輸入電壓及輸入電流波形.....	26

圖 4.8 200W 輸出 400Hz 模擬輸入電壓及輸入電流波形.....	30
圖 4.9 400W 輸出 400Hz 模擬輸入電壓及輸入電流波形.....	30
圖 4.10 600W 輸出 400Hz 模擬輸入電壓及輸入電流波形.....	30
圖 4.11 60Hz 模擬輸入電壓下暫態響應	34
圖 4.12 400Hz 模擬輸入電壓下暫態響應	35
圖 5.1 FPGA 元件基本結構.....	37
圖 5.2 實作電路組成架構圖	38
圖 5.3 實作電路照片	39
圖 5.4 A/D 轉換電路.....	40
圖 5.5 D/A 轉換電路.....	41
圖 5.6 開關驅動電路.....	41
圖 5.7 無電流感測控制模組時序圖.....	42
圖 5.8 200W 輸出 60Hz 實作輸入電壓及電流波形	45
圖 5.9 400W 輸出 60Hz 實作輸入電壓及電流波形	45
圖 5.10 600W 輸出 60Hz 實作輸入電壓及電流波形	45
圖 5.11 200W 輸出 400Hz 實作輸入電壓及電流波形	47
圖 5.12 400W 輸出 400Hz 實作輸入電壓及電流波形	47
圖 5.13 600W 輸出 400Hz 實作輸入電壓及電流波形	47
圖 5.14 60Hz 實作輸入電壓及電流之暫態響應.....	48
圖 5.15 400Hz 實作輸入電壓及電流之暫態響應.....	49

表目錄

表 1.1 諧波規範 IEC 61000-3-2 之各種設備之分類.....	4
表 1.2 IEC 61000-3-2 之諧波電流規範.....	4
表 1.3 DO-160 諧波電流限制.....	5
表 2.1 二種 PFC 電路的電流路徑表.....	9
表 3.1 開關訊號真值表.....	12
表 4.1 無電流感測模擬參數.....	21
表 4.2 200W 輸出 60Hz 模擬輸入電流諧波規範.....	27
表 4.3 400W 輸出 60Hz 模擬輸入電流諧波規範.....	28
表 4.4 600W 輸出 60Hz 模擬輸入電流諧波規範.....	29
表 5.1 各種架構下之 FPGA 使用率.....	43
表 5.2 60Hz 實作輸入電壓下各次諧波及其 THD_i	44
表 5.3 400Hz 實作輸入電壓下各次諧波及其 THD_i	46
表 5.4 200W 輸出 60Hz 實作輸入電流與諧波規範.....	51
表 5.5 400W 輸出 60Hz 實作輸入電流與諧波規範.....	52
表 5.6 600W 輸出 60Hz 實作輸入電流與諧波規範.....	53
表 5.7 200W 輸出 400Hz 實作輸入電流與諧波規範.....	54
表 5.8 400W 輸出 400Hz 實作輸入電流與諧波規範.....	55
表 5.9 600W 輸出 400Hz 實作輸入電流與諧波規範.....	56

第一章

緒論

1.1 研究動機

隨著科技的進步，各類電器從電網汲取的電能越來越多，且國際組織對於電力品質的要求越來越嚴格，功率因數校正(Power Factor Correction, PFC)，已成為電力電子產業中的重點。圖 1.1 分別繪出了傳統橋式整流和使用 PFC 電路的電流輸入波形，在圖 1.1(a)中，僅使用大電容濾波，將輸入電壓頻率 50/60Hz 的交流電壓濾波成直流電壓再給後級直流/直流變換器使用，此種方式的輸入電流是非正弦的，隱含的諧波含量甚大。而帶有功率因數校正的開關電源，如圖 1.1(b)，其輸入電流和輸入電壓波形相同，並且相位一致。這樣就可減少從電源中取得的無效功率降低，減少輸送電能的損失，更加有效的使用電能。

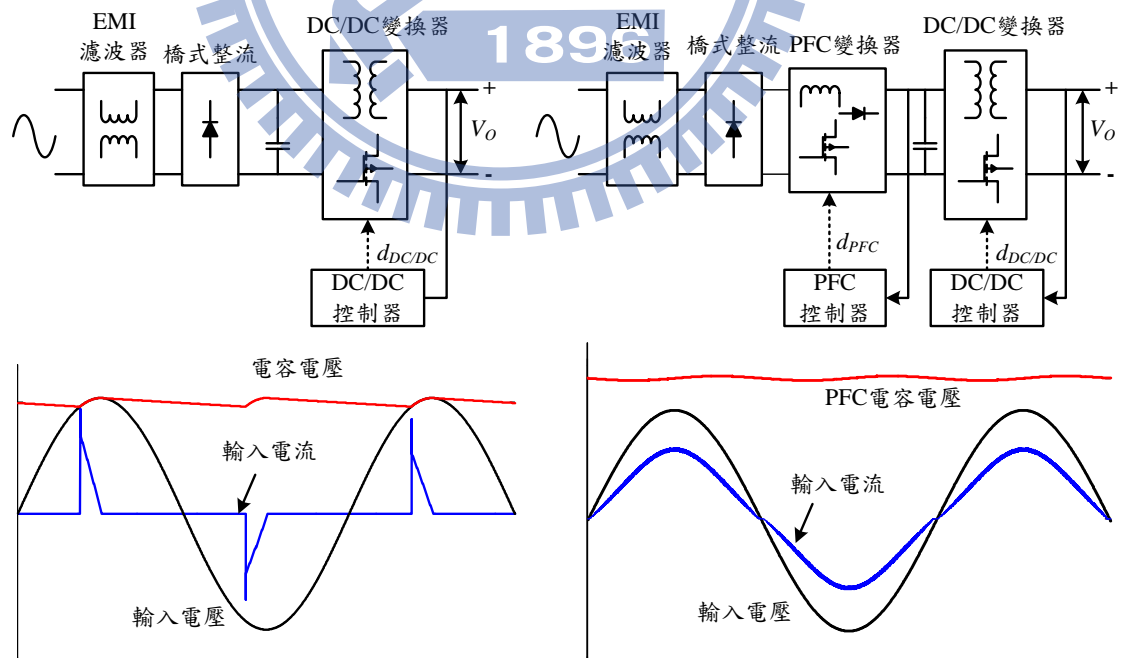


圖 1.1 傳統橋式整流的轉換器和帶有 PFC 的變換器比較

(a)傳統橋式整流的轉換器

(b)帶有 PFC 的轉換器

1.2 諧波電流

在上一小節中，描述了一理想電壓源加至一非線性負載，如圖 1.1(a)中之橋式整流電路等等，其輸入電流則不再是弦波，而電流的失真將帶有諧波的成分，同時也會造成公用點(PCC)線電壓的失真，由於電壓的失真通常都很小，故為了簡化分析，假設輸入電壓 $v_s = V_{SP} \sin(\omega t)$ 為無失真之理想輸入電壓，其中 V_{SP} 為理想輸入電壓之峰值，如下圖 1.2 所示。

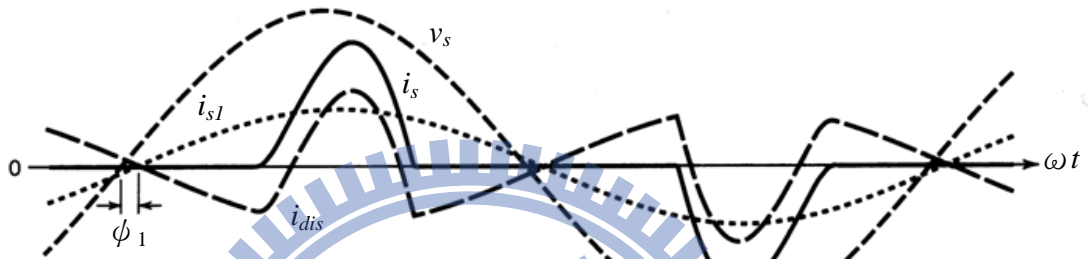


圖 1.2 電流失真

輸入電流 $i_s(t)$ 可以由下式表示，其中 $i_{s1}(t)$ 表示輸入電流之基本波， $i_{sh}(t)$ 表示輸入電流之第 h 次諧波。 i_{dis} 為基本波以外之電流成分。

$$i_s(t) = i_{s1}(t) + \sum_{h=2}^{\infty} i_{sh}(t) = i_{s1}(t) + i_{dis}(t) \quad (1.1)$$

由於輸入電流 $i_s(t)$ 為週期性波形，故可進一步採傅立葉級數展開為：

$$i_s(t) = I_{s1p} \sin(\omega_1 t - \phi_1) + \sum_{h=2}^{\infty} I_{shp} \sin(h\omega_1 t - \phi_h) \quad (1.2)$$

其中 ϕ_1 為輸入電壓和輸入電流基本波之相位差， ϕ_h 則為輸入電壓與輸入電流各次諧波之相位差。 I_{s1p} 及 I_{shp} 分別代表輸入電流之基本波峰值及諧波峰值大小。

因此定義另一項用來計量一波形之非弦波性質之名詞，亦即總電流諧波失真。總電流諧波失真為所有非基本波頻率有效值總和與基本波頻率有效值之比值：

$$\%THD_i = \sqrt{\frac{\sum_{h=2}^{\infty} I_{sh,rms}^2}{I_{s1,rms}^2}} \times 100\% \quad (1.3)$$

上式亦可改寫為：

$$\%THD_i = \sqrt{\frac{I_{rms}^2 - I_{sh,rms}^2}{I_{s1,rms}^2}} \times 100\% \quad (1.4)$$

1.3 諧波規範

根據前述的討論我們可以知道，電力系統中之電流諧波對整個系統的影響非常巨大，因此許多國家及國際組織皆定有類似的諧波規範。例如 IEC 61000-3-2、IEEE 519 等。

以 IEC 61000-3-2 而言，此規範的使用範圍為，當一設備之單相輸入電流小於 16 安培時，則均符合 IEC 61000-3-2 之諧波規範。其中 IEC 61000-3-2 又將各種設備作 A 至 D 之分類。其分類之定義如下表所示，接著根據表 1.1 中之分類可以得到表 1.2 之各類諧波之限制值。

A 類的規範為單純定義各諧波之大小限制，B 類則為 A 類各諧波限制值之 1.5 倍。C 類的各諧波限制值定義為基本波頻率下之電流的百分比，其中 λ 代表線路之功率因數。對於 D 類，各次諧波電流之限制是依據額定功率來決定的，輸入電流的各次諧波應不超過 D 類表推算出來之值。

然而對於高頻率航空用的諧波限制更是嚴苛，如 DO-160D 其諧波限制比起 IEC 61000-3-2 標準更加嚴峻，當輸入 400Hz 時，需要比 50/60Hz 更大的頻寬來控制來達到相同的電流諧波失真，表 1.3 列出了 DO-160 的諧波限制範圍。

表 1.1 諧波規範 IEC 61000-3-2 之各種設備之分類

A 類	<ul style="list-style-type: none"> ■ 平衡之三相系統 ■ 家用設備，不包含 D 類中之設備 ■ 白熾燈調光器 ■ 音響設備 ■ 工具，但不包含手持工具 ■ 任何不為 B 類、C 類、D 類中之設備
B 類	<ul style="list-style-type: none"> ■ 手持工具 ■ 非專業之焊接設備
C 類	<ul style="list-style-type: none"> ■ 照明設備
D 類	<ul style="list-style-type: none"> ■ 個人電腦及螢幕 ■ 電視接受器 <p>註：此類設備瓦數限制範圍為，大於 75 瓦特但不超過 600 瓦特</p>

表 1.2 IEC 61000-3-2 之諧波電流規範

諧波次數	A 類 (安培)	B 類 (安培)	C 類 (% i_{sI})	D 類 (mA/W)	
奇次諧波	3	2.30	3.45	$30 \times \lambda$	3.4
	5	1.14	1.71	10	1.9
	7	0.77	1.155	7	1.0
	9	0.40	0.60	5	0.5
	11	0.33	0.495	3	0.35
	13	0.21	0.315	3	0.296
	$15 \leq n \leq 39$	$0.15 \times \frac{15}{n}$	$0.225 \times \frac{15}{n}$	3	$\frac{3.85}{n}$
偶次諧波	2	1.08	1.62	2	
	4	0.43	0.645		
	6	0.30	0.45		
	$8 \leq n \leq 40$	$0.23 \times \frac{8}{n}$	$0.345 \times \frac{8}{n}$		

表 1.3 DO-160 諧波電流限制

諧波次數		單相
奇次諧波	3 倍數的奇次諧波 (h=3,9,.....,39)	$I_h = 0.15 \times \frac{I_1}{h}$
	h=5,7,11,13	$I_h = 0.3 \times \frac{I_1}{h}$
	h=17,19	
	h=23,25	
	非三倍數之奇次諧波 (h=29,31,.....,37)	
偶次諧波	h=2,4	$I_h = 0.01 \times \frac{I_1}{h}$
	其他偶次諧波 (h=6,.....,40)	$I_h = 0.0025 \times \frac{I_1}{h}$

1.4 本文架構

第一章：簡介諧波的影響，列出諧波規範。

第二章：介紹無橋式 PFC 電路，對傳統昇壓型 PFC 和無橋式 PFC 電路簡析。

第三章：對無橋式 PFC 電路模型化，推導無電流感測控制，並且提出實現方法。

第四章：進行電腦模擬控制器架構，用以驗證無電流感測控制。

第五章：介紹實作電路組成並實作驗證。

第六章：總結本論文之研究成果以及主要貢獻。

第二章

無橋式 PFC 電路

2.1 無橋式 PFC 電路介紹

在功因校正(PFC)電路中最常使用的就是昇壓型切換式整流器(Boost-type switching-mode rectifiers，後稱之為傳統昇壓型 PFC 電路)，如圖 2.1(a)，該架構簡單控制方便，為一橋式整流器加上一昇壓型轉換器而成，且此種電路架構的 PFC 改善方式已經相當成熟，但由於傳統昇壓型 PFC 的前端的橋式整流器導通損失較為嚴重，導致此種架構的效率受到一定程度的影響。但隨著半導體技術不斷的發展，具有低導通電阻的 MOSFET 等開關元件，及幾乎無反相回復特性的二極體，大幅度降低了 PFC 電路開關元件的損耗。但整流器造成的損耗卻一直無法得到根本的抑制，為了降低整流器所帶來的導通損耗，而電源工程師提出了一些不需要輸入整流器的新電路架構。在這些新的電路架構中，一種稱之為無橋式 PFC(Bridgeless PFC)電路架構，如圖 2.1(b)，為無橋式 PFC 電路由於此架構也相當簡單、開關電路驅動方便，省略了輸入的橋式整流器，比起傳統昇壓型 PFC 電路，效率可提高一些，若在大功率應用下效果更加明顯。

而無橋式 PFC 電路中本是全橋式 PFC 電路衍生出來，如圖 2.1(c)，全橋式 PFC 電路架構是由四個開關元件來代替橋式整流器的功能，並加以適當的開關驅動訊號來控制開關的導通與截止以實現功率因數校正的功能，但四個開關的驅動電路成本較高，控制電路也很複雜。而由全橋式 PFC 衍生出其他的電路架構，如圖 2.1(d)的 Totem-Pole PFC 架構，這些電路架構均屬廣義的無橋式 PFC 的範圍，此二種電路的區別僅僅是開關元件取代整流器或二極體和開關元件擺放的位置不同，從驅動電路上來無橋式 PFC 的二個開關元件的源極均接在同一點，驅動電路較為簡單。而 Totem-Pole PFC 電路架構中的開關元件，因擺放在同一臂中的上下橋，則需要隔離驅動：從控制上來說無橋式 PFC 的二個開關元件可以同步控制，

開極驅動信號是一樣的，由輸入電壓的極性來選擇某個開關元件動作。而

Totem-Pole 的驅動也比無橋式 PFC 要複雜許多，另工作中的 Totem-Pole PFC 電路會遇到開關元件本身二極體(Body Diode)反相恢復的問題，而無橋式 PFC 電路中的二極體僅實現續流的功用，就不存在此種困擾。

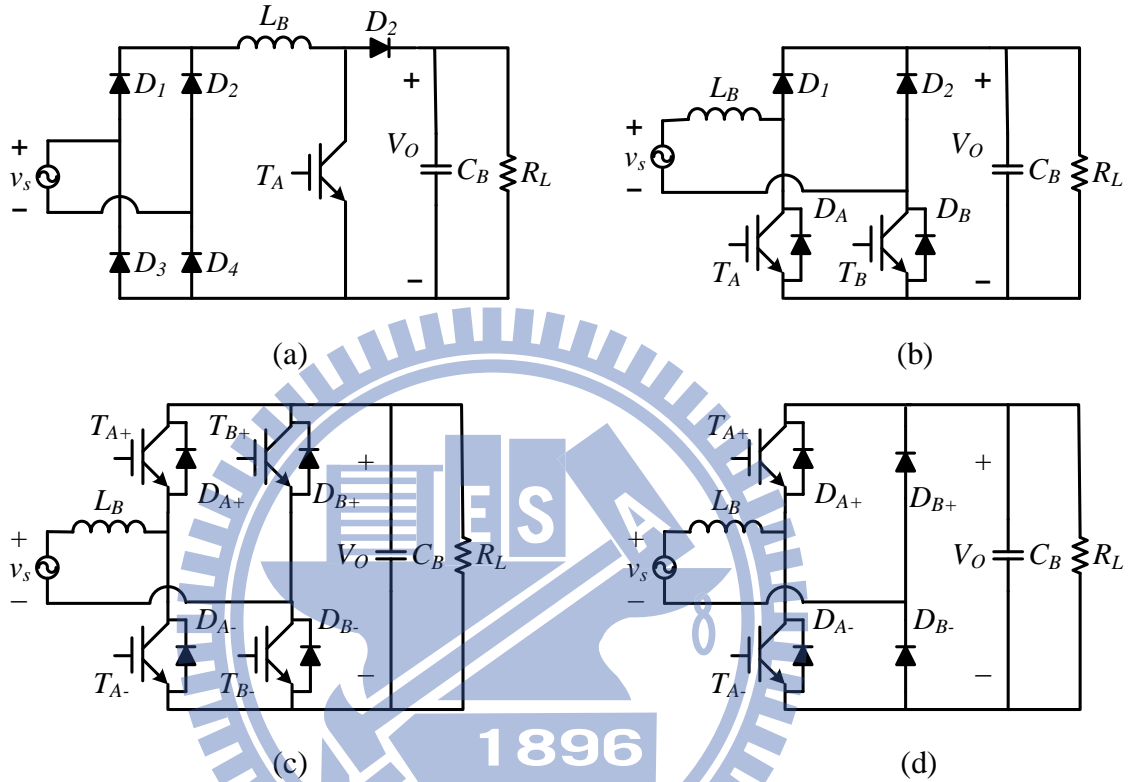


圖 2.1 PFC 電路圖

(a)傳統昇壓型 PFC 電路圖

(b) 無橋式 PFC 電路圖

(c)全橋式 PFC 電路圖

(d) Totem-Pole PFC 電路圖

2.2 無橋式 PFC 的工作原理及特點

無橋式 PFC 電路的工作流程如傳統的昇壓型 PFC 電路類似，其工作狀態依照輸入電壓的方向可以分為二個階段。第一階段如圖 2.2(a)所示當輸入電壓為正半週期， D_1 和 T_A 動作如同昇壓型轉換器(Boost-Type Converter)般，當 T_A 導通時流過電感 L 的電流增加(電流方向如圖所示)，流過 T_A 本身及 T_B 的本體二極體(Body Diode)，此時電感電流增加電感儲能。當 T_A 截止時電感電流通過 D_1 向電容儲能及

負載提供能量，電流再透過 T_B 的本體二極體續流，電感電流減少。第二階段如圖 2.2(b) 所示，當輸入電壓轉為負半週期時，改由 T_B 和 D_2 工作也如昇壓型轉換器，而電流透過 T_A 的本體二極體續流。二個階段交替出現，所以無橋式 PFC 的電感電流方向是周期性變換的，不像圖 2.1(a) 中傳統昇壓型 PFC 的電感電流感測是單相性的，比起一般 PFC 控制所需的電感電流檢測較為困難。

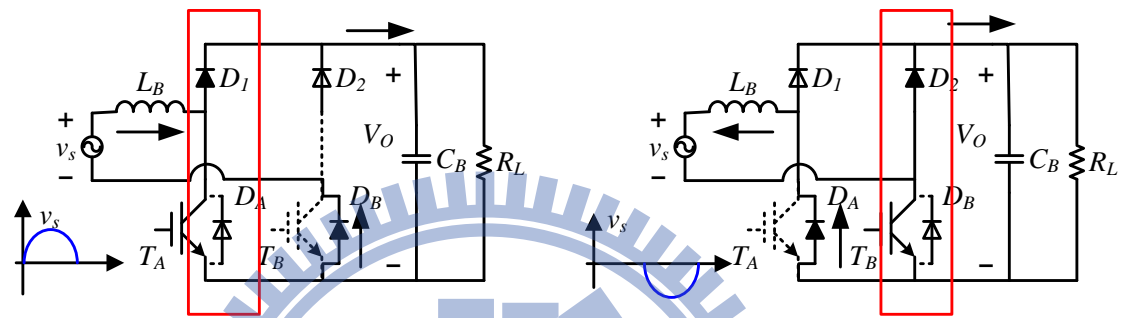


圖 2.2 無橋式 PFC 電路動作狀態

(a) 輸入電壓正半週期

(b) 輸入電壓負半週期

由上述可知基本的無橋式 PFC 工作狀態如二個昇壓型轉換器，故有另一名稱為 Dual Boost PFC，由於電路在輸入電壓的正負半週期內，自動切換所工作的昇壓型轉換器，二個昇壓型轉換器的開關元件驅動電路是可同步的，所以無橋式 PFC 電路的開關元件控制方式與傳統昇壓型 PFC 差異無多，使得在設計驅動電路上較為方便。

2.3 傳統昇壓型 PFC 和無橋式 PFC 效率簡析

本小節分析傳統昇壓型 PFC 和無橋式 PFC 電路的效率參數，闡述了無橋式 PFC 架構在效率上的優勢。首先須做幾點假設：

1. 不考慮輸入端 EMI 濾波器引起的損失。
2. 只對基本電路架構進行效率分析，不包含控制電路和驅動電路的損失。
3. 假設傳統昇壓型 PFC 和無橋式 PFC 的電感損耗相當。

而因輸入的電壓有周期對稱性，損耗分析僅對半個週期計算。假設圖 2.3(a)

為輸入電壓為正半週期，圖中標示出了此半週期內的電流方向，由此分析二種電路架構在導通損耗上的區別。而根據圖 2.3(b)所示之電流路徑，損失分析可歸納為下表。

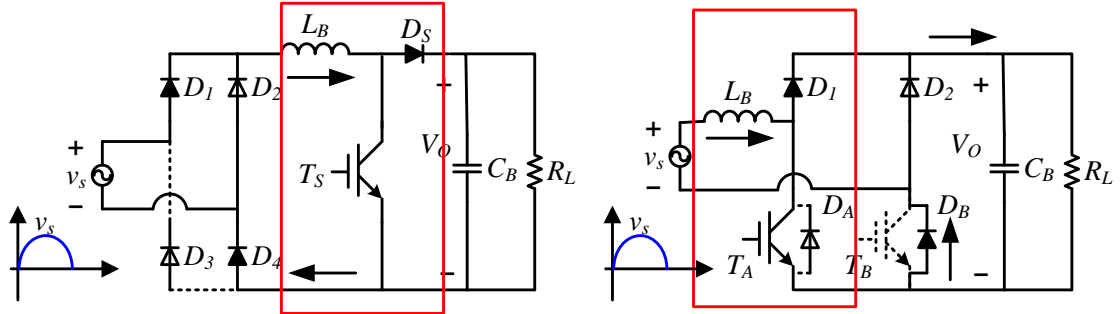


圖 2.3 正半週期內二種電路架構之電流路徑

(a)傳統昇壓型 PFC

(b)無橋式 PFC

表 2.1 二種 PFC 電路的電流路徑表

	傳統昇壓型 PFC	無橋式 PFC
整流二極體	2 個/ D_1+D_4	無
PFC 昇壓二極體	1 個/ D_S	1 個/ D_I
處於開關狀態的開關元件	1 個/ T_S	1 個/ T_A
處於續流的開關元件	無	1 個/ D_B
開關導通時電流全路徑	$D_1+T_S+D_4$	T_A+D_B
開關截止時電流全路徑	$D_1+D_S+D_4$	D_1+D_B

由表 2.1 可知，從電流通過的路徑上來看無橋式 PFC 電路，每個時段都比傳統昇壓型 PFC 電路少一個造成導通損失的元件。圖 2.3(a)中的 L_B 、 T_S 、 D_S 可看成一個昇壓型轉換器。同理圖 2.3(b)中的 L_B 、 T_A 、 D_I 也組成了一個昇壓型轉換器，其損失在理論上是一樣的。但無橋式 PFC 電路相對於傳統昇壓型 PFC 電路效率的提昇就在續流的開關元件和後者整流器二極體的造成的損失差別。

2.4 無橋式 PFC 電路的變化型

在上小節中，得知了無橋式 PFC 比起傳統昇壓型 PFC 導通損失上雖較為降低，在效率上稍作提昇，但也有其缺點，在參考文獻[4]中有說明，和傳統昇壓型 PFC 電路相較，其 EMI 中的共模干擾有顯著的影響，觀察圖 2.1(a)傳統昇壓型 PFC 的電路架構，電路在工作時無論任何時間下，其電流都有透過橋式整流器將輸入電源和輸出電壓的負端相連接；反觀圖 2.1(b) 無橋式 PFC 所示，僅僅在輸入電壓正半週期時，電流有透過開關元件 T_B 之本體二極體將輸入電源和輸出電壓的負端相連，但在輸入電壓為負半週期時則沒有，而輸出電壓的負端和輸入電源間猶如加上一個振幅相當為輸出電壓的高頻脈衝訊號，這高頻的電壓脈衝訊號透過輸入電源和輸出電壓的負端之間的寄生電容充電和放電，造成令人堪憂的共模干擾。而為了減少此一缺點，在電路中將原先的電感拆成二個感值相等的電感，並增加了二個二極體 D_3 和 D_4 ，如圖 2.4，使之更像二個直流/直流昇壓型切換式轉換器電路(Boost-Type Converter)，在輸入電壓為正半週期時，電流透過 D_4 將輸入電源與輸出電壓之負端相連接；同理，在輸入電壓負半週期時，電流透過 D_3 將輸入電源的與輸出電壓之正端相連接，來減少二端寄生電容造成的共模干擾。雖然增加了二個二極體但導通損失如同圖 2.3(b)未加二極體的無橋式 PFC 電路相當，減少了 EMI 影響對電路的影響，而本篇文章將以此電路作為主電路架構。

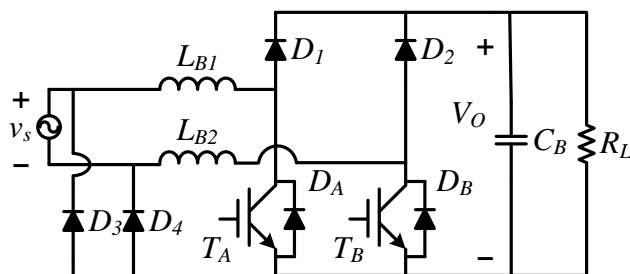


圖 2.4 增加二個二極體之無橋式 PFC 電路

第三章

無橋式 PFC 之無電流感測控制模型

本章將先對無橋式 PFC 電路做分析，如圖 3.1 所示，輸入電源正負端透過電感後連接上臂皆為二極體，下臂則為 IGBT 背接二極體所構成之無橋式 PFC 電路，下臂透過高頻開關的切換，使輸出電壓 V_O 能夠穩定為一固定電壓值，並且使輸入電流 i_s 維持弦波，並與輸入電壓同相為此電路的主要目的。在此圖中可以看見僅有二個電壓感測器，並不存在電流感測器，而詳細控制的方法，在後面章節加以推導，以達成無電流感測控制。

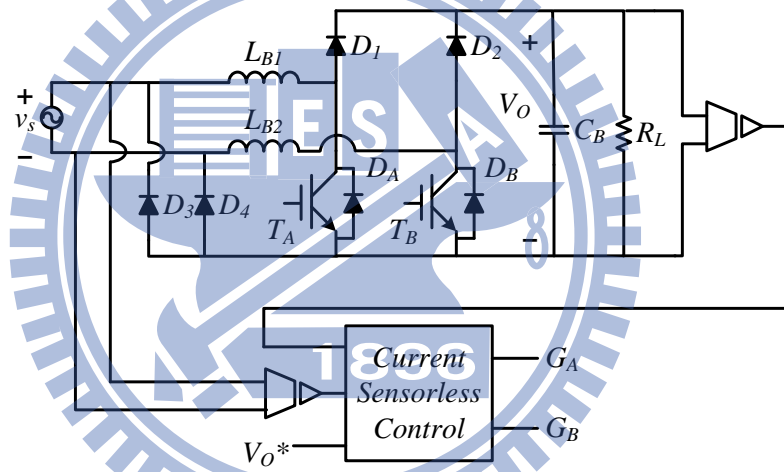


圖 3.1 無橋式 PFC 主要電路及控制方塊圖

3.1 電路模型假設

- (i) 假設開關之切換頻率 f_s 相當高，遠高於輸入電壓 v_s 之頻率，也就是在開關在切換週期 T_s 內可視為一定值。
- (ii) 因電路中任何情況下皆有二個半導體元件導通，在此先假設之二極體及開關之導通壓降平均為 V_{FT} 。
- (iii) 電路中二電感等效為一理想電感 L 加上電感本身之等效串連電阻 r_L 。
- (iv) 在輸出之直流系統上，忽略輸出電壓之漣波值。因此在電路穩態響應中，直流輸出電壓 V_O 等於輸出電壓命令值 V_O^* ($V_O \approx V_O^*$)。

3.2 開關訊號產生方式

在 2.2 小節中，我們已經介紹了無橋式 PFC 電路開關動作、電流方向，然後再寫下各個開關導通時所形成的各種狀態之數學式，再利用狀態平均法推出一個可以表示系統的通式，首先依照輸入電壓的正負週期、開關是否導通動作，寫出四種狀態，再對此四種狀態做推導並設計控制器。其開關分配的方式可參照表 3.1 及圖 3.2。

其中先導入一 $sign(x)$ 函數，為符號運算元。

$$sign(x) = \begin{cases} 1, & \text{when } x \geq 0 \\ -1, & \text{when } x < 0 \end{cases} \quad (3.1)$$

表 3.1 開關訊號真值表

$sign(v_s)$	$d(t)$	G_A	G_B	對應等效電路圖
1	H	On	Off	圖 3.3(a)
1	L	Off	Off	圖 3.3(b)
-1	H	Off	On	圖 3.4(a)
-1	L	Off	Off	圖 3.4(b)

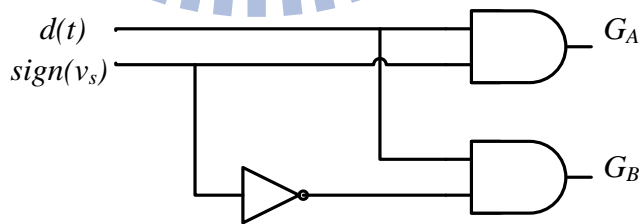


圖 3.2 開關訊號邏輯圖

需要的開關訊號邏輯方程式如下：

$$G_A = sign(v_s) \cdot d(t) \quad (3.2)$$

$$G_B = \overline{sign(v_s)} \cdot d(t) \quad (3.3)$$

3.3 電路推導

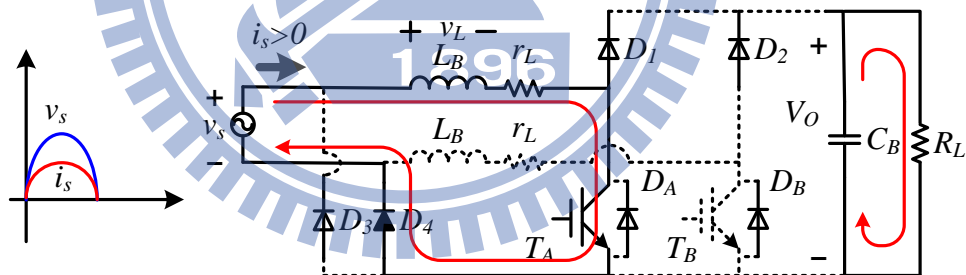
3.3.1 輸入電壓正半週期時($sign(v_s) > 0$)

於輸入電壓正半週期時($sign(v_s) > 0$)，且 PWM 訊號 $d(t)$ 為 High 時， G_A 為 High，而 G_B 為 Low，其等效電路如圖 3.3(a)所示，且定義輸入電源正端流出之電流為正及電感極性(如圖標示)，此時輸入電流和定義方向相同流過電感 L 、開關元件 T_A ，再透過二極體 D_4 形成一封閉迴路，此時對電感儲能，在此狀態下電感電流的斜率為正，此時輸入電流 i_s 上升，可寫出下式：

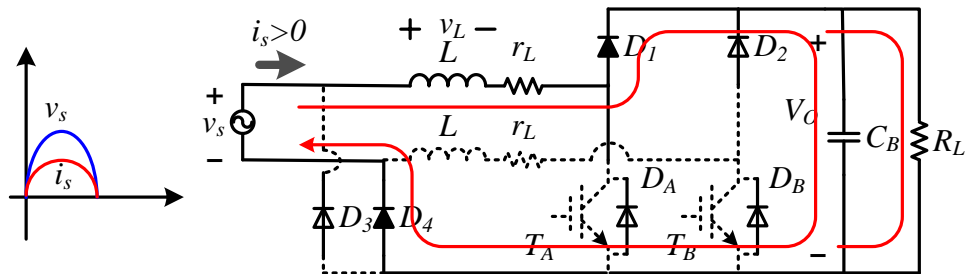
$$v_L = v_s - r_L i_s - V_{FS} - V_{Fd} \quad (3.4)$$

當 PWM 訊號為 Low 時， G_A 和 G_B 皆為 Low，其等效電路如圖 3.3(b)所示，進入此狀態時，電路瞬間加入了一定電壓源 V_O ，電路形成一交流電壓源、一電感及一直流電壓源串聯，而輸出的直流端電壓峰值大於交流電壓峰值，此時電感電流的斜率為負，此時輸入電流 i_s 下降，可寫出下式：

$$v_L = v_s - r_L i_s - 2V_{Fd} - V_O^* \quad (3.5)$$



(a)



(b)

圖 3.3 輸入電壓為正半週期時電路示意圖

(a) PWM 訊號 $d(t)$ 為 High (b) PWM 訊號 $d(t)$ 為 Low

3.3.2 輸入電壓負半週期時($sign(v_s) < 0$)

於輸入電壓負半週期時($sign(v_s) < 0$)，且 PWM 訊號 $d(t)$ 為 High 時， G_B 為 High，而 G_A 為 Low，其等效電路如圖 3.4(a)所示，輸入電流由輸入電壓負端流出，也形成一封閉迴路，在此狀態下電感電流的斜率為負，此時輸入電流 i_s 下降，可寫出下式：

$$v_L = v_s - r_L i_s + V_{Fd} + V_{FS} \quad (3.6)$$

當 PWM 訊號為 Low 時， G_A 和 G_B 皆為 Low，其等效電路如圖 3.4(b)所示，進入此狀態時，圖中可以看到此時輸出電壓兩端交叉接到電感端以及輸入電壓交流端，因輸出的直流電壓峰值大於交流電壓峰值，此時電感電流的斜率為正，此時輸入電流 i_s 將上昇，可寫出下式：

$$v_L = v_s - r_L i_s - 2V_{Fd} - V_O^* \quad (3.7)$$

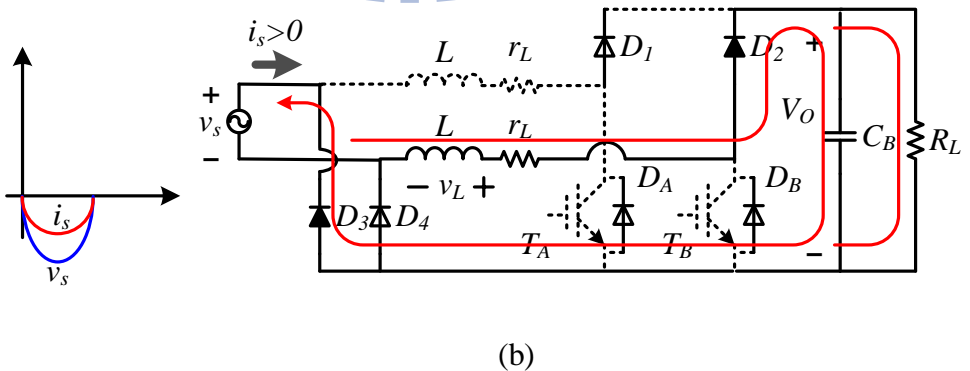
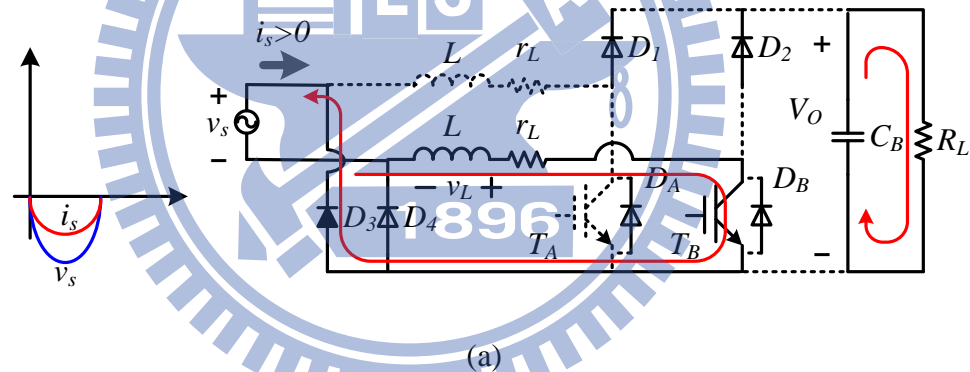


圖 3.4 輸入電壓為負半週期時電路示意圖

(a) PWM 訊號 $d(t)$ 為 High (b) PWM 訊號 $d(t)$ 為 Low

3.3.3 等效模型

式 3.1 至 3.4 中之 $(V_{FS}+V_{FD})$ 或 $2V_F$ ，我們在 3.1 小節第(ii)項中，將所有半導體元件的導通壓降平均為 V_{FT} 以簡化，再將式中 PWM 訊號同為 High 的狀態合併，觀察方程式(3.4)、(3.6)中，僅剩 V_{FT} 的符號不同，因此加入符號運算子 $sign(v_s)$ 後可以合併為通式(3.8)：

$$v_L = v_s - sign(v_s)V_{FT} - r_L i_s \quad (3.8)$$

同理可將 PWM 訊號同為 Low 的狀態合併，方程式(3.5)、(3.7)加入符號運算子 $sign(v_s)$ 後可以合併為通式(3.9)：

$$v_L = v_s - sign(v_s)V_{FT} - r_L i_s - sign(v_s)V_O^* \quad (3.9)$$

式(3.8)、(3.9)即為狀態的通式，經過等效電路模型化可見圖 3.5 所示。

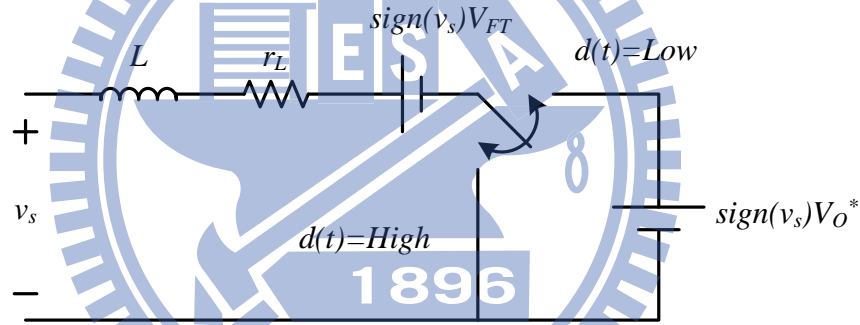


圖 3.5 等效電路圖

根據等效電路，引入狀態平均法的觀念，當責任週期 $d(t)$ 為 High 的時候，系統狀態為式(3.8)，當 $d(t)$ 為 Low 時，系統狀態為式(3.9)，若我們設定一個開關週期長度為 T_S ，而在此時間內 $d(t)$ 為 High 的時間為 $\bar{d} T_S$ ，反之 $d(t)$ 為 Low 的時間為 $(1-\bar{d})T_S$ ，則我們可利用平均狀態法將(3.8)、(3.9)乘上對應時間，並將式(3.9)之 v_L 標示為 v'_L 以示和式(3.8)區別，得下式所示：

$$\left[v_L \times \bar{d} T_S + v'_L \times (1-\bar{d}) T_S \right] / T_S \quad (3.10)$$

$$\Rightarrow \bar{v}_L = v_s - sign(v_s)V_{FT} - r_L i_s - (1-\bar{d})sign(v_s)V_O^*$$

其中 \bar{d} 及 $(1-\bar{d})$ 為 0 到 1 之間的純量與變數 $d(t)$ 不同，經由狀態平均法後之電感電壓 \bar{v}_L 亦加以上標以表示平均一開關週期內電感電壓，式(3.10)為此架構最終的狀態方程式。

3.4 無電流感測控制架構

下圖 3.6 為本論文所提及之無橋式 PFC 電路無電流感測器控制方塊圖，此方塊圖為一閉迴路系統，僅回授二個訊號，一為輸入電壓 v_s ，一為輸出電壓 V_o 。輸出二項訊號，各為輸入電壓符號函數 $sign(v_s)$ 以及開關導通訊號 $d(t)$ ，此二項訊號將做為開關分配及各導通時間的依據。

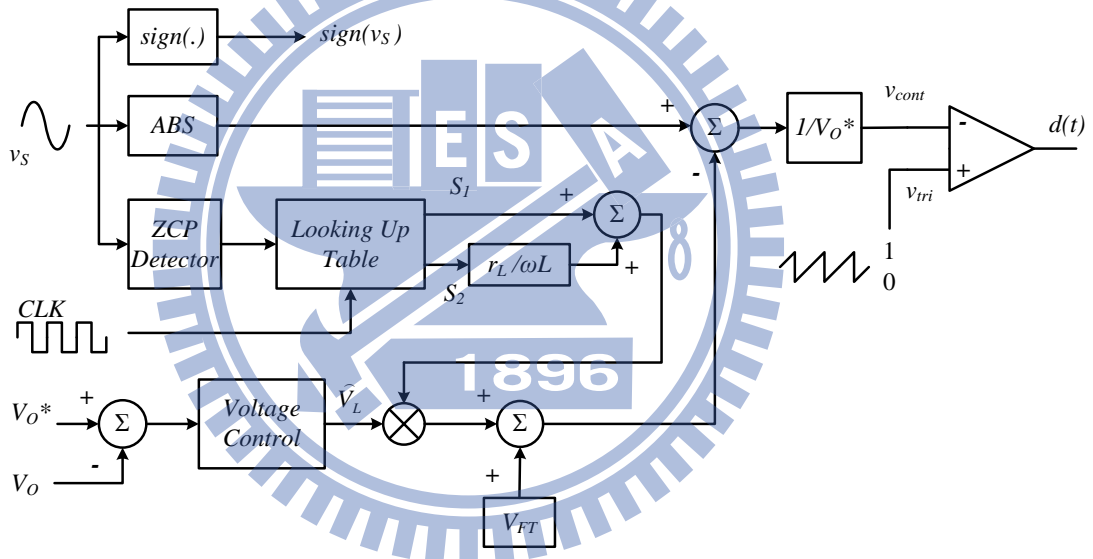


圖 3.6 無電流感測控制電路方塊圖

回授輸出電壓 V_o 與命令電壓 V_o^* 相減取得誤差訊號，再經過電壓控制器計算出估測電感電壓峰值 \hat{V}_L 。另一方面回授輸入電壓 v_s 經過零交越點偵測電路，如圖 3.7 所示，若是 ZCP 訊號打入，則系統內的時間軸立刻校正，此時間軸經過三角函數表，分別產生 s_1 、 s_2 訊號，其中 $s_1 = sign(v_s) \cos(\omega t)$ 、 $s_2 = |\cos(\omega t)|$ ，此訊號各自帶有正弦、餘弦函數，將產出 s_2 乘以 $\frac{r_L}{\omega L}$ 後與 s_1 相加，再與 \hat{V}_L 相乘並加入二極體與

開關元件的導通電壓補償，輸入電壓經過絕對值運算扣掉上述總和，再除以直流側的命令電壓 V_O^* 得到控制電壓 v_{cont} 。

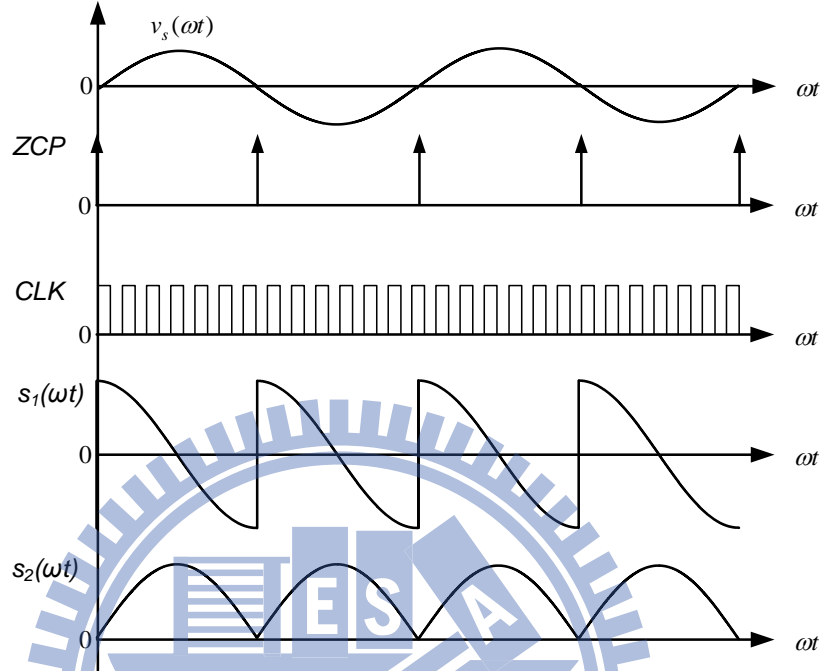


圖 3.7 控制迴路參考波型

將 v_{cont} 置於比較器之負端，將以固定三角波 v_{tri} 放置於正端，一般來說開關的頻率取決於此三角波的頻率，在現有的 IGBT 開關元件實現當中均可高達數十 kHz。當控制訊號 v_{cont} 小於三角波 v_{tri} 時， $d(t)$ 輸出為 1；反之當控制訊號波形 v_{cont} 大於或等於三角波型 v_{tri} 時， $d(t)$ 輸出為 0，也如式(3.11)表示。

$$d(t) = \begin{cases} 0, & \text{when } v_{cont} \geq v_{tri} \\ 1, & \text{when } v_{cont} < v_{tri} \end{cases} \quad (3.11)$$

因此可以得到無橋式 PFC 電路在無電流感測控制下，平均責任週期 \bar{d} 與控制週期所產生控制信號 v_{cont} 之間的關係式：

$$\bar{d} = 1 - v_{cont} = 1 - \frac{1}{V_O^*} \left[|v_s| - V_{FT} - \hat{V}_L \left(S_1 + S_2 \frac{r_L}{\omega L} \right) \right] \quad (3.12)$$

3.5 無橋式 PFC 電路之無電流感測控制推導

依據前章 2.3 節所推導之無橋式 PFC 電路系統個狀態所得到之通式為：

$$\overline{v_L} = v_s - \text{sign}(v_s)V_{FT} - r_L i_s - (1 - \overline{d})\text{sign}(v_s)V_O^* \quad (3.13)$$

並將(3.12)式代入式(3.13)中可得

$$\begin{aligned} \overline{v_L} &= v_s - \text{sign}(v_s)V_{FT} - r_L i_s - \\ &= \frac{1}{V_O^*} \left[|v_s| - V_{FT} - \widehat{V}_L \left(\text{sign}(v_s)\cos(\omega t) + \frac{r_L}{\omega L} |\sin(\omega t)| \right) \right] \text{sign}(v_s) V_O^* \\ &= -r_L i_s + \widehat{V}_L \cos(\omega t) + \widehat{V}_L \frac{r_L}{\omega L} \sin(\omega t) \end{aligned} \quad (3.14)$$

為了使輸入電流為弦波且同步於輸入電壓，我們希望電感電壓及電感電流為下(3.15)、(3.16)式：

$$\overline{v_L} = \widehat{V}_L \cos(\omega t) \quad (3.15)$$

$$\overline{i_L} = \frac{\widehat{V}_L}{\omega L} \sin(\omega t) \quad (3.16)$$

根據我們需要的電感電壓及電感電流，假設(3.15)和(3.16)式，若是將此二式中任一式代入(3.14)式均可得到另外一式，合乎所設定假設，因此設計控制信號為：

$$(1 - \overline{d}) = v_{cont} = \frac{1}{V_O^*} \left[|v_s| - V_{FT} - \widehat{V}_L \left(\text{sign}(v_s)\cos(\omega t) + \frac{r_L}{\omega L} |\sin(\omega t)| \right) \right] \quad (3.17)$$

在式(3.17)中 $\sin(\omega t)$ 項為式(3.14)式所產生，即希望的輸入電流，在設計的假設裡，輸入電流的相位必須與輸入電壓 v_s 同相，故 $\text{sign}(v_s)\sin(\omega t)$ 可改寫為 $|\sin(\omega t)|$ 。

為了方便後續的實現，我們將控制命令分為兩個部分，控制命令 S_1 、 S_2 ，此二訊號的相位必須與輸入電壓角度相同，將此二式代入則可將(3.15)式改寫為：

$$(1 - \overline{d}) = v_{cont} = \frac{1}{V_O^*} \left[|v_s| - V_{FT} - \widehat{V}_L \left(S_1 + S_2 \frac{r_L}{\omega L} \right) \right] \quad (3.18)$$

依據此推導之結果，設計控制訊號，實現方塊圖、模擬、實作均依照此推導基

礎來建立。

3.6 無電流感測控制的實現

此控制法運作方式如 3.4 節詳述，本節將針對實現此控制法必須注意之事項逐一說明：

1. 控制信號必須注意 s_1 、 s_2 訊號必須經過鎖相使之與輸入電壓 v_s 角度相同，才符合前小節推導過程，若沒有對準相位則很難控制輸入電壓及輸入電流達到預設的要求。
2. s_2 必須乘上 $\frac{r_L}{\omega L}$ ，用以補償電感的內阻，其中 ω 為輸入電壓的角頻率， r_L 為電感之內阻、 L 電感值均可由 LCR 計量測，寫入模擬或實作參數中，若輸入電壓頻率固定，則此值為一常數值。
3. 在做輸入電壓 v_s 、輸出電壓 V_O 取樣時，通常必須先做降壓，但在圖 3.6 中產生 v_{cont} 之前必須乘上 $\frac{I}{V_O^*}$ ，因此降壓的比例換算以及在控制器內數位轉換必須要精準，否則難以使輸出電壓追上命令電壓。

第四章

模擬驗證

在此利用來模擬的軟體為 PSIM，這套軟體廣泛的使用在電力電子領域的研究中。由於其簡單的操作介面，以及模擬快速的特性，使得使用者更利於在繁複的驗證中更快速的取得需要的資訊。

另外 PSIM 廣受電力電子領域研究所使用的原因，莫過於其內部的馬達驅動零件庫相當豐富，以及很強的控制系統模擬能力，在頻率響應的分析方便，另外它可以和 C/C++ 程式做連結也是其中一個相當重要的原因。以下簡單介紹 PSIM 這套軟體。

在 PSIM 中有提供五種功能模組供使用者使用，分別為供給馬達驅動系統使用的馬達驅動模組、設計數位控制系統的數位控制模組、提供 Matlab/Simulink 共同模擬的 Simulink 耦合模組、提供 MagCoupler 共同模擬的 JMAG 耦合模組及計算功率開關熱損失與溫度變化的熱分析模組，可以利用圖 4.1 簡單描述此五種功能模組。

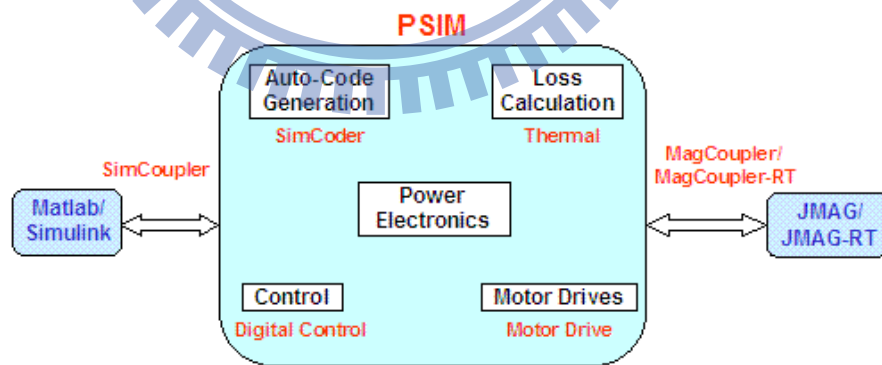


圖 4.1 PSIM 功能模組概要圖

在馬達驅動模組中，由於電動機模型與控制迴路相當複雜，因此對於馬達驅動系統設計與分析工作具有高度的挑戰性，利用 PSIM 的馬達驅動模組中的電動機模型與負載模型可以使複雜的分析工作簡化，此外豐富的零件庫可以使馬達驅動的模擬系統迅速的建立。在 PSIM 中馬達驅動模組包括：直流電動機、三相鼠籠式

與轉子繞線式感應電動機、永磁與外激是同步電動機、無刷直流電動機、切換式磁組電動機、定力矩、定功率及一般常用機械負載、速率與力矩感測器及換速齒輪箱。在數位控制模組中，不同於類比控制器，數位控制器的設計上，取樣頻率的影響、取樣延遲、量化誤差以及類比數位轉換上的誤差都必須考慮，因此利用數位控制模組，可以確認控制器的效能跟穩定度、研究取樣頻率對於系統效能的影響以及類比決定數位轉換器的解析度需求，對於數位控制器的設計有很大的幫助。在 PSIM 中數位控制模組包括：零階維持、單位延遲、Z 域轉移函數方塊、數位 FIR 與 IIR 濾波器、量化方塊、數位積分與微分控制器及環型緩衝器。

PSIM 正因為強大的計算能力以及豐富的資料庫內容，對於本論文在電路以及控制器驗證上有很大的幫助，以下的模擬也將大量的使用 PSIM 來做各種情況的模擬。

4.1 模擬電路及元件參數

表 4.1 無電流感測模擬參數

輸入電壓(峰值)	$\hat{v}_s = 155(110V_{rms})$
輸出電壓命令	$V_O^* = 200V$
負載電阻	$R=200、100、66.7\Omega$
開關切換頻率	$f_{tri}=40kHz$
輸入電壓頻率	$f=60Hz、400Hz$
電感	$L=2.6mH$
電感等效內阻	$r_L=0.3\Omega$
電容	$C_O=1410\mu F$
二極體導通壓降	$V_F=1.4V$
開關飽和電壓	$V_{sat}=1.8V$
模擬補償電壓	$V_{FT}=1.6V$

模擬電路圖如 4.2 圖所示，其中包含了無橋式 PFC 主電路、第三章所提及之無電流感測控制電路、開關分配訊號以及開關飽和電壓和二極體導通壓降的補償電路。

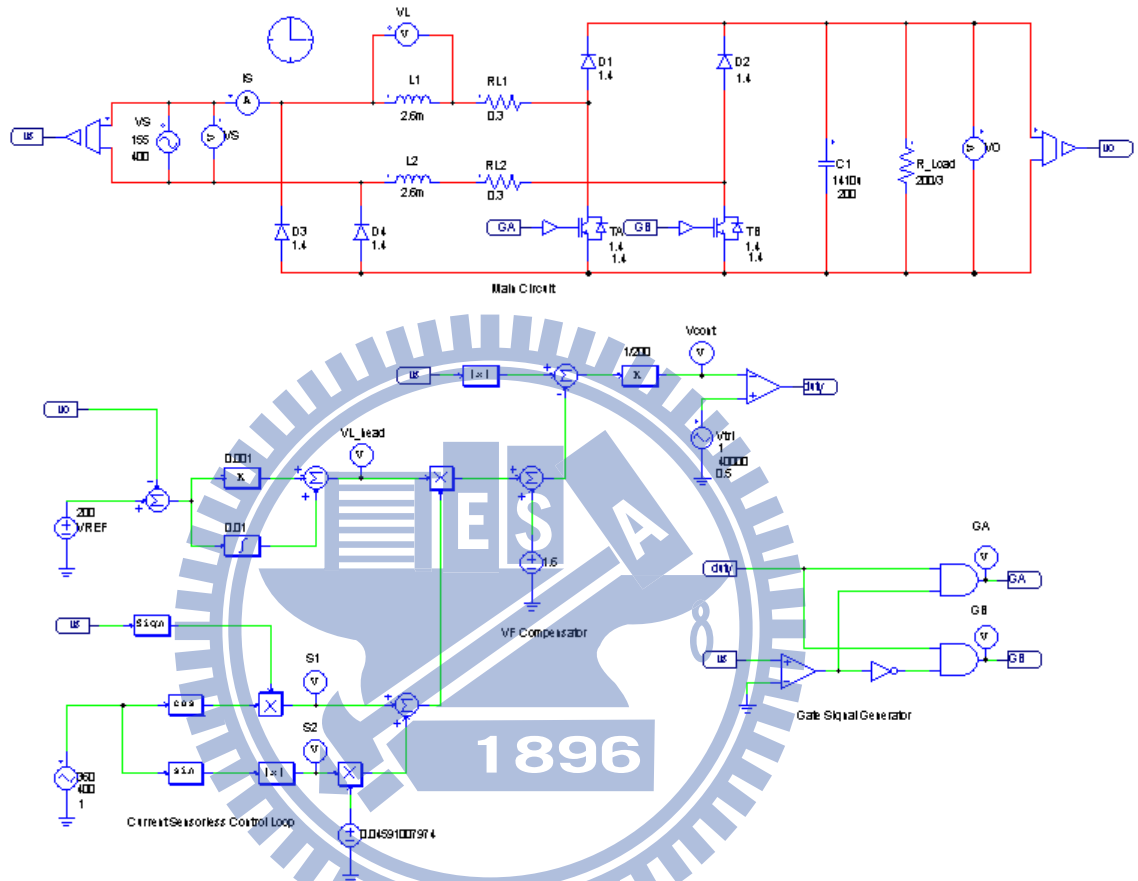


圖 4.2 PSIM 內無橋式 PFC 無電流感測電路模擬圖

4.2 無電流感測控制之模擬

此節利用電腦模擬軟體 PSIM 來驗證無電流感測控制下，是否能在無橋式 PFC 電路達到輸入電流波形規劃和輸出電壓調節的兩種特性。分別針對在 60Hz 和 400Hz 理想輸入電壓下做不同瓦數輸出的模擬。模擬會分為二部份分別為穩態模擬和暫態模擬，穩態模擬主要觀察其輸入電流波形是否為弦波且與輸入電壓是否同相，並計算輸入電流諧波失真 THD_i ，模擬輸出功率為 200W、400W 及 600W，其中將額定功率 P_o 為 400W 的控制器內部訊號繪出作為觀察。暫態模擬會將輸出功

率變動為 400W 至 600W，觀察輸出電壓是否穩定於輸出電壓的參考命令

$$V_o^* = 200V。$$

4.2.1 穩態模擬

將表 4.1 參數代入圖 4.2 無電流感測控制模擬圖進行模擬，然後觀察其內部訊號如 \hat{V}_L 、 s_1 、 s_2 、 v_{cont} 的觀測，若能得知在相對應時間下，各訊號彼此間的關係，對於實作電路上有一定程度的幫助，可從中發現實作電路中訊號是否正確。下圖 4.3 繪出在 60Hz 理想輸入電壓下輸出功率 400W 下控制器內部訊號的波形，圖 4.4 繪出在頻率 400Hz 理想輸入電壓下輸出功率 400W 下控制器內部訊號的波形。並對輸出 200W、400W 及 600W 下作穩態輸入電壓及輸入電流波形之模擬，如圖 4.5 至圖 4.10，並記錄其各次輸入電流諧波，和總諧波失真率並整理成下表 4.2 至表 4.7。



圖 4.3 為輸出 400W 時，在 60Hz 理想輸入電壓下，所進行的穩態模擬，控制輸出直流鏈電壓為 200V，可以觀察到輸入電流 i_s 保持與輸入電壓同相位，其 $THD_i = 8.28\%$ ，而 s_1 、 s_2 如同推導與輸入電壓時間軸同步，可依據此訊號作為實作時的參考。

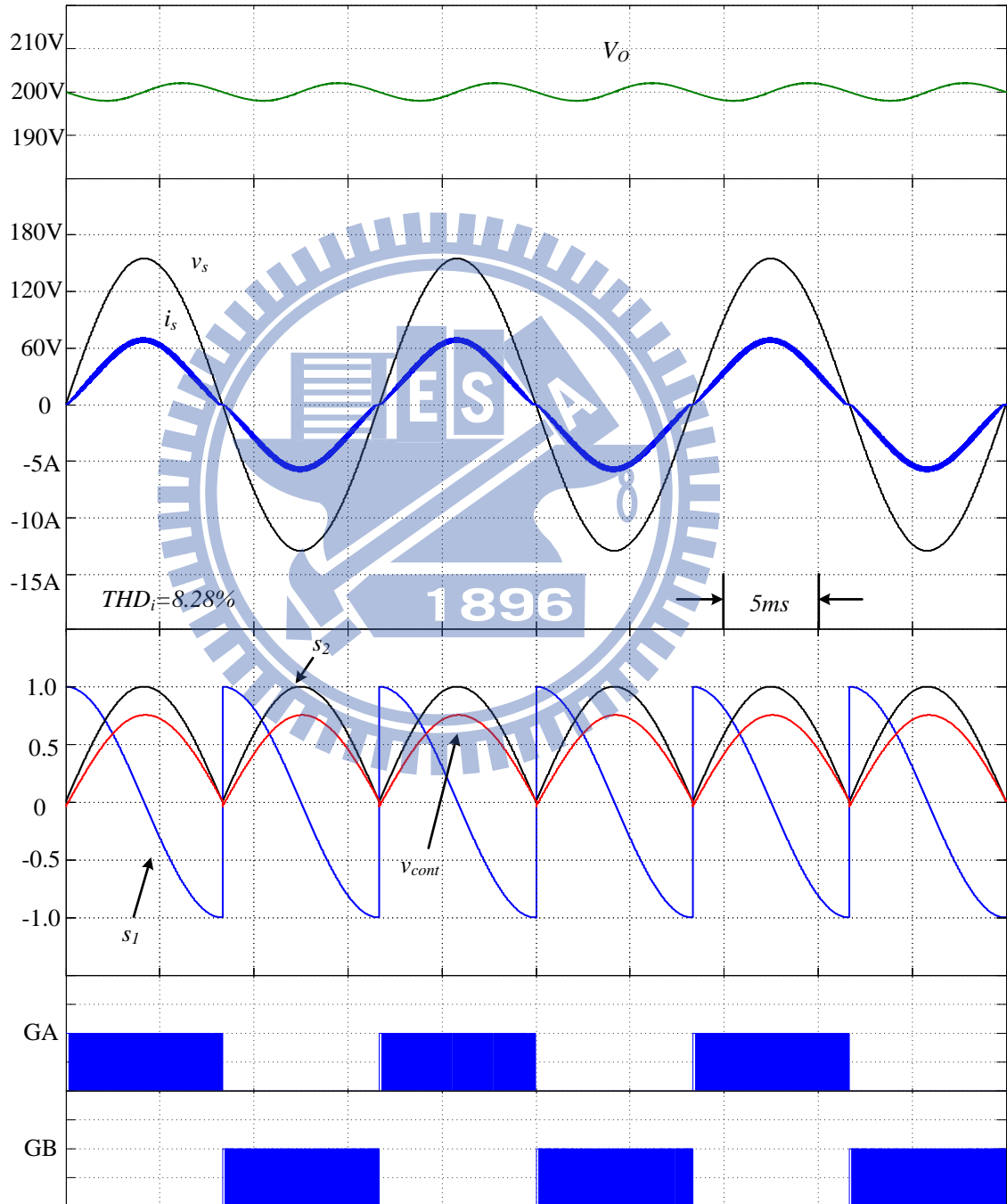


圖 4.3 60Hz 輸入電壓下穩態響應及控制器內部訊號

同時，亦模擬將輸入電壓頻率改為 400Hz，再觀察其控制器內部訊號，仍維持良好的正弦波形，其電流諧波失真 $THD_i = 9.37\%$ 。

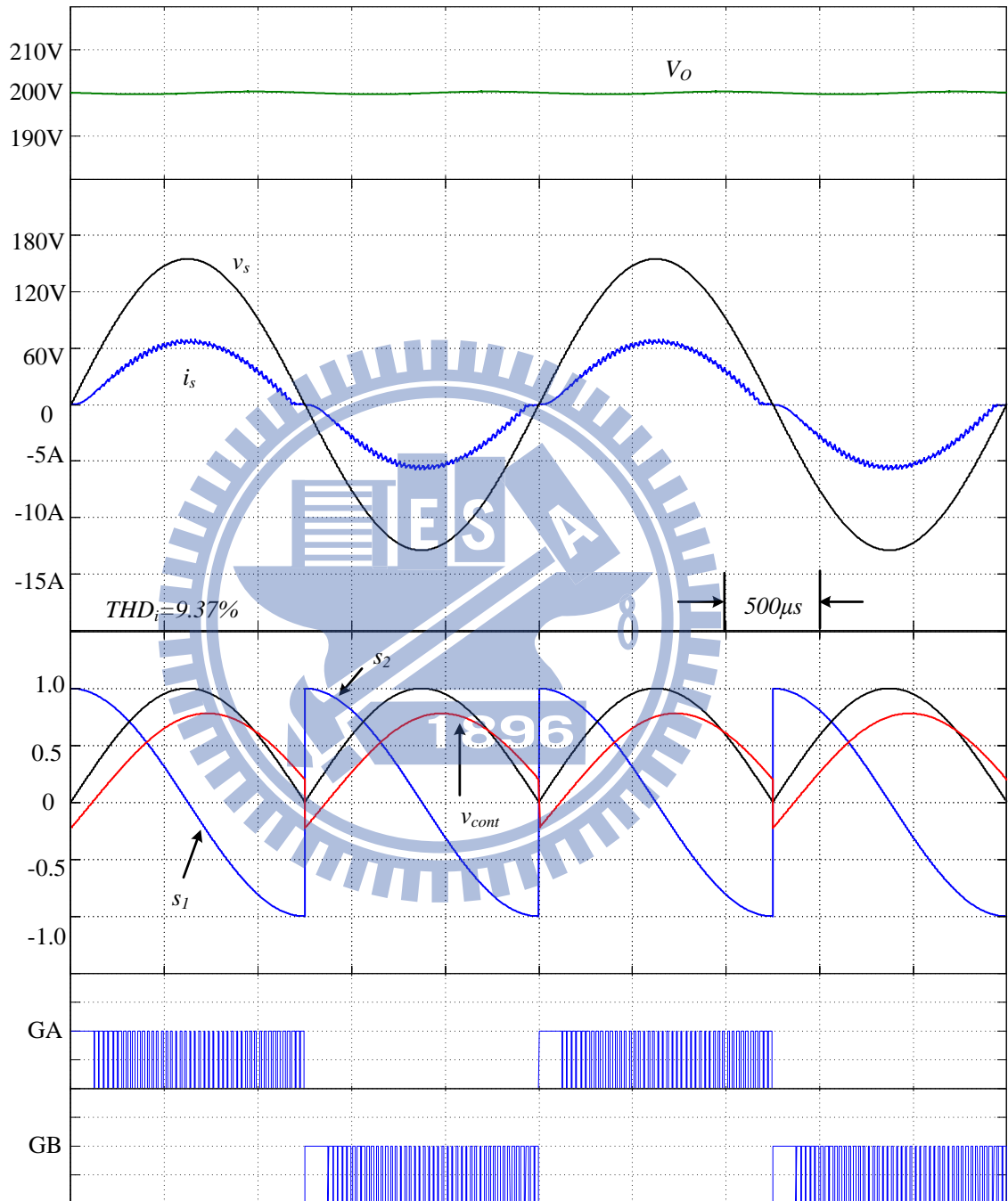


圖 4.4 400Hz 輸入電壓下穩態響應及控制器內部訊號

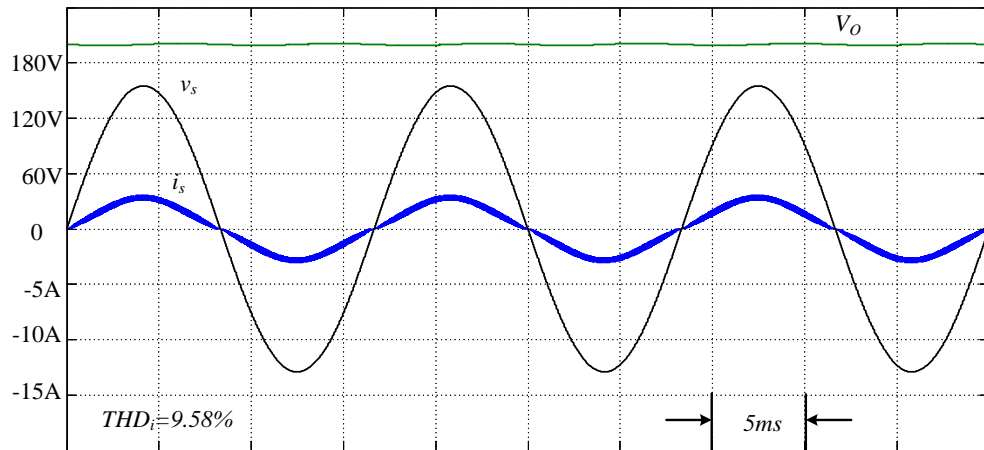


圖 4.5 200W 輸出 60Hz 模擬輸入電壓及輸入電流波形

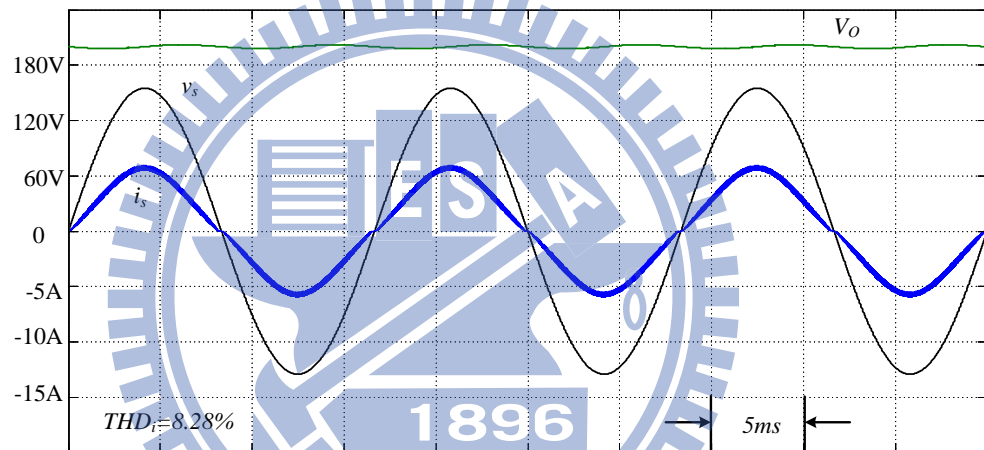


圖 4.6 400W 輸出 60Hz 模擬輸入電壓及輸入電流波形

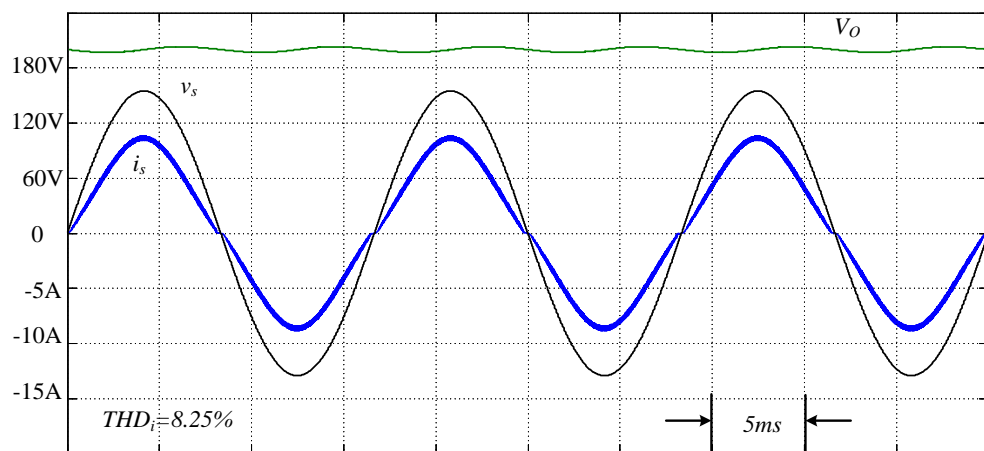


圖 4.7 600W 輸出 60Hz 模擬輸入電壓及輸入電流波形

表 4.2 200W 輸出 60Hz 模擬輸入電流諧波規範

輸出功率		A 類規範(安培)	D 類規範(安培)	輸入電流
諧波次數				
1-基本波 (單位：A)				1.8768
奇次諧波	3	2.30	$200 \times 0.0034 = 0.68$	0.1382
	5	1.14	$200 \times 0.0019 = 0.38$	0.0142
	7	0.77	$200 \times 0.0010 = 0.20$	0.0115
	9	0.40	$200 \times 0.0005 = 0.10$	0.0088
	11	0.33	$200 \times 0.00035 = 0.07$	0.0071
	13	0.21	$200 \times 0.000296 = 0.0592$	0.0059
	15	$0.15 \times \frac{15}{15} = 0.15$	$200 \times \frac{0.00385}{15} = 0.0513$	0.0050
	17	$0.15 \times \frac{15}{17} = 0.1324$	$200 \times \frac{0.00385}{17} = 0.0453$	0.0043
	19	$0.15 \times \frac{15}{19} = 0.1184$	$200 \times \frac{0.00385}{19} = 0.0405$	0.0038
偶次諧波	2	1.08		$7.3403e^{-6}$
	4	0.43		$9.9038e^{-7}$
	6	0.30		$7.9234e^{-7}$
	8	$0.23 \times \frac{8}{8} = 0.23$		$5.9488e^{-7}$
	10	$0.23 \times \frac{8}{10} = 0.184$		$4.7683e^{-7}$
	12	$0.23 \times \frac{8}{12} = 0.1533$		$3.9800e^{-7}$
	14	$0.23 \times \frac{8}{14} = 0.1314$		$3.4152e^{-7}$
	16	$0.23 \times \frac{8}{16} = 0.115$		$2.9907e^{-7}$
	18	$0.23 \times \frac{8}{18} = 0.1022$		$2.6624e^{-7}$
	20	$0.23 \times \frac{8}{20} = 0.092$		$2.4000e^{-7}$
THD_i (%)				9.58%

表 4.3 400W 輸出 60Hz 模擬輸入電流諧波規範

輸出功率 諧波次數		A 類規範(安培)	D 類規範(安培)	輸入電流
1-基本波 (單位：A)				3.7737
奇次諧波	3	2.30	$400 \times 0.0034 = 1.36$	0.2859
	5	1.14	$400 \times 0.0019 = 0.76$	0.0343
	7	0.77	$400 \times 0.0010 = 0.40$	0.0271
	9	0.40	$400 \times 0.0005 = 0.20$	0.0208
	11	0.33	$400 \times 0.00035 = 0.14$	0.0168
	13	0.21	$400 \times 0.000296 = 0.1184$	0.0139
	15	$0.15 \times \frac{15}{15} = 0.15$	$400 \times \frac{0.00385}{15} = 0.1027$	0.0118
	17	$0.15 \times \frac{15}{17} = 0.1324$	$400 \times \frac{0.00385}{17} = 0.0906$	0.0101
	19	$0.15 \times \frac{15}{19} = 0.1184$	$400 \times \frac{0.00385}{19} = 0.0811$	0.0087
偶次諧波	2	1.08		$1.4392e^{-5}$
	4	0.43		$1.9232e^{-6}$
	6	0.30		$1.5502e^{-6}$
	8	$0.23 \times \frac{8}{8} = 0.23$		$1.1646e^{-6}$
	10	$0.23 \times \frac{8}{10} = 0.184$		$9.3400e^{-7}$
	12	$0.23 \times \frac{8}{12} = 0.1533$		$7.7993e^{-7}$
	14	$0.23 \times \frac{8}{14} = 0.1314$		$6.6969e^{-7}$
	16	$0.23 \times \frac{8}{16} = 0.115$		$5.8704e^{-7}$
	18	$0.23 \times \frac{8}{18} = 0.1022$		$5.2253e^{-7}$
	20	$0.23 \times \frac{8}{20} = 0.092$		$4.7127e^{-7}$
THD_i (%)				8.28%

表 4.4 600W 輸出 60Hz 模擬輸入電流諧波規範

輸出功率 諧波次數		A 類規範(安培)	D 類規範(安培)	輸入電流
1-基本波 (單位：A)				5.6917
奇次諧波	3	2.30	$600 \times 0.0034 = 2.04$	0.4446
	5	1.14	$600 \times 0.0019 = 1.14$	0.0611
	7	0.77	$600 \times 0.0010 = 0.60$	0.0472
	9	0.40	$600 \times 0.0005 = 0.30$	0.0362
	11	0.33	$600 \times 0.00035 = 0.21$	0.0290
	13	0.21	$600 \times 0.000296 = 0.1776$	0.0239
	15	$0.15 \times \frac{15}{15} = 0.15$	$600 \times \frac{0.00385}{15} = 0.154$	0.0200
	17	$0.15 \times \frac{15}{17} = 0.1324$	$600 \times \frac{0.00385}{17} = 0.1359$	0.0170
	19	$0.15 \times \frac{15}{19} = 0.1184$	$600 \times \frac{0.00385}{19} = 0.1216$	0.0146
偶次諧波	2	1.08		$2.1230e^{-5}$
	4	0.43		$2.8028e^{-6}$
	6	0.30		$2.2802e^{-6}$
	8	$0.23 \times \frac{8}{8} = 0.23$		$1.7141e^{-6}$
	10	$0.23 \times \frac{8}{10} = 0.184$		$1.3757e^{-6}$
	12	$0.23 \times \frac{8}{12} = 0.1533$		$1.1496e^{-6}$
	14	$0.23 \times \frac{8}{14} = 0.1314$		$9.8804e^{-7}$
	16	$0.23 \times \frac{8}{16} = 0.115$		$8.6676e^{-7}$
	18	$0.23 \times \frac{8}{18} = 0.1022$		$7.7236e^{-7}$
	20	$0.23 \times \frac{8}{20} = 0.092$		$6.9682e^{-7}$
THD_i (%)				8.25%

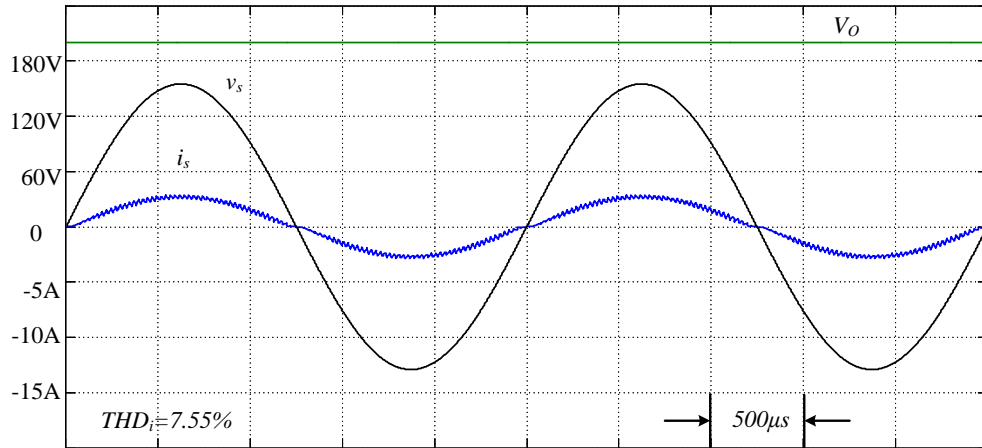


圖 4.8 200W 輸出 400Hz 模擬輸入電壓及輸入電流波形

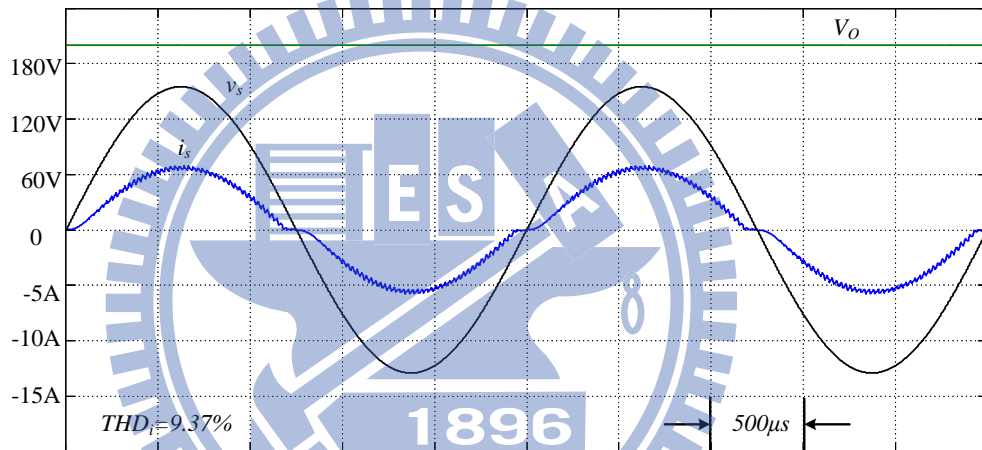


圖 4.9 400W 輸出 400Hz 模擬輸入電壓及輸入電流波形

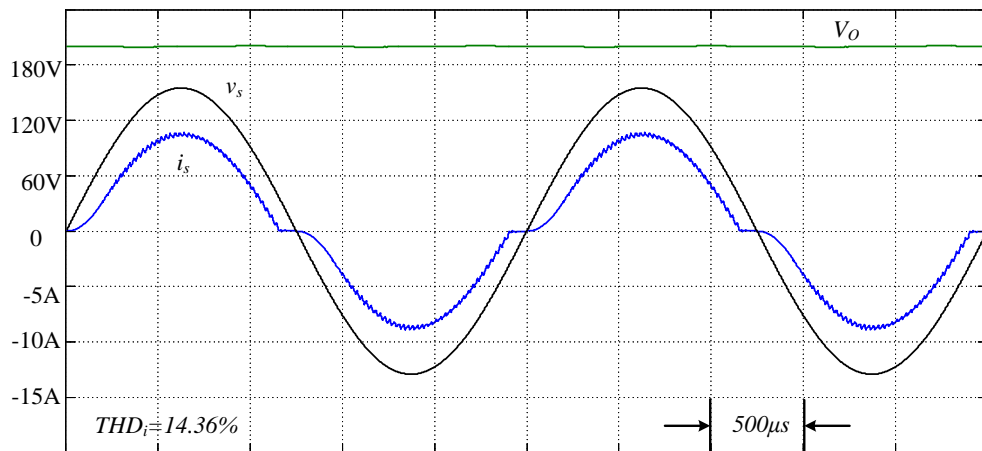


圖 4.10 600W 輸出 400Hz 模擬輸入電壓及輸入電流波形

表 4.5 200W 輸出 400Hz 模擬輸入電流諧波規範

輸出功率：200(W)					
基本波：1.8576(A)					
DO-160 奇次諧波規範		電流	DO-160 偶次諧波規範		電流
3	$0.15 \times \frac{1.8576}{3} = 0.0929$	0.0671	2	$0.01 \times \frac{1.8576}{2} = 0.0093$	$23.167e^{-6}$
5	$0.3 \times \frac{1.8576}{5} = 0.1115$	0.0381	4	$0.01 \times \frac{1.8576}{4} = 0.0046$	$7.9903e^{-6}$
7	$0.3 \times \frac{1.8576}{7} = 0.0796$	0.0262	6	$0.0025 \times \frac{1.8576}{6} = 774e^{-6}$	$4.9482e^{-6}$
9	$0.15 \times \frac{1.8576}{9} = 0.0309$	0.0194	8	$0.0025 \times \frac{1.8576}{8} = 580.5e^{-6}$	$3.5654e^{-6}$
11	$0.3 \times \frac{1.8576}{11} = 0.0507$	0.0149	10	$0.0025 \times \frac{1.8576}{10} = 464.4e^{-6}$	$2.7639e^{-6}$
13	$0.3 \times \frac{1.8576}{13} = 0.0429$	0.0117	12	$0.0025 \times \frac{1.8576}{12} = 387e^{-6}$	$2.2408e^{-6}$
15	$0.15 \times \frac{1.8576}{15} = 0.0186$	0.0092	14	$0.0025 \times \frac{1.8576}{14} = 331.71e^{-6}$	$1.8763e^{-6}$
17	$0.3 \times \frac{1.8576}{17} = 0.0328$	0.0073	16	$0.0025 \times \frac{1.8576}{16} = 290.25e^{-6}$	$1.6125e^{-6}$
19	$0.3 \times \frac{1.8576}{19} = 0.0293$	0.0058	18	$0.0025 \times \frac{1.8576}{18} = 258e^{-6}$	$1.1469e^{-6}$
$THD_i(\%)$			7.55%		

表 4.6 400W 輸出 400Hz 模擬輸入電流諧波規範

輸出功率：400(W)					
基本波：3.7729(A)					
DO-160 奇次諧波規範		電流	DO-160 偶次諧波規範		電流
3	$0.15 \times \frac{3.7729}{3} = 0.1886$	0.2763	2	$0.01 \times \frac{3.7729}{2} = 0.0189$	$23.167e^{-6}$
5	$0.3 \times \frac{3.7729}{5} = 0.2264$	0.1497	4	$0.01 \times \frac{3.7729}{4} = 0.0094$	$7.9903e^{-6}$
7	$0.3 \times \frac{3.7729}{7} = 0.1617$	0.0935	6	$0.0025 \times \frac{3.7729}{6} = 1.572e^{-3}$	$4.9482e^{-6}$
9	$0.15 \times \frac{3.7729}{9} = 0.0629$	0.0604	8	$0.0025 \times \frac{3.7729}{8} = 1179.03e^{-6}$	$3.5654e^{-6}$
11	$0.3 \times \frac{3.7729}{11} = 0.1029$	0.0387	10	$0.0025 \times \frac{3.7729}{10} = 943.23e^{-6}$	$2.7639e^{-6}$
13	$0.3 \times \frac{3.7729}{13} = 0.0870$	0.0240	12	$0.0025 \times \frac{3.7729}{12} = 786.02e^{-6}$	$2.2408e^{-6}$
15	$0.15 \times \frac{3.7729}{15} = 0.0377$	0.0142	14	$0.0025 \times \frac{3.7729}{14} = 673.73e^{-6}$	$1.8763e^{-6}$
17	$0.3 \times \frac{3.7729}{17} = 0.0666$	0.0084	16	$0.0025 \times \frac{3.7729}{16} = 589.52e^{-6}$	$1.6125e^{-6}$
19	$0.3 \times \frac{3.7729}{19} = 0.0596$	0.0058	18	$0.0025 \times \frac{3.7729}{18} = 524.01e^{-6}$	$1.1469e^{-6}$
THD_i (%)			7.55%		

表 4.7 600W 輸出 400Hz 模擬輸入電流諧波規範

輸出功率：600(W)					
基本波：5.6872(A)					
DO-160 奇次諧波規範		電流	DO-160 偶次諧波規範		電流
3	$0.15 \times \frac{5.6872}{3} = 0.2844$	0.7128	2	$0.01 \times \frac{5.6872}{2} = 0.0284$	$180.8e^{-6}$
5	$0.3 \times \frac{5.6872}{5} = 0.3412$	0.3371	4	$0.01 \times \frac{5.6872}{4} = 0.0142$	$2.8656e^{-6}$
7	$0.3 \times \frac{5.6872}{7} = 0.2437$	0.1668	6	$0.0025 \times \frac{5.6872}{6} = 2369.67e^{-6}$	$0.8897e^{-6}$
9	$0.15 \times \frac{5.6872}{9} = 0.0948$	0.0769	8	$0.0025 \times \frac{5.6872}{8} = 1777.25e^{-6}$	$1.1081e^{-6}$
11	$0.3 \times \frac{5.6872}{11} = 0.1551$	0.0361	10	$0.0025 \times \frac{5.6872}{10} = 1421.8e^{-6}$	$1.1273e^{-6}$
13	$0.3 \times \frac{5.6872}{13} = 0.1312$	0.0263	12	$0.0025 \times \frac{5.6872}{12} = 1184.83e^{-6}$	$0.8549e^{-6}$
15	$0.15 \times \frac{5.6872}{15} = 0.0569$	0.0226	14	$0.0025 \times \frac{5.6872}{14} = 1015.57e^{-6}$	$0.5419e^{-6}$
17	$0.3 \times \frac{5.6872}{17} = 0.1004$	0.0164	16	$0.0025 \times \frac{5.6872}{16} = 888.63e^{-6}$	$0.4542e^{-6}$
19	$0.3 \times \frac{5.6872}{19} = 0.0898$	0.0109	18	$0.0025 \times \frac{5.6872}{18} = 789.89e^{-6}$	$0.5143e^{-6}$
THD_i (%)		14.36%			

4.2.2 暫態模擬

除了考慮電路穩態下輸入電流的波形，同時也必須關注的是暫態響應，當負載突然變化的時候，由輕載變重載或由重載變輕載時，檢視其控制器是否能穩定的控制，直流鏈輸出電壓是否能儘快的調節，輸入電流波形是否仍保持正弦並和輸入電壓同相。同樣的，頻率 400Hz 輸入電壓下，再觀察其變化是否符合預期。如圖 4.11，在此模擬負載由 400W 升至 600W 之暫態響應，可看出 V_o 受到擾動約 7V，輸入電流 i_s 峰值由 6A 上升至 10A，其安定時間大約在 50ms。

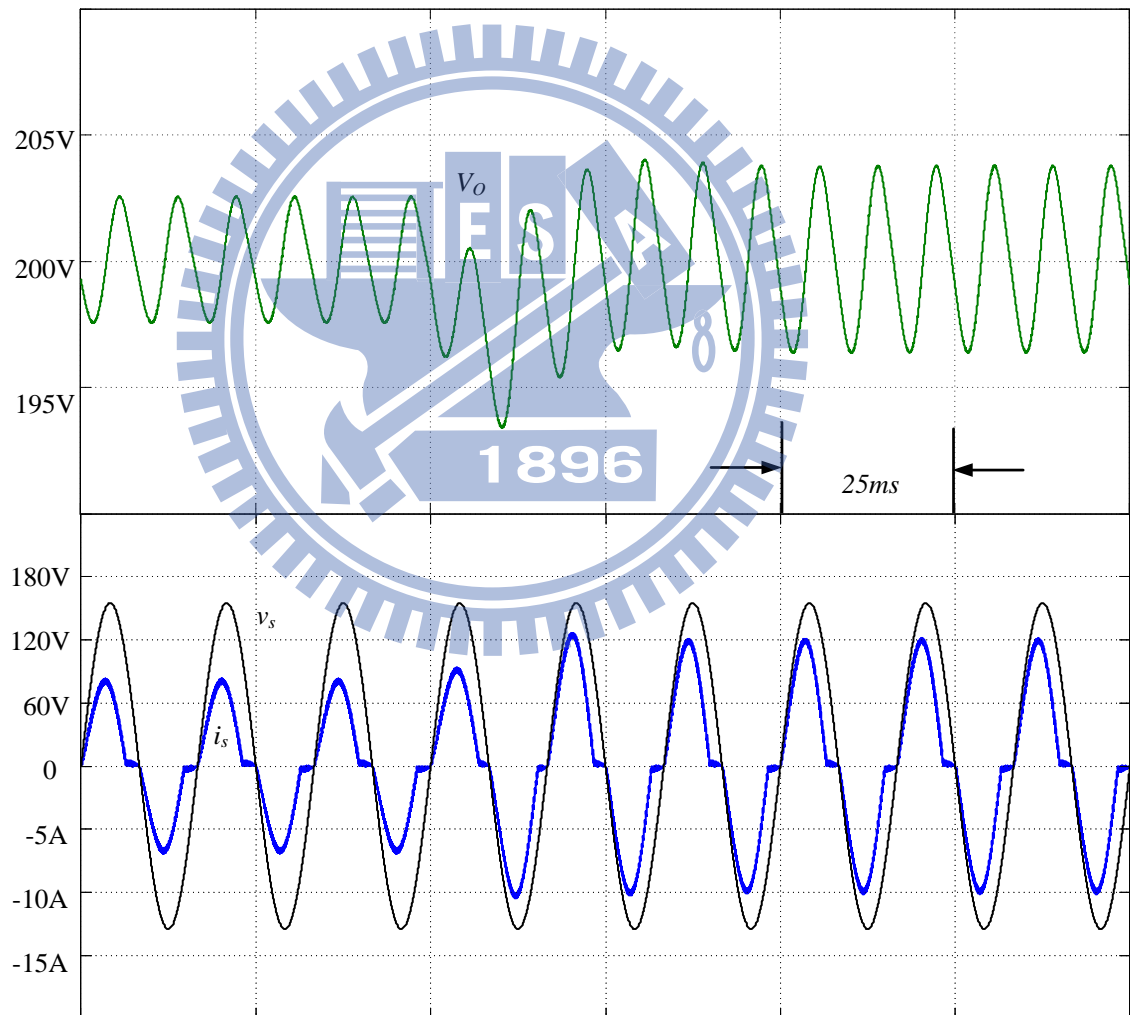


圖 4.11 60Hz 模擬輸入電壓下暫態響應

同樣地，改由頻率 400Hz 輸入電壓，如圖 4.12，觀察其波形可看出，大致和 60Hz 輸入電壓差距無幾， V_o 受到擾動約 8V，輸入電流 i_s 峰值由 5A 上升至 10A，其安定時間大約 200ms。

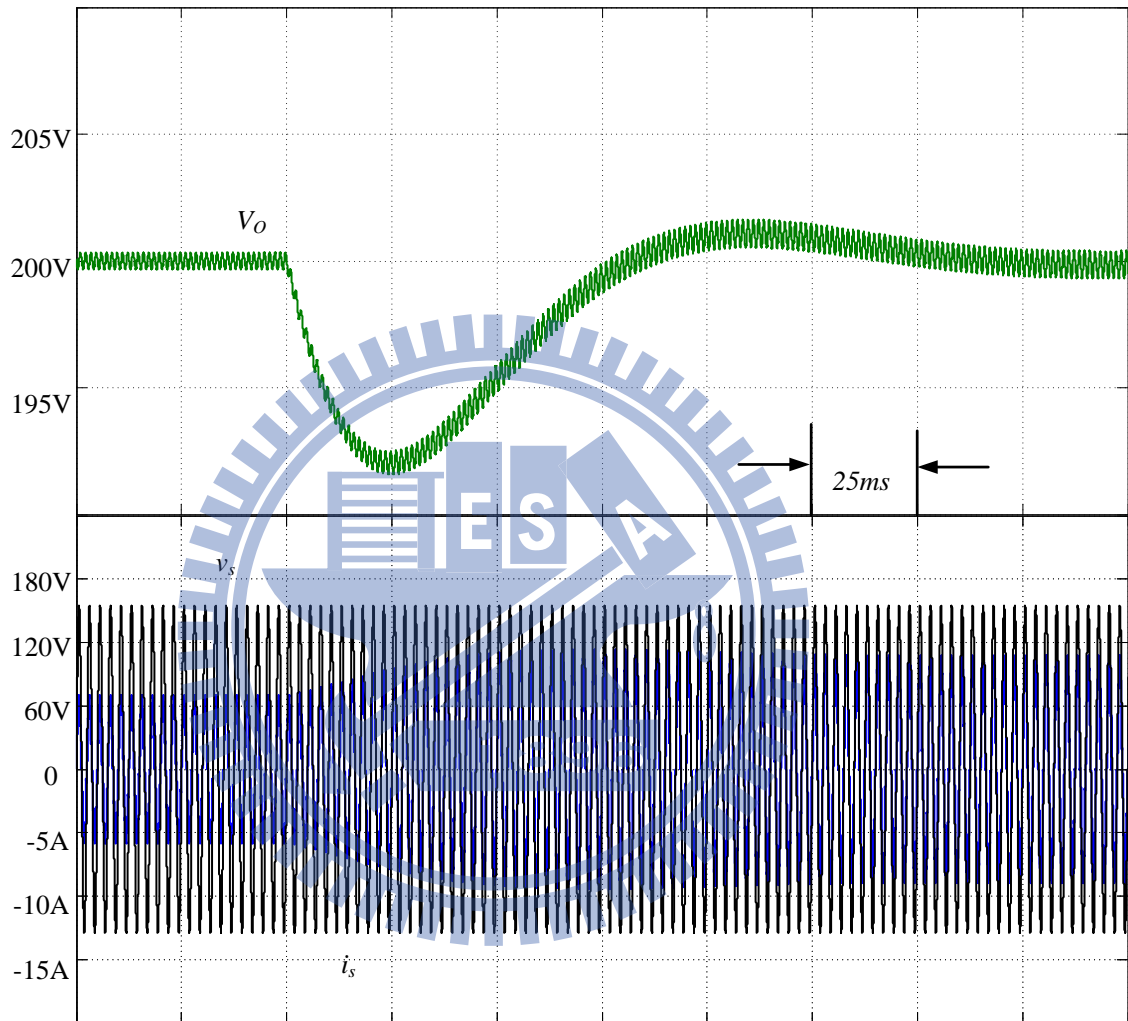


圖 4.12 400Hz 模擬輸入電壓下暫態響應

第五章

電路實作

5.1 現場可規劃邏輯陣列

現場可規劃邏輯陣列(Field Programmable Gate Array, FPGA)是一種可規劃邏輯陣列 IC, 它提供了「邏輯陣列」(Gate Array, GA)元件的特性與「可程式陣列邏輯」(Programmable Array Logic, PAL)元件的規劃彈性, 具有以下幾個特點: 一、使用者可以任意規劃邏輯電路; 二、FPGA 元件可以重複使用燒錄; 三、可以快速合成使用者的電路; 四、具有完善的軟體可以配合使用。

利用 FPGA 來實現控制系統有著不少優勢在, 除了有較高的運算時脈外, 其平行運算的能力也相當高。除此之外, 利用 FPGA 實現的控制器, 其行為也非常接近等效之類比控制器, 而且不但能保留其優點, 如無計算上的延遲(No Calculation Delay)、更高的頻寬(Higher Bandwidth)等; 更能將其缺點排除, 如參數飄移(Parameter Drifting)、較低的積分層級(Poor Level of Integration)[]。除此之外, 與其類似功能之 CPLD 做比較亦存在不少優點, 在編程上 FPGA 比 CPLD 具有更大的靈活性, 且 FPGA 非常適合複雜邏輯結構, 這些優勢也讓 FPGA 更適合使用在控制系統上。

FPGA 元件內部主要包含了三大部分, 數百個到數千個標準的可程式邏輯單元(Configurable Logic Blocks, CLBs), 排列形成 $M \times N$ 的電路矩陣; 每個邏輯單元均連接至縱向網路及橫向網路, 外部在圍繞一圈輸入輸出單元, 如圖 5.1 所示, 當電路完成設計後, 便可以把產生之電路架構編譯檔下載至 FPGA 中進行連線規劃, 成為一顆具有特定功能之 IC。這樣不但縮短了研發時間, 同時又擁有多次燒錄等功能, 而且也大大增加設計方法上之彈性。

接下來將介紹 FPGA 的設計流程。首先 FPGA 的設計方法有二種, 分別為圖形化流程(Schematic Flow)以及硬體描述語言編輯(HDL Editor), 然而近年來電路設

計的規模日益龐大複雜，若使用圖形化的設計方法相當的費時且容易發生錯誤，因此利用硬體描述語言的設計方法成為主流，一般常見的硬體描述語言有 VHDL、Verilog 等，而本論文所使用的為 Verilog，由於接近 C 語言的語法撰寫，使得較 VHDL 來的容易上手，而撰寫硬體描述語言最重要的就是電路的觀念，以及訊號實續的正確性，因此在程式撰寫的概念上有別於 C 語言與 MATLAB 等逐行編譯的程式語言。

將所希望設計的電路利用圖形化流程或是硬體描述語言完成後，接下來則可以做行為模擬(Behavior Simulation)及合成後的函數模擬(Function Simulation)，藉此可以瞭解所設計的電路功能是否正確。驗證功能正確後，就可以進行時序模擬(Timing Simulation)，模擬電路在燒入 FPGA 後，所造成的延遲是否符合需求。

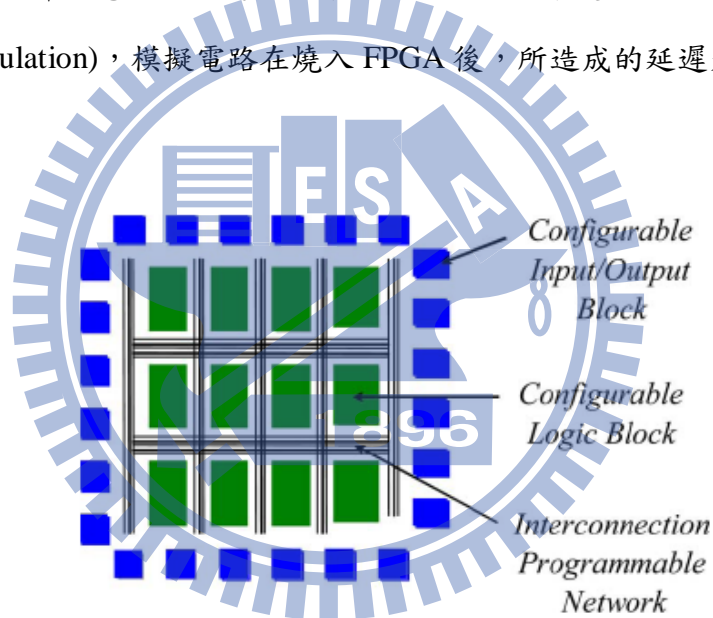


圖 5.1 FPGA 元件基本結構

本實驗所使用之現場可規劃邏輯陣列發展板，型號為 UBD-Spartan3E-ST3E。其核心為 Xilinx 公司研製，型號為 Spartan-3E XC3S250E，並配合 Xilinx ISE 8.2i 之軟體進行電路撰寫及模擬。此核心內含 4896 個 4 輸入 LUT(Look-Up Table)及 D 型正反器(D-Type Flip-Flop)，封裝為 PQFP 208 隻腳位，其中供使用者使用的腳位有 158 隻。此外，還包含了有 12 個 18K-bits 的 Block RAMs，12 個 18 乘 18 的硬體乘法器。實驗板上則提供 8 個指撥開關及 LED 介面及 40MHz 之石英震盪器等。

5.2 實作電路組成

無橋式昇壓型 PFC 電路之無電流感測控制實作，利用 FPGA 做為實現無電流感測實作之平台，並有數位類比轉換電路及開關驅動電路，當作主電路和 FPGA 之間的媒介。大致上其架構如圖 5.2 所示。

數位類比轉換電路分別為 A/D 轉換電路及 D/A 轉換電路，A/D 轉換電路是迴授輸出電壓及輸入電壓，D/A 轉換電路是對 FPGA 內部訊號進行觀察，雖 D/A 轉換電路並非無電流感測控制所必須，其目的為確保數位訊號的正確性。

開關驅動電路從 FPGA 接受 PWM 訊號，進行隔離放大以驅動無橋式昇壓型 PFC 中的主動元件 IGBT 開關。實際電路由圖 5.3 呈現。

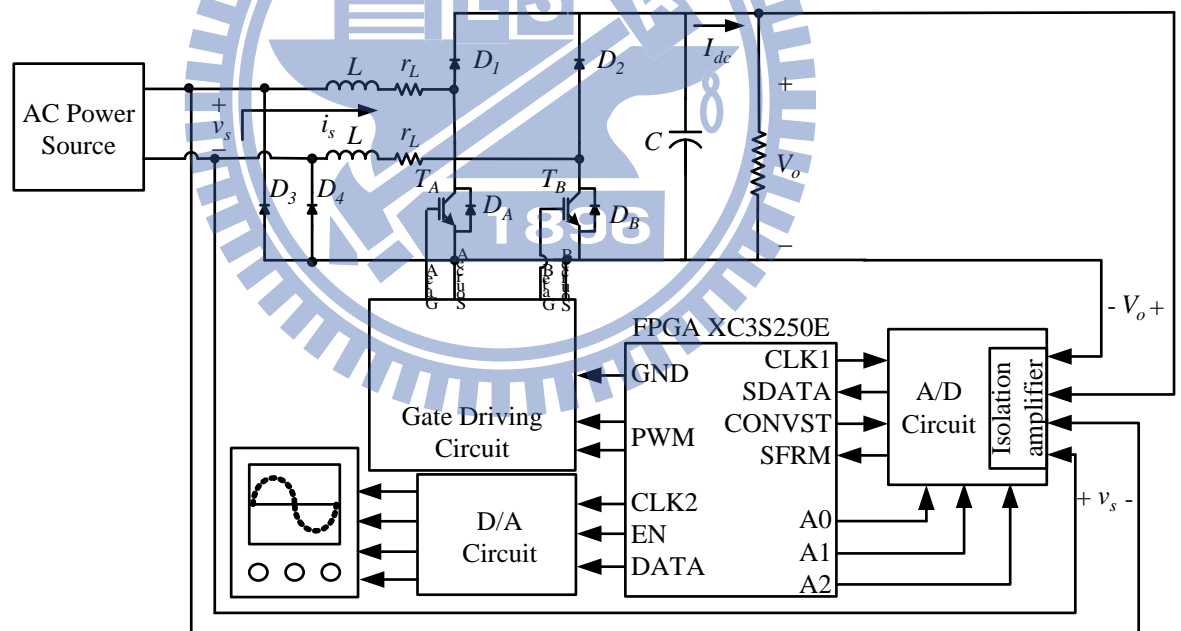


圖 5.2 實作電路組成架構圖

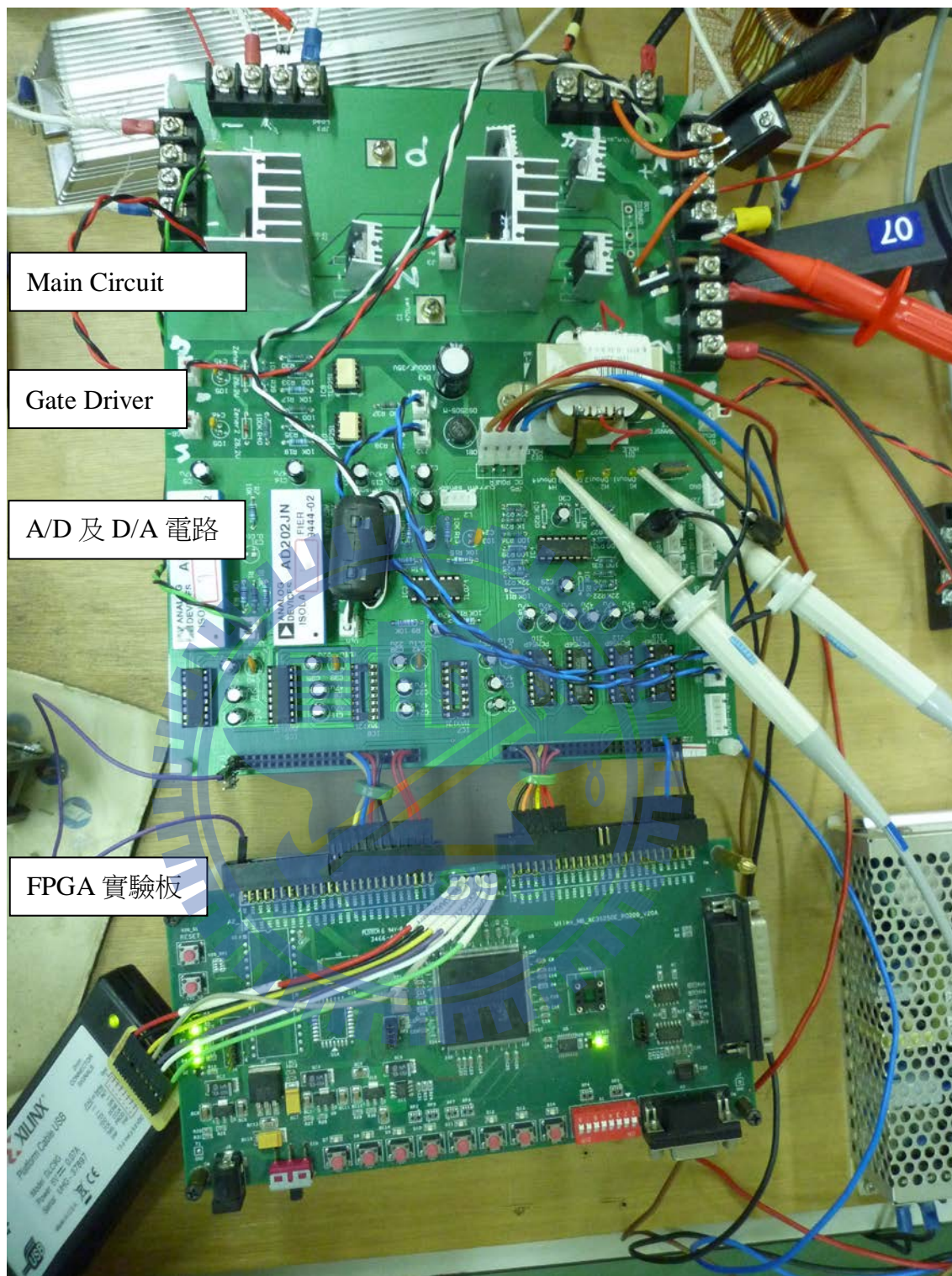


圖 5.3 實作電路照片

5.2.1 數位類比轉換電路(A/D Circuit & D/A Circuit)

實作中，完整 A/D 轉換電路組成如圖 5.4，整個電路實作需回授輸出電壓 v_o 及輸入電壓 v_s 。由於所使用 ADC 為 IC MAX121，其輸入取樣電壓為在 $\pm 5V$ 之間，故先藉由電阻分壓使其控制在 $\pm 5V$ 之間。電阻分壓後再用隔離放大器 IC AD202，使輸出電壓和輸入電壓二者與取樣電路隔離，避免輸出電壓與輸入電壓直接進入取樣電路對其造成損傷。再藉由 FPGA A/D 模組與 MAX121 兩者做溝通，即完成 A/D 轉換。

D/A 轉換器電路組成就如同圖 5.5 所示，首先由 FPGA 內部的 D/A 模組給予由數位轉類比 IC PCM56P 有特殊含意的訊號，再經過一個反相放大器將 PCM56P 輸出的 $\pm 3V$ 放大到 $\pm 10V$ 給予示波器觀測。而 D/A 轉換器在系統中同樣對於驗證 FPGA 程式有相當大的助益。

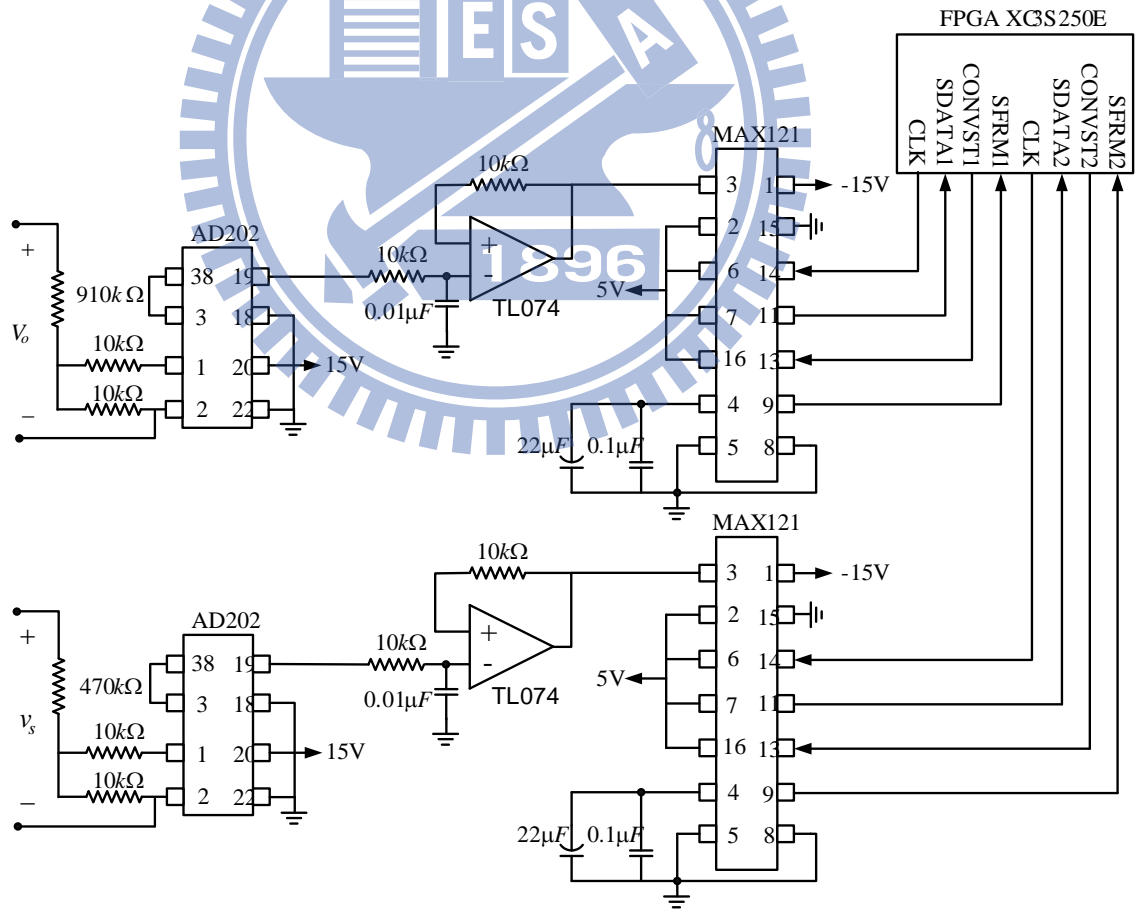


圖 5.4 A/D 轉換電路

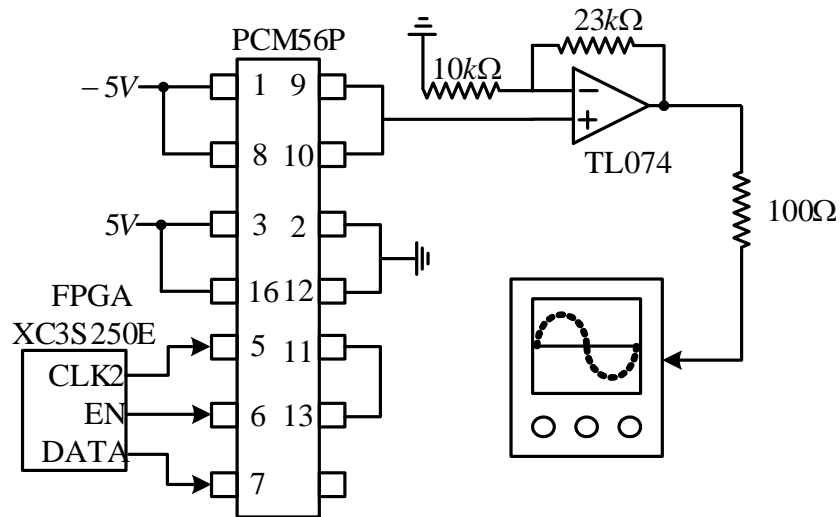


圖 5.5 D/A 轉換電路

5.2.2 開關驅動電路(Gate Driving Circuit)

如圖 5.6 所示，從 FPGA 取得 PWM 開關訊號進入開關驅動電路，在產生足夠大的電壓以驅動主電路中 IGBT 開關。從 PWM 訊號首先由光耦合 IC TLP250 進行隔離，當 PWM 為高電位時，TLP250 中腳位 6、8 會導通，經由後級電路可得 Gate 端與 Source 端產生約 16V 的電壓使得開關 IGBT 導通；反之 PWM 為低電位時，TLP250 中腳位 5、6 會導通，經由後級電路可得 Gate 端與 Source 端產生約 -8V 的電壓使得開關 IGBT 截止。

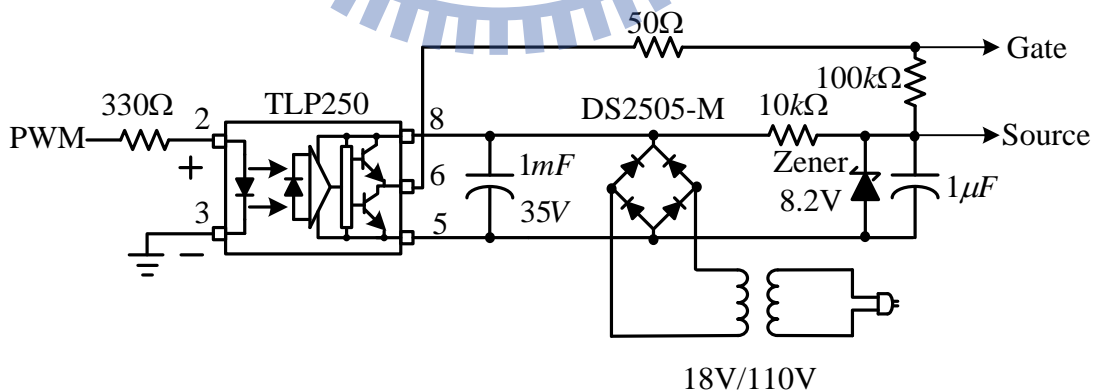


圖 5.6 開關驅動電路

5.3 控制器合成

在合成實作硬體電路後，接下來就需要在 FPGA 電路板中完成無電流感測控制架構之電路合成。在 FPGA 中所包含模組，主要有無電流感測控制模組，另外還有除頻模組、A/D 模組、D/A 模組、查表模組、以及 PWM 模組。而無電流感測控制模組的合成可用圖 5.7 無電流感測控制模組時序圖做解釋。

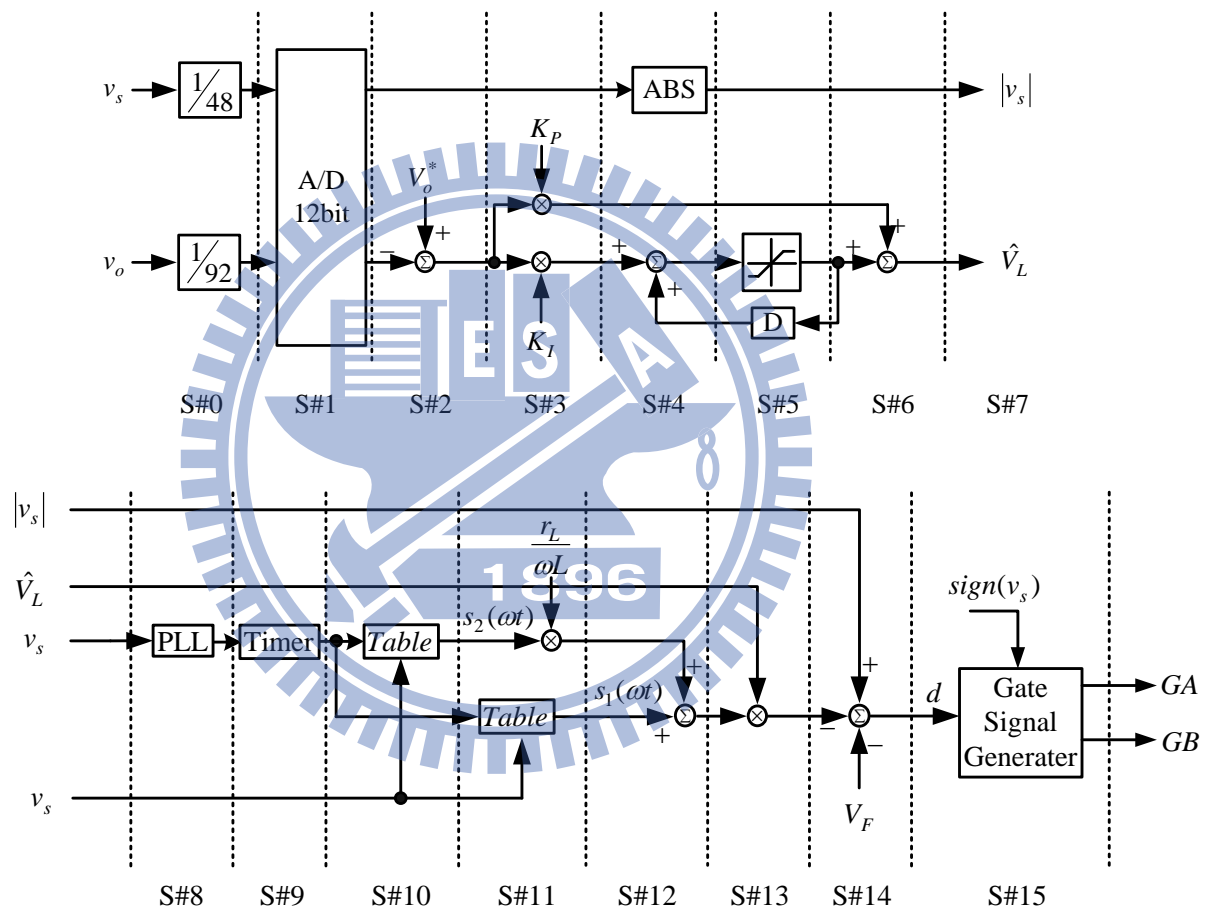


圖 5.7 無電流感測控制模組時序圖

S#1 到 S#7 為電壓迴路，其中包含 A/D 模組和一個電壓控制器。首先將回授的輸出電壓和輸入電壓經過 A/D 模組轉換成 12 位元的資料，最高位元為正負位元。電壓控制器是由一個 PI 控制器所構成，由 A/D 模組產生輸出電壓的數位訊號

與參考電壓 V_O^* 相減後所得誤差，進入 S#3 到 S#6 的 PI 控制器產生出電感電壓命令 \hat{V}_L 。

S#9 到 S#17 為開關訊號產生迴路，S#8 將輸入電壓 v_s 鎖相，並抓取其零交越點，然後在 S#9 以週期(1/60)及(1/400)秒設定一計時器以用來查表，S#10、S#11 中內建 2 個表分別為 $s_1(\omega t)$ 和 $s_2(\omega t)$ ，考慮在每一次開關週期查表一次，亦即查表的時間間隔為 $25 \mu s$ 。而 S#10 到 S#11 為將此查表位置累加，而進入 S#12、S#13 再乘上估測電感電壓 \hat{V}_L 。而後 S#14 加上開關及二極體不理想因素的補償，產生出控制訊號 d。S#15 為 PWM 的模組，利用 S#14 所產生的控制訊號 d，經過開關分配而得到二組開關訊號輸出。整個控制迴路利用 $25 \mu s$ 完成。其餘在 FPGA 所實現的模組還有除頻模組和 D/A 模組。除頻模組是將 FPGA 基頻 40MHz 分別產生 20MHz、10MHz、4MHz、40kHz 和 20kHz 等時脈，來當作觸發其餘模組所用。D/A 模組雖非無電流感測控制器所必須，但可藉由此模組觀察 FPGA 內部訊號，對於 FPGA 的合成有很大的幫助。表 5.1 為此架構下之 FPGA 使用率。

表 5.1 各種架構下之 FPGA 使用率

	可使用	已使用(使用率)
4 Input LUTs	4896	1394(28%)
D-type Flip-Flops	4896	377(7%)
IOBs	158	23(14%)
MULT 18x18s	12	5(41%)

5.4 無電流感測之實作

在本節中將以 FPGA 實作理想輸入電壓下 60Hz 及 400Hz 頻率下，輸出 200W、400W、600W 輸出，並記錄其各次輸入電壓及電流諧波值，並與 A 類及 D 類規範

或 DO-160 規範做比較，驗證其控制法則的可行性。

5.4.1 頻率 60Hz 輸入電壓下之穩態實作

在下列的穩態實作中，將輸入定為 60Hz 理想電壓 $110V_{rms}$ ，此時輸出端負載接電阻 200Ω 、 100Ω 及 66.67Ω 使輸出分別為 200W、400W 及 600W，此時利用示波器量測輸入電壓及輸入電流，如圖 5.8 至圖 5.10，並透過功率計記錄各次電流諧波的大小值，並整理成下表 5.2。

表 5.2 60Hz 實作輸入電壓下各次諧波及其 THD_i

輸出功率		200W	400W	600W
諧波次數				
1-基本波 (單位：A)		1.7616	3.4403	5.2297
奇次諧波	3	0.1575	0.2844	0.5743
	5	0.0551	0.0389	0.0844
	7	0.0147	0.0368	0.0520
	9	0.0145	0.0258	0.0338
	11	0.0162	0.0251	0.0235
	13	0.0169	0.0190	0.0158
	15	0.0195	0.0209	0.0139
	17	0.0185	0.0157	0.0111
	19	0.0136	0.0133	0.0085
偶次諧波	2	0.0718	0.0460	0.1069
	4	0.0420	0.0253	0.0284
	6	0.0380	0.0353	0.0238
	8	0.0214	0.0243	0.0177
	10	0.0166	0.0183	0.0138
	12	0.0093	0.0086	0.0125
	14	0.0099	0.0112	0.0070
	16	0.0040	0.0021	0.0057
	18	0.0027	0.0035	0.0040
	20	0.0029	0.0012	0.0029
THD_i (%)		11.27%	8.80%	11.41%

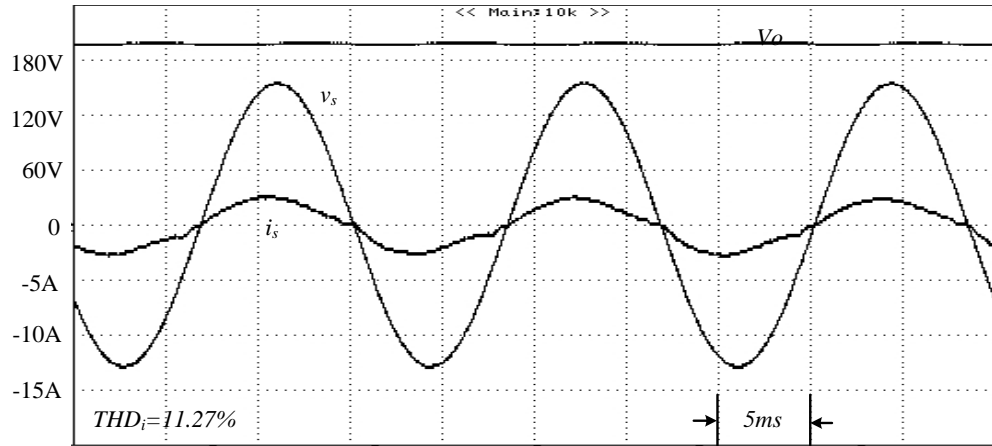


圖 5.8 200W 輸出 60Hz 實作輸入電壓及電流波形

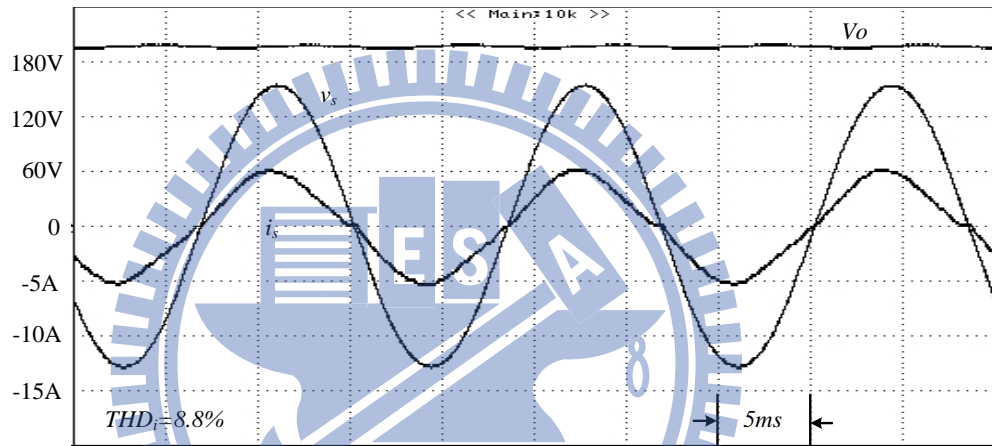


圖 5.9 400W 輸出 60Hz 實作輸入電壓及電流波形

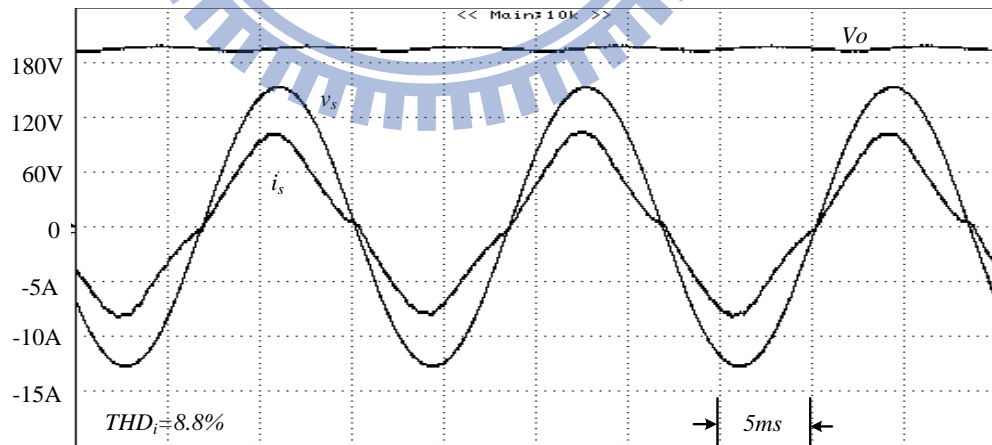


圖 5.10 600W 輸出 60Hz 實作輸入電壓及電流波形

5.4.2 頻率 400Hz 輸入電壓下之穩態實作

此時輸入電源改為 400Hz 理想電壓 $110V_{\text{rms}}$ ，輸出端負載接電阻 200Ω 、 100Ω 及 66.67Ω 使輸出分別為 200W、400W 及 600W，此時利用示波器量測輸入電壓及輸入電流，如圖 5.11 至圖 5.13，並透過功率計記錄各次電流諧波的大小值，並整理成下表 5.3。

表 5.3 400Hz 實作輸入電壓下各次諧波及其 THD_i

輸出功率		200W	400W	600W
諧波次數				
1-基本波 (單位：A)		1.8428	3.6783	5.6295
奇次諧波	3	0.1790	0.2531	0.4722
	5	0.0680	0.0924	0.1016
	7	0.0426	0.0606	0.0795
	9	0.0196	0.0399	0.0568
	11	0.0178	0.0331	0.0521
	13	0.0118	0.0252	0.0396
	15	0.0058	0.0179	0.0319
	17	0.0070	0.0142	0.0225
	19	0.0053	0.0105	0.0193
偶次諧波	2	0.0183	0.0332	0.0409
	4	0.0322	0.0370	0.0329
	6	0.0166	0.0241	0.0198
	8	0.0115	0.0179	0.0113
	10	0.0046	0.0125	0.0088
	12	0.0023	0.0107	0.0042
	14	0.0011	0.0087	0.0031
	16	0.0005	0.0070	0.0027
	18	0.0009	0.0056	0.0014
	20	0.0002	0.0064	0.0027
THD_i (%)		11.15%	8.00%	8.95%

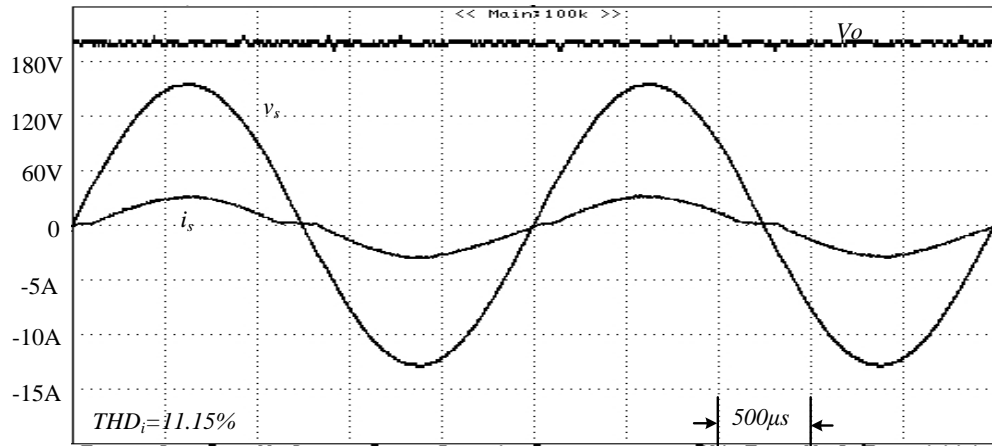


圖 5.11 200W 輸出 400Hz 實作輸入電壓及電流波形

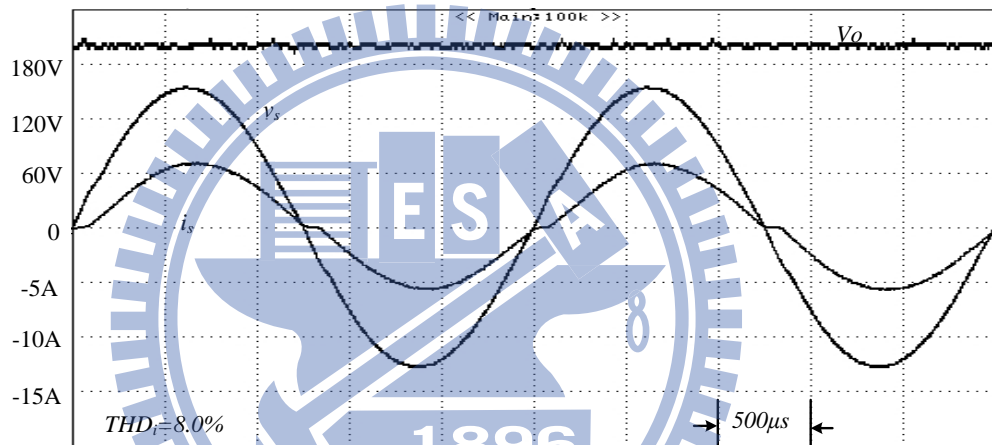


圖 5.12 400W 輸出 400Hz 實作輸入電壓及電流波形

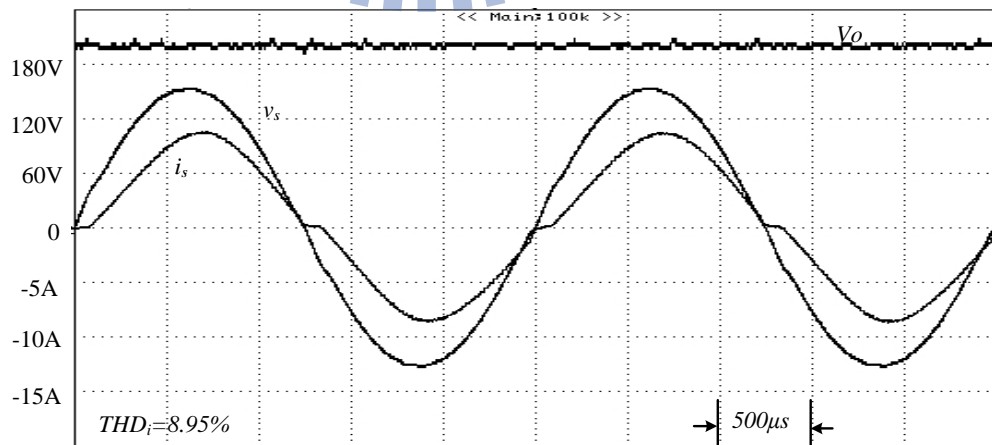


圖 5.13 600W 輸出 400Hz 實作輸入電壓及電流波形

5.4.3 頻率 60Hz 輸入電壓下之暫態實作

在原來輸出 400W 下切載至 600W，利用示波器觀察輸入電流及輸出電壓的變化，其切換負載大至在 100ms 內，輸入電流可以平穩的變化，而輸出電壓僅漣波電壓稍微增加，如下圖 5.14。

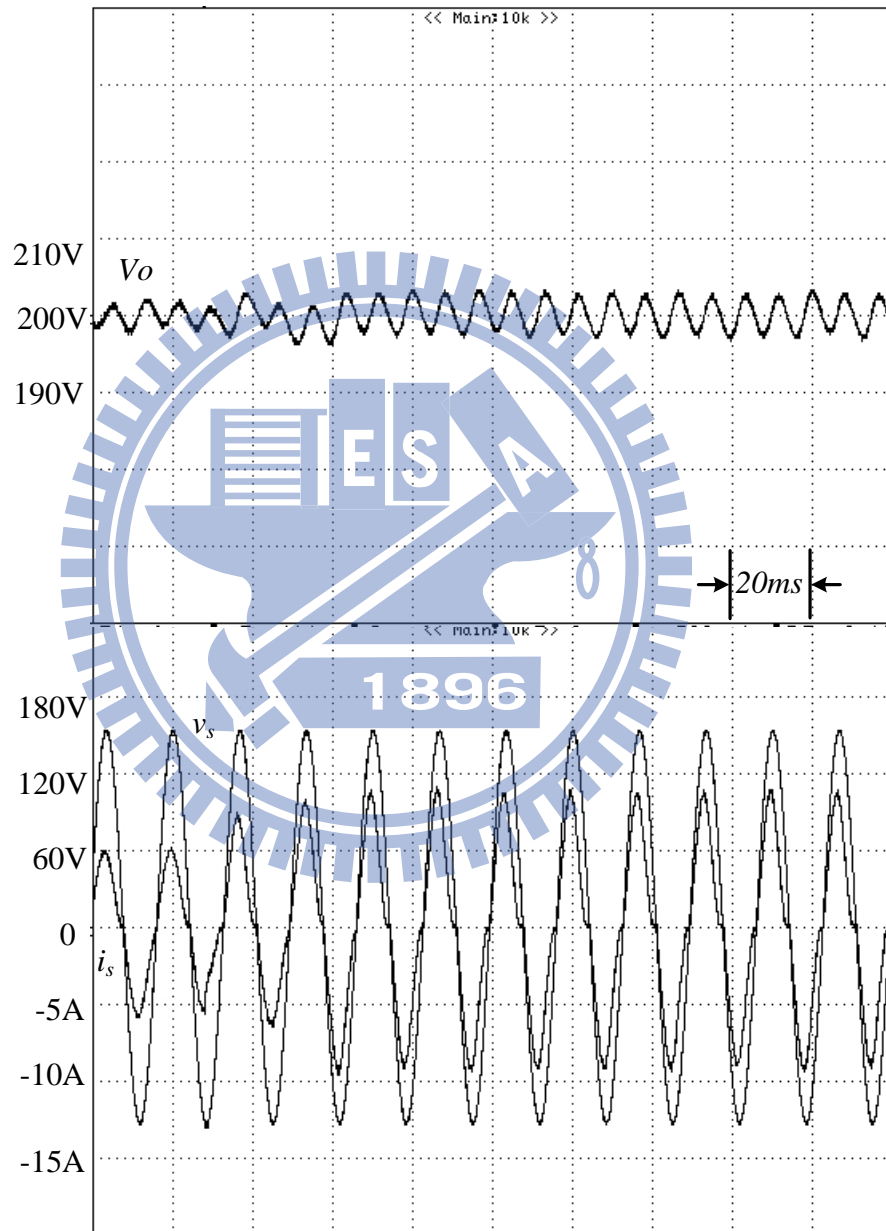


圖 5.14 60Hz 實作輸入電壓及電流之暫態響應

5.4.4 頻率 400Hz 輸入電壓下之暫態實作

如同上一小節，將輸入改為 400Hz 理想輸入電壓，輸出 400W 下切載至 600W，利用示波器觀察輸入電流及輸出電壓的變化，其切換負載大致仍在 100ms 內，如下圖 5.15。

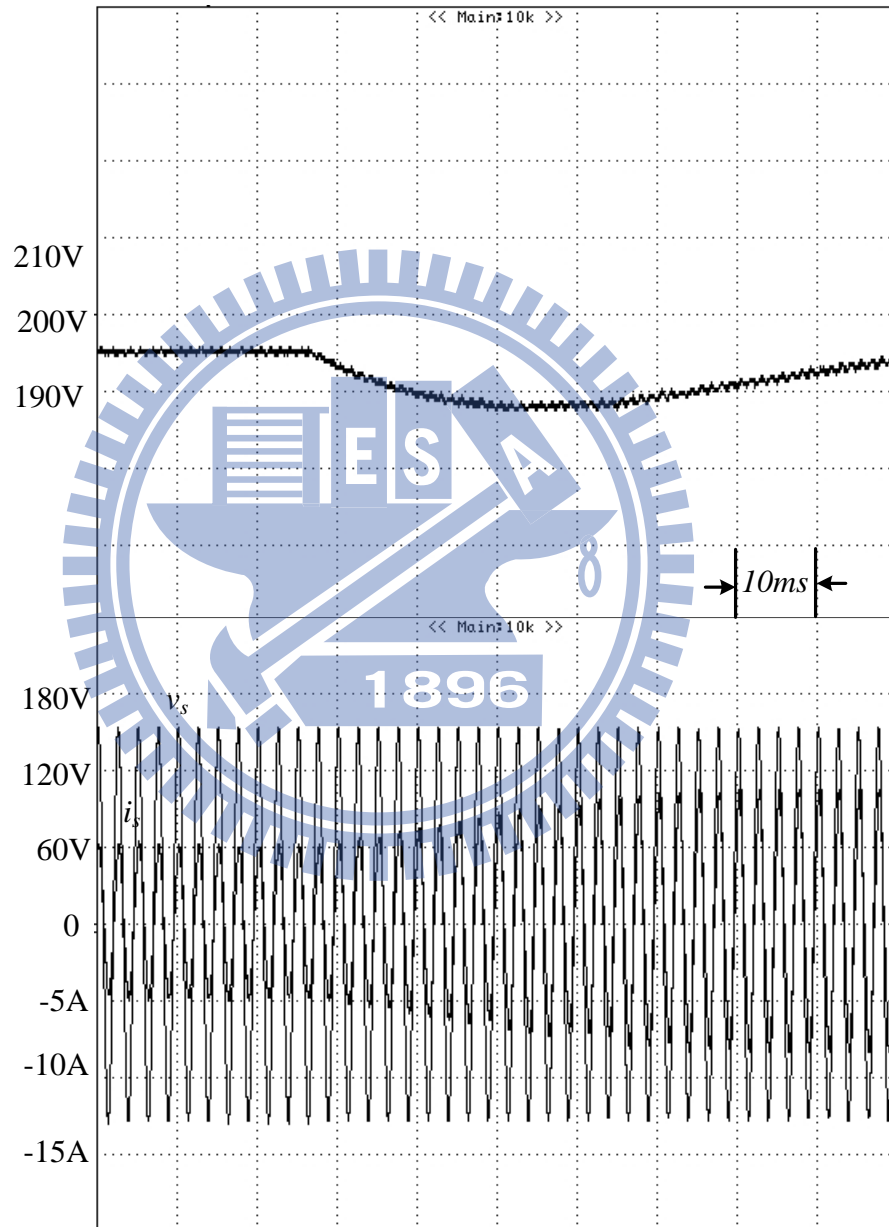


圖 5.15 400Hz 實作輸入電壓及電流之暫態響應

5.5 諧波規範比較

在下表 5.4 至 5.6 中，分列出在頻率 60Hz 理想輸入電壓下，各輸出功率下所得電流諧波成分與 IEC 61000-3-2 之 A、D 類規範做比較，均有符合此規範。而在表 5.7 中，列出了在頻率 400Hz 理想輸入電壓下，各輸出功率下所得電流諧波成分與 DO-160 規範，在偶次諧波尚未達到標準，而奇次諧波僅 3 次諧波尚未符合規範。



表 5.4 200W 輸出 60Hz 實作輸入電流與諧波規範

輸出功率		A 類規範(安培)	D 類規範 (安培)	輸入電流
諧波次數				
1-基本波 (單位：A)				1.7676
奇次諧波	3	2.30	$200 \times 0.0034 = 0.68$	0.1575
	5	1.14	$200 \times 0.0019 = 0.38$	0.0551
	7	0.77	$200 \times 0.0010 = 0.20$	0.0147
	9	0.40	$200 \times 0.0005 = 0.10$	0.0145
	11	0.33	$200 \times 0.00035 = 0.07$	0.0162
	13	0.21	$200 \times 0.000296 = 0.0592$	0.0169
	15	$0.15 \times \frac{15}{15} = 0.15$	$200 \times \frac{0.00385}{15} = 0.0513$	0.0195
	17	$0.15 \times \frac{15}{17} = 0.1324$	$200 \times \frac{0.00385}{17} = 0.0453$	0.0185
	19	$0.15 \times \frac{15}{19} = 0.1184$	$200 \times \frac{0.00385}{19} = 0.0405$	0.0136
偶次諧波	2	1.08		0.0718
	4	0.43		0.0420
	6	0.30		0.0380
	8	$0.23 \times \frac{8}{8} = 0.23$		0.0214
	10	$0.23 \times \frac{8}{10} = 0.184$		0.0166
	12	$0.23 \times \frac{8}{12} = 0.1533$		0.0093
	14	$0.23 \times \frac{8}{14} = 0.1314$		0.0099
	16	$0.23 \times \frac{8}{16} = 0.115$		0.0040
	18	$0.23 \times \frac{8}{18} = 0.1022$		0.0027
	20	$0.23 \times \frac{8}{20} = 0.092$		0.0029
$THD_i(\%)$				11.27%

表 5.5 400W 輸出 60Hz 實作輸入電流與諧波規範

輸出功率 諧波次數		A 類規範(安培)	D 類規範(安培)	輸入電流
1-基本波 (單位：A)				3.4403
奇次諧波	3	2.30	$400 \times 0.0034 = 1.36$	0.2844
	5	1.14	$400 \times 0.0019 = 0.76$	0.0389
	7	0.77	$400 \times 0.0010 = 0.40$	0.0368
	9	0.40	$400 \times 0.0005 = 0.20$	0.0258
	11	0.33	$400 \times 0.00035 = 0.14$	0.0251
	13	0.21	$400 \times 0.000296 = 0.1184$	0.0190
	15	$0.15 \times \frac{15}{15} = 0.15$	$400 \times \frac{0.00385}{15} = 0.1027$	0.0209
	17	$0.15 \times \frac{15}{17} = 0.1324$	$400 \times \frac{0.00385}{17} = 0.0906$	0.0157
	19	$0.15 \times \frac{15}{19} = 0.1184$	$400 \times \frac{0.00385}{19} = 0.0811$	0.0133
偶次諧波	2	1.08		0.0460
	4	0.43		0.0253
	6	0.30		0.0353
	8	$0.23 \times \frac{8}{8} = 0.23$		0.0243
	10	$0.23 \times \frac{8}{10} = 0.184$		0.0183
	12	$0.23 \times \frac{8}{12} = 0.1533$		0.0086
	14	$0.23 \times \frac{8}{14} = 0.1314$		0.0112
	16	$0.23 \times \frac{8}{16} = 0.115$		0.0021
	18	$0.23 \times \frac{8}{18} = 0.1022$		0.0035
	20	$0.23 \times \frac{8}{20} = 0.092$		0.0012
THD_i (%)				8.80%

表 5.6 600W 輸出 60Hz 實作輸入電流與諧波規範

輸出功率 諧波次數		A 類規範(安培)	D 類規範(安培)	輸入電流
1-基本波 (單位：A)				5.2297
奇次諧波	3	2.30	$600 \times 0.0034 = 2.04$	0.5743
	5	1.14	$600 \times 0.0019 = 1.14$	0.0844
	7	0.77	$600 \times 0.0010 = 0.60$	0.0520
	9	0.40	$600 \times 0.0005 = 0.30$	0.0338
	11	0.33	$600 \times 0.00035 = 0.21$	0.0235
	13	0.21	$600 \times 0.000296 = 0.1776$	0.0158
	15	$0.15 \times \frac{15}{15} = 0.15$	$600 \times \frac{0.00385}{15} = 0.154$	0.0139
	17	$0.15 \times \frac{15}{17} = 0.1324$	$600 \times \frac{0.00385}{17} = 0.1359$	0.0111
	19	$0.15 \times \frac{15}{19} = 0.1184$	$600 \times \frac{0.00385}{19} = 0.1216$	0.0085
偶次諧波	2	1.08		0.1069
	4	0.43		0.0284
	6	0.30		0.0238
	8	$0.23 \times \frac{8}{8} = 0.23$		0.0177
	10	$0.23 \times \frac{8}{10} = 0.184$		0.0138
	12	$0.23 \times \frac{8}{12} = 0.1533$		0.0125
	14	$0.23 \times \frac{8}{14} = 0.1314$		0.0070
	16	$0.23 \times \frac{8}{16} = 0.115$		0.0057
	18	$0.23 \times \frac{8}{18} = 0.1022$		0.0040
	20	$0.23 \times \frac{8}{20} = 0.092$		0.0029
THD_i (%)				

表 5.7 200W 輸出 400Hz 實作輸入電流與諧波規範

輸出功率：200(W)					
基本波：1.8428(A)					
DO-160 奇次諧波規範		電流	DO-160 偶次諧波規範		電流
3	$0.15 \times \frac{1.8428}{3} = 0.0921$	0.1790	2	$0.01 \times \frac{1.8428}{2} = 9.214e^{-3}$	0.0183
5	$0.3 \times \frac{1.8428}{5} = 0.1106$	0.0680	4	$0.01 \times \frac{1.8428}{4} = 4.607e^{-3}$	0.0322
7	$0.3 \times \frac{1.8428}{7} = 0.0790$	0.0426	6	$0.0025 \times \frac{1.8428}{6} = 767.83e^{-6}$	0.0166
9	$0.15 \times \frac{1.8428}{9} = 0.0307$	0.0196	8	$0.0025 \times \frac{1.8428}{8} = 575.88e^{-6}$	0.0115
11	$0.3 \times \frac{1.8428}{11} = 0.0503$	0.0178	10	$0.0025 \times \frac{1.8428}{10} = 460.7e^{-6}$	0.0046
13	$0.3 \times \frac{1.8428}{13} = 0.0425$	0.0118	12	$0.0025 \times \frac{1.8428}{12} = 383.92e^{-6}$	0.0023
15	$0.15 \times \frac{1.8428}{15} = 0.0186$	0.0058	14	$0.0025 \times \frac{1.8428}{14} = 329.07e^{-6}$	0.0011
17	$0.3 \times \frac{1.8428}{17} = 0.0325$	0.0070	16	$0.0025 \times \frac{1.8428}{16} = 287.94e^{-6}$	0.0005
19	$0.3 \times \frac{1.8428}{19} = 0.0291$	0.0053	18	$0.0025 \times \frac{1.8428}{18} = 255.94e^{-6}$	0.0009
$THD_i(\%)$			11.15%		

表 5.8 400W 輸出 400Hz 實作輸入電流與諧波規範

輸出功率：400(W)					
基本波：3.6783(A)					
DO-160 奇次諧波規範		電流	DO-160 偶次諧波規範		電流
3	$0.15 \times \frac{3.6783}{3} = 0.1839$	0.2531	2	$0.01 \times \frac{3.6783}{2} = 0.0184$	0.0332
5	$0.3 \times \frac{3.6783}{5} = 0.2207$	0.0924	4	$0.01 \times \frac{3.6783}{4} = 9.196e^{-3}$	0.0370
7	$0.3 \times \frac{3.6783}{7} = 0.1576$	0.0606	6	$0.0025 \times \frac{3.6783}{6} = 1532.63e^{-6}$	0.0241
9	$0.15 \times \frac{3.6783}{9} = 0.0613$	0.0399	8	$0.0025 \times \frac{3.6783}{8} = 1149.47e^{-6}$	0.0179
11	$0.3 \times \frac{3.6783}{11} = 0.1003$	0.0331	10	$0.0025 \times \frac{3.6783}{10} = 919.58e^{-6}$	0.0125
13	$0.3 \times \frac{3.6783}{13} = 0.0849$	0.0252	12	$0.0025 \times \frac{3.6783}{12} = 766.31e^{-6}$	0.0107
15	$0.15 \times \frac{3.6783}{15} = 0.0368$	0.0179	14	$0.0025 \times \frac{3.6783}{14} = 656.84e^{-6}$	0.0087
17	$0.3 \times \frac{3.6783}{17} = 0.0649$	0.0142	16	$0.0025 \times \frac{3.6783}{16} = 574.73e^{-6}$	0.0070
19	$0.3 \times \frac{3.6783}{19} = 0.0581$	0.0105	18	$0.0025 \times \frac{3.6783}{18} = 510.86e^{-6}$	0.0056
THD_i (%)			8.00%		

表 5.9 600W 輸出 400Hz 實作輸入電流與諧波規範

輸出功率：600(W)					
基本波：5.6295(A)					
DO-160 奇次諧波規範		電流	DO-160 偶次諧波規範		電流
3	$0.15 \times \frac{5.6295}{3} = 0.2815$	0.4722	2	$0.01 \times \frac{5.6295}{2} = 0.0281$	0.0409
5	$0.3 \times \frac{5.6295}{5} = 0.3378$	0.1016	4	$0.01 \times \frac{5.6295}{4} = 0.0141$	0.0329
7	$0.3 \times \frac{5.6295}{7} = 0.2413$	0.0795	6	$0.0025 \times \frac{5.6295}{6} = 9.383e^{-3}$	0.0198
9	$0.15 \times \frac{5.6295}{9} = 0.0938$	0.0568	8	$0.0025 \times \frac{5.6295}{8} = 7.037e^{-3}$	0.0113
11	$0.3 \times \frac{5.6295}{11} = 0.1535$	0.0521	10	$0.0025 \times \frac{5.6295}{10} = 5.630e^{-3}$	0.0088
13	$0.3 \times \frac{5.6295}{13} = 0.1299$	0.0396	12	$0.0025 \times \frac{5.6295}{12} = 1.173e^{-3}$	0.0042
15	$0.15 \times \frac{5.6295}{15} = 0.0563$	0.0319	14	$0.0025 \times \frac{5.6295}{14} = 1.005e^{-3}$	0.0031
17	$0.3 \times \frac{5.6295}{17} = 0.0993$	0.0225	16	$0.0025 \times \frac{5.6295}{16} = 879.61e^{-6}$	0.0027
19	$0.3 \times \frac{5.6295}{19} = 0.0889$	0.0193	18	$0.0025 \times \frac{5.6295}{18} = 781.88e^{-6}$	0.0014
THD_i (%)			8.95%		

5.6 實驗改善方向

本論文未將針對實作的誤差，提出幾點本實驗可以改善的方向：

1. 本論文實驗提出的無電流感測控制，僅靠二個 A/D 轉換電路感測輸入電壓及輸出電壓，故此 A/D 轉換電路的可靠度顯得相當的重要，無論是感測出的大小或是頻寬不足造成時間的誤差，將影響控制的效果。
2. 所提出的控制法中，補償開關及二極體不理想特性的壓降，是假設一定值，但其壓降的大小會隨著溫度而有所改變，其補償的大小應有所不同，故實驗時半導體元件散熱也會影響實驗的結果。



第六章

結論

總結以上實驗歸納，若是採用本系統有以下優點：

1. 本電路結構簡單，相較於傳統之電流控制迴路減少了一個控制迴圈，因此可降低電路複雜造成的誤差。
2. 因取消了電流的控制迴路，減少了電流感測器，因此可以有相較於電流控制迴路控制更便宜的電路費用。
3. 在本控制推導的方法，亦可用在頻率較高的範圍，且仍保持為理想弦波，使應用範圍廣泛。

亦有缺點存在：

由於無感測輸入電流，因此我們必須得知電感的感值以及內阻值來加以補償，其精確程度將與 PFC 效果呈正比。



參考文獻

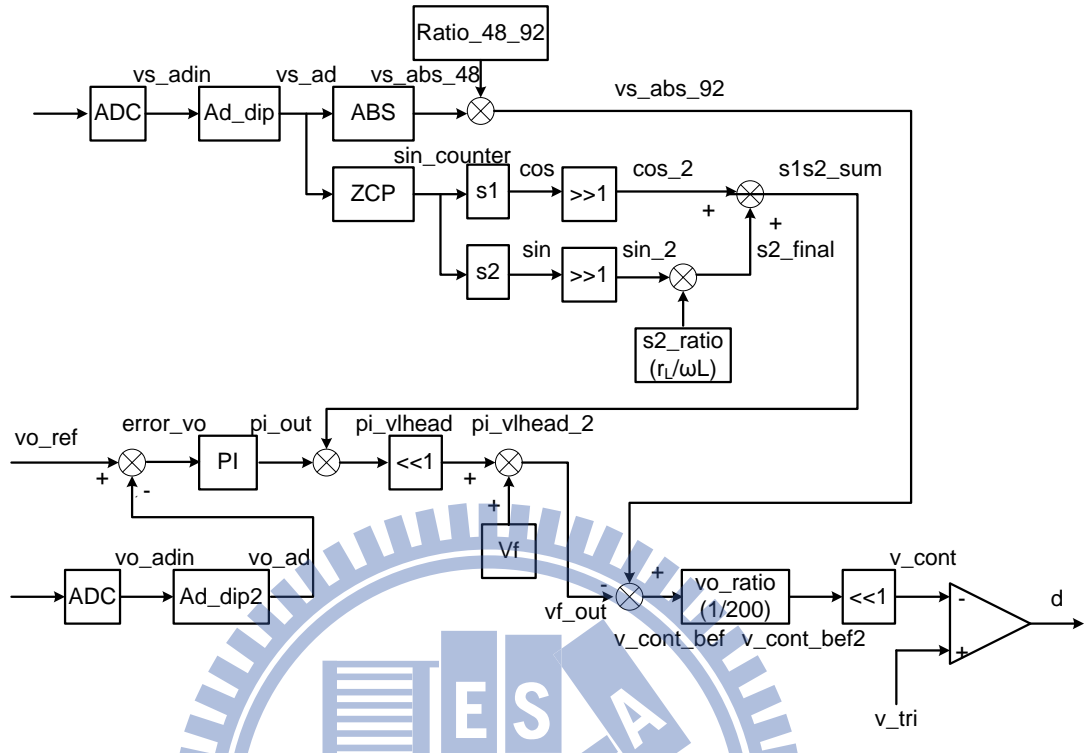
- [1] H. C. Chen, "Duty phase control for single-phase boost-type SMR," *IEEE Trans. on Power Electron*, vol. 23, no. 4, pp. 1927-1934, Jul. 2008.
- [2] H. C. Chen, "Single-Loop Current Sensorless Control for Single-Phase Boost-Type SMR," *IEEE Trans. on Power Electron*, vol. 24, no. 1, Jan. 2009.
- [3] David M. Van de Sype; Koen De Gussemé; Alex P. M. Van den Bossche; Jan A. Melkebeek, "Duty-Ratio Feedforward for Digitally Controlled Boost PFC Converters," *IEEE Trans on Industrial Electron*, vol. 52, no. 1, pp 108-115, Feb. 2005.
- [4] PSIM WEBSITE---<http://www.powersimtech.com>
- [5] Xilinx, Spartan-3 FPGA Family Data Sheet, 2008.
- [6] Laszlo Huber; Yungtaek Jang; Milan M. Jovanovic, "Performance evaluation of bridgeless PFC boost rectifier," *IEEE Trans on Power Electron*, vol 23, no. 3, May 2008.
- [7] Yungtaek Jang; Milan M. Jovanovic, "A bridgeless PFC boost rectifier with optimized magnetic utilization," *IEEE Trans on Power Electron*, vol. 24, no. 1, Jan 2009.
- [8] "IEEE Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems," *IEEE Std 519-1992*
- [9] Jian Sun; Min Chen; Kamiar J. Karimi, "Aircraft Power System Harmonics Involving Single-Phase PFC Converters," *IEEE Trans on Aerospace and Electron*, vol. 44, no. 1, pp 217-226, Jan 2008.
- [10] Jian Sun, "Analysis and Design of Single-Phase PFC Converters for Airborne Systems," IEEE, 2003
- [11] J. Rajagopalan, F. C. Lee, and P. Nora, "A general technique for derivation of

average current mode control laws for single-phase power-factor correction circuits without input voltage sensing,” *IEEE Trans. on Power Electron.*, vol. 14,no. 4, pp. 663-672, Jul. 1999.

- [12] K. Rustom, and I. Batarseh, “Recent Advances in Single-Stage Power Factor Correction,” *IEEE International Conference 2003*, pp. 1089-1095, vol. 12.



附錄一、程式控制方塊圖



附錄二、程式碼

主程式(pfc.v):

```

module pfc(clk_40MHz,
           da_clk_1, la_1, out_1, out_2, out_3, out_4, //for DA
           ad_clk_1, sfrm_1, convst_1, sdata_1, sdata_2, dipsw7, //for AD
           dipsw2, dipsw3, dipsw4, //for KI
           dipsw5, dipsw6, //for KP
           dipsw1, //for PI
           //clk_40k_1, t, //Test Freq
           duty, an, bn, startsw); //for tosw

```

```

input startsw;
input clk_40MHz;
//output clk_40k_1;
wire [15:0] divider_10M, divider_4M, divider_20k, divider_20M, divider_40k;
wire clk_10MHz, clk_4MHz, clk_4kHz, clk_40kHz;
output duty;
//assign clk_40k_1=clk_40kHz;
//output t;

```

```

//除頻器,因為是正緣觸發,本身就除頻為20MH,再除以(divider+1)
assign divider_10M=16'd1;
assign divider_4M=16'd4;
assign divider_20k=16'd999;
assign divider_20M=16'd0;
assign divider_40k=16'd499;

```

```

//4 組 DA, 硬體電路上的 clk 和 la 腳位共用, 故 la_2, la_3, la_4 都設為 la_1, clk 亦同
output da_clk_1, la_1, out_1, out_2, out_3, out_4;
wire la_2, la_3, la_4; //配合程式 DA Module 使用
wire da_clk_2, da_clk_3, da_clk_4; //配合程式 DA Module 使用
wire [15:0] in_1, in_2, in_3, in_4;
assign la_2=la_1;
assign la_3=la_1;
assign la_4=la_1;
assign da_clk_2=da_clk_1;
assign da_clk_3=da_clk_1;
assign da_clk_4=da_clk_1;
assign da_clk_1=clk_10MHz;

//2 組 AD, 硬體電路上 sfrm, convst 和 clk 腳位共用
wire adsw; //模組內有設一個開關當 ad 的 switch
assign adsw=1'b0; //adsw=0 時, AD 程式才作用
input sfrm_1, sdata_1, sdata_2;
output ad_clk_1, convst_1;
wire [15:0] vo_adin, vs_adin; //原始 AD 硬體轉換出來的值
wire sfrm_2, ad_clk_2, convst_2; //因 sfrm, convst, adclk 硬體共用
wire [15:0] vo_ad, vs_ad; //修正後的值
assign sfrm_2=sfr_1;
assign convst_2=convst_1;
assign ad_clk_2=ad_clk_1;
assign ad_clk_1=clk_4MHz;
//assign vo_ad=vo_adin+16'd1224; //補償 MAX121 的 offset, 輸入 0V 時, AD 值 FB38=1224
//assign vs_ad=vs_adin+16'd788; //補償 MAX121 的 offset, 輸入 0V 時, AD 值 FCEC=788

//誤差放大器
wire [15:0] vo_ref;
wire [15:0] error_vo;
assign vo_ref=16'd13474; //Vo=200V 時, vo_ref=200/92/5*32767=14247,
//Vo=200V 時, 60Hz 設 13900 輸出會變低, 但 THD 會變好
//Vo=200V 時, 400Hz 200W 設 13774, 400W 設 13474 波型較佳, 600W 設 13574

//PI
input dipsw1; //為 PI 控制器的開關
input dipsw2, dipsw3, dipsw4; //用指撥開關的 3bit module 可以調 KI, KP 或 AD 值
input dipsw5, dipsw6, dipsw7; //同上
wire [15:0] kp;
wire [15:0] ki;
wire [15:0] pi_out;
assign kp=16'd50; //待調整
assign ki=16'd50; //待調整

//zcp
wire vs_plus; //當 Vin 為正時, vs_plus=0
wire [8:0] sin_counter;
//sin_table
wire [15:0] sin, cos, sin_2, cos_2;
wire [15:0] s2_final;
wire [15:0] s2_final2; //s2_final=s2*rL/WL

```



```

wire [15:0] s1s2_sum;
wire [15:0] s2_ratio;
wire [2:0] shift_bit1, shift_bit2;
assign s2_ratio=16'd1862; //s2_ratio=rL/WL=0.5/(2*pi*60*4.6*e-3)*32767=9447
//for 1L 60Hz
//s2_ratio=0.3/(2*pi*60*2.6*e-3)*32767=10028 for 2L 60Hz
//s2_ratio=0.3/(2*pi*60*2.1*e-3)*32767=12417 for 2L 60Hz
//s2_ratio=0.3/(2*pi*400*2.6*e-3)*32767=1504 for 2L 400Hz
//s2_ratio=0.3/(2*pi*400*2.1*e-3)*32767=1862 for 2L 400Hz
//s2_ratio=0.3/(2*pi*400*0.28*.e-3)*32767=13968 for 2L 400Hz
//s2_ratio=0.3/(2*pi*400*0.88*e-3)*32767=4444 for 2L 400Hz

assign s2_final2=~s2_final+16'd1; //配合減法器當加法器使用
assign shift_bit1=3'd1; //本查表為最大值 32767 當實際值 1, 但 s1+s2>>1 出現,
//故先又移 1 位除以 2
assign shift_bit2=3'd1; //同上

//vcont 查表, 用查表當 Vcont
//wire [15:0] vcont_200v, vcont_40v, vcont_200v_dis, vcont_40v_dis;

//乘法器
wire [15:0] pi_vlhead, pi_vlhead_2;
wire [15:0] vs_abs_48, vs_abs_92;
wire [15:0] ratio_48_92;
wire [15:0] vf_out;
wire [15:0] v_cont_bef, v_cont_bef2, v_cont, v_cont2;
wire [15:0] vf_comp1, vf_comp2;
wire [2:0] shift_bit3, shift_bit4;
wire [15:0] vo_ratio;
wire duty;
assign vf_comp1=duty?16'd228:16'd199; //導通時壓降為 1.8+1.4, 截止時壓降為
//1.4+1.4, vf=壓降/92/5*32767=202(截止)
assign vf_comp2=~vf_comp1+16'd1; //配合減法器當加法器使用
assign ratio_48_92=16'd17096; //48/92*32767=17096
assign shift_bit3=3'd1;
assign shift_bit4=3'd2;
assign vo_ratio=16'd18841; //vo_ratio=2.3/4*32737=18841, 2.3=92*5/200, 需左移 2bit
//若輸出為 40V, vo_ratio=92*5/40=11.5, 則需左移 4bit, 再乘上 11.5/16*32767=23551
//wire [8:0] sin_counter2;
//assign sin_counter2=sin_counter+9'd20;
//將 ZCP 偵測出的零交越點 shift 個值

wire [8:0] triangle_1, triangle_2, vcont;

//tosw
output an, bn;
clk_conversion clk_10M (clk_40MHz, divider_10M, clk_10MHz); //20MHz/(divider+1)
clk_conversion clk_4M (clk_40MHz, divider_4M, clk_4MHz);
clk_conversion clk_20k (clk_40MHz, divider_20k, clk_20kHz);
clk_conversion clk_20M (clk_40MHz, divider_20M, clk_20MHz);
clk_conversion clk_40k (clk_40MHz, divider_40k, clk_40kHz);

ad ad_1(ad_clk_1, sdata_1, vo_adin, any1, convst_1, sfrm_1, adsw, t1);

```

```

//輸出 vo_ad 為 16bit 值，此 AD 硬體輸入，經取樣轉成 14bit 的值，
//但輸出傳回成 16bit，後面會用到
ad ad_2(ad_clk_2, sdata_2, vs_adin, any2, convst_2, sfrm_2, adsw, t2);
//輸出 vo_ad 為 16bit 值，此 AD 硬體輸入，經取樣轉成 14bit 的值，
//但輸出傳回成 16bit，後面會用到

ad_dip ad_regulator(clk_40kHz, vs_adin, vs_ad, dipsw7, dipsw6, dipsw5, vs_plus);
//判斷輸入電壓 vs 正負，為正時 vs_plus=0，利用指撥開關 5~7 補償一定的數值，調整後為 vs_ad
ad_dip2 ad_regulator2(clk_40kHz, vo_adin, vo_ad, dipsw4, dipsw3, dipsw2);
//利用指撥開關 2~4 調整 vo_adin，調整後為 vo_ad

da da_1(la_1, clk_10MHz, out_1, in_1); //DA 出來只有正負 3V，再經過硬體電路放大 3.3 倍
da da_2(la_2, clk_10MHz, out_2, in_2);
da da_3(la_3, clk_10MHz, out_3, in_3);
da da_4(la_4, clk_10MHz, out_4, in_4);

subtraction voltage(vo_ref, vo_ad, error_vo);
//減法器為 vo_ref-vo_adin(或 vo_ad)，同 16bit
pi voltage_controller(error_vo, pi_out, kp, ki, clk_40kHz, dipsw1);
//pi_out=VL_head, dipsw1 為 PI 的開關

zcp vin_zcp(clk_40kHz, vs_plus, vs_ad, sin_counter);
//零交越點偵測電路，為了要查表用，輸入為 vs_adin(或 vs_ad)

Hz_sin_table sin_list(sin, cos, sin_counter); //可得到 sin 和 cos 查表值
//60Hz 使用 sin_table, 400Hz 使用 Hz_sin_table
shift_R s1(clk_40kHz, shift_bit1, cos, cos_2); //左 shift_bit1 數，最多移 4 位
shift_R s2(clk_40kHz, shift_bit2, sin, sin_2);
multiplication s2_multi(sin_2, s2_ratio, s2_final); //s2_final=sin_2*s2_ratio
//1/60 sec 用 40kHz 取樣的話，會有 666 點
//s2_final=s2*rL/WL 值, s2_ratio 為 rL/WL=0.5/(2*pi*60*4.6*e-3)*32767

subtraction s1s2_sum_module(cos_2, s2_final2, s1s2_sum);
//s2_final2 為 s2_final 取負值，配合減法器取代加法器
//因此 s1s2_sum=cos_2+s2_final1

multiplication pi_compensate(pi_out, s1s2_sum, pi_vlhead);
//pi_vlhead=pi_out*s1s2_sum

shift_L pi_vlhead_shift(clk_40kHz, shift_bit3, pi_vlhead, pi_vlhead_2);
//pi_vlhead_2 為 pi_vlhead 左移(shift_bit3)位數

subtraction vf_compensate(pi_vlhead_2, vf_comp2, vf_out);
//vf_out 為 pi_vlhead_2+vf_comp1，vf+comp1 為導通或截止的補償值

absolute vs_voltage(clk_40kHz, vs_ad, vs_abs_48);
//vs_abs_48 為 vs 電壓降為 1/48 之 AD 值
multiplication vs_voltage_multi(vs_abs_48, ratio_48_92, vs_abs_92);
//vs_abs_92 為 vs 電壓轉為 1/92
//cont_table control(vcont_200v, vcont_40v, sin_counter);
//用查表當 vcont

subtraction v_cont_before(vs_abs_92, vf_out, v_cont_bef);

```

```

//v_cont_bef=vs_abs_92-vf_out

multiplication v_cont_multi(v_cont_bef, vo_ratio, v_cont_bef2);
//v_cont_bef2=v_cont_bef*vo_ratio

shift_L v_cont_shift(clk_40kHz, shift_bit4, v_cont_bef2, v_cont);
//v_cont=v_cont_bef2 左移(shift_bit4)位數

latch latch_1(v_cont, v_cont2, clk_40kHz);
triangle tr(v_cont2, duty, duty2, clk_40MHz, startsw, triangle_1, triangle_2, vcont);
// clk/1000 為三角波的頻率, vcont=v_cont 取 9bit

tosw tosw1(duty, vs_plus, an, bn, startsw);
//startsw=0 時, 開關訊號才會送出

//ki_dip ki_ki(clk_40kHz, ki, dipsw4, dipsw3, dipsw2);
//kp_dip kp_kp(clk_40kHz, kp, dipsw4, dipsw3, dipsw2);

//DA 要顯示的
assign in_1=vs_ad;
assign in_2=cos;
assign in_3=sin;
assign in_4=pi_vlhead;

endmodule

```

減法器(subtraction.v):

```

module subtraction(minuend, subtrahend, error);
input [15:0] minuend, subtrahend;
output reg [15:0] error;
reg [1:0] sign_bit;
always @(minuend, subtrahend)
begin
    sign_bit[1]=minuend[15];
    sign_bit[0]=subtrahend[15];
    case(sign_bit)
    2' b00:
        error=minuend-subtrahend;
    2' b10:
        begin
            error=minuend-subtrahend;
            if(error[15]==0)
                error=16' b1000_0000_0000_0000;
        end
    2' b11:
        error=minuend-subtrahend;
    2' b01:
        begin
            error=minuend-subtrahend;
            if(error[15]==1)
                error=16' b0111_1111_1111_1111;
        end
    endcase
end

```

```

end
endmodule

```

乘法器(multiplication.v):

```

module multiplication (multiplicand,multiplier,answer);
input [15:0] multiplicand,multiplier;
output reg [15:0] answer;
reg [15:0] multiplicand_p,multiplier_p;
reg [31:0] answer_temp;
reg [1:0] sign_bit;
always @(multiplicand,multiplier)
begin
    if(multiplicand[15])
        begin
            if(multiplicand==16'b1000_0000_0000_0000)
                multiplicand_p=16'b0111_1111_1111_1111;
            else
                multiplicand_p=~multiplicand+16'd1;
            end
        else
            multiplicand_p=multiplicand;
        if(multiplier[15])
            begin
                if(multiplier==16'b1000_0000_0000_0000)
                    multiplier_p=16'b0111_1111_1111_1111;
                else
                    multiplier_p=~multiplier+16'd1;
                end
            else
                multiplier_p=multiplier;

            sign_bit[0] = multiplicand[15];
            sign_bit[1] = multiplier[15];
            answer_temp=(multiplicand_p*multiplier_p)<<1;
            case (sign_bit)
            2'b00:
                answer=(answer_temp)>>16;
            2'b01:
                answer=(~answer_temp+1)>>16;
            2'b10:
                answer=(~answer_temp+1)>>16;
            2'b11:
                answer=(answer_temp)>>16;
            endcase
        end
    end
endmodule

```

PI 控制器(PI.v):

```

module pi(error,PI_out,kp,ki,clk,start);
input start,clk;

```

```

input [15:0] error, kp, ki;
output [15:0] PI_out;
reg [15:0] y;
reg [2:0] contr_win;
reg [15:0] kp_c;
reg [15:0] ki_c;
reg [31:0] multi_temp_kp;
reg [31:0] multi_temp_ki;
reg [31:0] sum_temp_kp;
reg [31:0] sum_temp_ki;
reg [15:0] Duty_PI;
reg buffer;
wire [15:0] PI_out;
always @(negedge clk)
begin
    if(!start)
        begin
            if(!error[15])
                y=error;
            else
                begin
                    if(error==16' b1000_0000_0000_0000)
                        y=16' b0111_1111_1111_1111;
                    else
                        y=~error+16' d1;
                    end
                if(ki[15]==1)
                    begin
                        if(ki==16' b1000_0000_0000_0000)
                            ki_c=16' b0111_1111_1111_1111;
                        else
                            ki_c=~ki+16' d1;
                        end
                    else
                        ki_c=ki;
                if(kp[15]==1)
                    begin
                        if(kp==16' b1000_0000_0000_0000)
                            kp_c=16' b0111_1111_1111_1111;
                        else
                            kp_c=~kp+16' d1;
                        end
                    else
                        kp_c=kp;
                contr_win[0] = error[15];
                contr_win[1] = ki[15];
                contr_win[2] = kp[15];
                multi_temp_kp=kp_c*y;
                multi_temp_ki=ki_c*y;
                multi_temp_kp=multi_temp_kp<<1;
                multi_temp_ki=multi_temp_ki<<1;
                case (contr_win)

```

```

3' b001:
    begin
        multi_temp_kp=~multi_temp_kp+1;
        multi_temp_ki=~multi_temp_ki+1;
    end
3' b011:
    begin
        multi_temp_kp=~multi_temp_kp+1;
        multi_temp_ki=multi_temp_ki;
    end
3' b010:
    begin
        multi_temp_kp=multi_temp_kp;
        multi_temp_ki=~multi_temp_ki+1;
    end
3' b000:
    begin
        multi_temp_kp=multi_temp_kp;
        multi_temp_ki=multi_temp_ki;
    end
3' b100:
    begin
        multi_temp_kp=~multi_temp_kp+1;
        multi_temp_ki=multi_temp_ki;
    end
3' b101:
    begin
        multi_temp_kp=multi_temp_kp;
        multi_temp_ki=~multi_temp_ki+1;
    end
3' b110:
    begin
        multi_temp_kp=~multi_temp_kp+1;
        multi_temp_ki=~multi_temp_ki+1;
    end
3' b111:
    begin
        multi_temp_kp=multi_temp_kp;
        multi_temp_ki=multi_temp_ki;
    end
endcase
buffer=sum_temp_ki[31];
sum_temp_ki=sum_temp_ki+multi_temp_ki;
if((multi_temp_ki[31]==0) & (buffer==0))
    begin
        if (sum_temp_ki[31]==1)
            sum_temp_ki=32'h7FFF_FFFF;
        end
    if ((multi_temp_ki[31]==1) & (buffer==1))
        begin
            if (sum_temp_ki[31]==0)
                sum_temp_ki=32'h8000_0000;
            end
        end
end

```

```

sum_temp_kp=sum_temp_ki+multi_temp_kp;
if((sum_temp_ki[31]==0) & (multi_temp_kp[31]==0))
begin
if (sum_temp_kp[31]==1)
sum_temp_kp=32'h7FFF_FFFF;
end
if ((sum_temp_ki[31]==1 )& (multi_temp_kp[31]==1))
begin
if (sum_temp_kp[31]==0)
sum_temp_kp=32'h8000_0000;
end
Duty_PI[15:0]=sum_temp_kp[31:16];
end
else
begin
Duty_PI=16'd0;
y=16'd0;
contr_win=3'd0;
kp_c=16'd0;
ki_c=16'd0;
multi_temp_kp=32'd0;
multi_temp_ki=32'd0;
sum_temp_kp=32'd0;
sum_temp_ki=32'd0;
buffer=1'd0;
end
end
assign PI_out=Duty_PI;
endmodule

```

左移(shift_L.v)

```

module shift_L(clk, shift_bit, vin, vin_shift);
input clk;
input [15:0] vin;
input [2:0] shift_bit;
reg vin_sign=1'b0;
reg [15:0] vin_temp;
output reg [15:0] vin_shift=16'd0;
always@(posedge clk)
begin
if(vin[15]==1'b0)
begin
vin_temp=vin;
case(shift_bit)
3'd1:begin
if((vin_temp[14]==1)) vin_temp=16'h7FFF;
else vin_temp=vin_temp<<shift_bit;
end
3'd2:begin
if((vin_temp[14]==1)|(vin_temp[13]==1)) vin_temp=16'h7FFF;
else vin_temp=vin_temp<<shift_bit;
end

```

```

        3' d3:begin
            if((vin_temp[14]==1)||((vin_temp[13]==1)||((vin_temp[12]==1)))
vin_temp=16' h7FFF;
            else vin_temp=vin_temp<<shift_bit;
            end
        3' d4:begin
            if((vin_temp[14]==1)||((vin_temp[13]==1)||((vin_temp[12]==1)||((vin_temp[11]==
1))) vin_temp=16' h7FFF;
            else vin_temp=vin_temp<<shift_bit;
            end
        default: vin_temp=vin_temp;
        endcase
        vin_shift=vin_temp;
    end
else //相當 vin 為負值時
begin
    vin_temp=~vin+16' d1;
    case(shift_bit)
    3' d1:begin
        if((vin_temp[14]==1)) vin_temp=16' h7FFF;
        else vin_temp=vin_temp<<shift_bit;
        end
    3' d2:begin
        if((vin_temp[14]==1)||((vin_temp[13]==1)) vin_temp=16' h7FFF;
        else vin_temp=vin_temp<<shift_bit;
        end
    3' d3:begin
        if((vin_temp[14]==1)||((vin_temp[13]==1)||((vin_temp[12]==1)))
vin_temp=16' h7FFF;
        else vin_temp=vin_temp<<shift_bit;
        end
    3' d4:begin
        if((vin_temp[14]==1)||((vin_temp[13]==1)||((vin_temp[12]==1)||((vin_temp[11]==
1))) vin_temp=16' h7FFF;
        else vin_temp=vin_temp<<shift_bit;
        end
        default: vin_temp=vin_temp;
        endcase
        vin_shift=~vin_temp+16' d1;
    end
end
endmodule

```

右移(shift_L.v)

```

module shift_R(clk, shift_bit, vin, vin_shift);
input clk;
input [15:0] vin;
input [2:0] shift_bit;
reg vin_sign=1' b0;
output reg [15:0] vin_shift=16' d0;
always@(posedge clk)
begin

```



```

vin_sign=vin[15];
if(vin[15]==1'b1)
    begin
        vin_shift=vin>>shift_bit;
        vin_shift[15]=1'b1;
    end
else
    begin
        vin_shift=vin>>shift_bit;
        vin_shift[15]=1'b0;
    end
end
endmodule

```

零交越偵測(zcp.v)

```

module zcp(clk, vin_plus, vin, sin_counter);
input clk;
input [15:0] vin;
output reg [8:0] sin_counter=9'd1;
output reg vin_plus=1'b0;
reg check_zero=1'b0;
always @(posedge clk)
begin
    vin_plus=vin[15];
    sin_counter=sin_counter+9'd1;
    if(check_zero!=vin_plus)
    begin
        sin_counter=9'd4;
        //本來設1, 60Hz 若改 shift 一個角度, 可設 0~334,
        //400Hz 偏差 92us, 故補償 4 個點, 要改為 9'd4, 範圍可設 0~50
    end
    if(sin_counter==9'd50) //若不需 shift, 則這行要移除
        sin_counter=9'd1; //若不需 shift, 則這行要移除
        check_zero=vin_plus;
    end
endmodule

```

AD 程式(ad.v)

```

module ad(clk, in, out_1, out_2, convst, sfrm, start, test);
input clk, sfrm, in, start;
output reg convst=1, test=0;
output reg [15:0] out_1=16'd0, out_2=16'd0;
reg [6:0] counter_1=7'd1;
reg [3:0] counter_2=4'b1110, counter_3=4'd1;
reg [15:0] data=16'd0;
reg i=1'd0;
always @(negedge clk)
begin
    if(!start)
    begin
        if(counter_1==5'd0)

```

```

begin
    convst=1' b0;
    counter_1=counter_1+5' b0_0001;
end
else if(counter_1==5' d1)
begin
    convst=1' b1;
    counter_1=counter_1+5' b0_0001;
end
else if(counter_1==5' d19)
    counter_1=5' b0_0000;
else
    counter_1=counter_1+5' b0_0001;
if(sfrm==1' b0)
begin
    if(i==1' d0)
begin
    data[counter_2-4' b0001]=in;
    counter_2=counter_2-4' b0001;
    if(counter_2==4' d0)
begin
        i=1' d1;
        if(counter_3==4' d1)
begin
            out_1=(data<<2);
            counter_3=counter_3+4' d1;
            test=1;
        end
        else if(counter_3==4' d2)
begin
            out_2=(data<<2);
            counter_3=counter_3+4' d1;
            test=0;
        end
        else if(counter_3==4' d5)
//改 counter_3 會改送出資料的頻率, 原來 10 是 20K, 改成 5 變 40K
            counter_3=4' d1;
        else
            counter_3=counter_3+4' d1;
        end
    end
end
end
else
begin
    i=1' d0;
    counter_2=4' b1110;
end
end
else
begin
    counter_2=4' b1110;
    counter_3=4' d1;
    out_1=16' d0;

```

```

        out_2=16' d0;
        data=16' d0;
        i=1' d0;
        counter_1=5' d0;
        convst=1;
    end
end
endmodule

```

DA 程式(da.v)

```

module da(la, clk, out, in);
output reg out=0, la=0;
input clk;
reg [15:0] data_save=16' d0;
reg [4:0] counter=5' d15;
input [15:0] in;
always @(negedge clk)
begin
    if(counter==5' d15)
    begin
        data_save=in;
        out=data_save[counter];
        la=0;
    end
    else if(counter==5' d7)
    begin
        la=1;
        out=data_save[counter];
    end
    else if(counter==5' d0)
    begin
        out=data_save[counter];
        counter=5' d16;
    end
    else
        out=data_save[counter];
        counter=counter-1' d1;
    end
end
endmodule

```

除頻程式(clk_conversion.v)

```

module clk_conversion(clk_in, divider, clk_out);
input clk_in;
input [15:0] divider;
output reg clk_out=1' d0;
reg [15:0] counter=16' d0;
always@(posedge clk_in)
begin
    if(counter==divider)
    begin

```

```

        counter=16'd0;
        clk_out=~clk_out;
    end
else
    counter=counter+1;
end
endmodule

```

三角波兼比較器(triangle.v)

```

module
triangle(vcont_temp, PWM_1_out, PWM_2_out, clk, start, triangle_1, triangle_2, vcont);
output reg PWM_1_out=1'b0, PWM_2_out=1'b0;
output reg [8:0] triangle_1=9'd0, triangle_2=9'd500, vcont=9'd0;
input [15:0] vcont_temp;
reg i=1'b0;
input clk, start;
always @ (negedge clk)
begin
//將 vcont 往上移
if(!start)
begin
if(vcont_temp[15]) //vcont_temp[15]=1 的時候, vcont=0
vcont=9'd0;
else
begin
vcont[8:0]=vcont_temp[14:6];
if(vcont>9'd500)
vcont=9'd500; // (解析度)
end
if(i==1'b1)
begin
triangle_1=triangle_1-1'b1;
triangle_2=triangle_2+1'b1;
if(triangle_1==9'd0)
i=1'b0;
end
else
begin
triangle_1=triangle_1+1'b1;
triangle_2=triangle_2-1'b1;
if(triangle_1==9'd500)
i=1'b1;
end
if(vcont>triangle_1)
PWM_1_out=1'b0;
else
PWM_1_out=1'b1;
if(vcont<triangle_2)
PWM_2_out=1'b0;
else
PWM_2_out=1'b1;
end
end

```

```

else
    begin
        triangle_1=9'd0;
        triangle_2=9'd500;
        PWM_1_out=1'b0;
        PWM_2_out=1'b0;
        i=1'b0;
        vcont=9'd0;
    end
end
endmodule

```

開關(tosw.v)

```

module tosw(duty, vin_plus, an, bn, startsw);
input duty, vin_plus, startsw;
output reg an=1'b0, bn=1'b0;
always @(duty)
begin
    if (startsw==1'b0)
    begin
        if (vin_plus==1'b0) //輸入電壓正半週期
        begin
            if(duty==1'b1)
            begin
                an=1'b1;
                bn=1'b0;
            end
            else
            begin
                an=1'b0;
                bn=1'b0;
            end
        end
        else //輸入電壓在負半週期
        begin
            if(duty==1'b1)
            begin
                an=1'b0;
                bn=1'b1;
            end
            else
            begin
                an=1'b0;
                bn=1'b0;
            end
        end
    end
    else //startsw 未開時
    begin
        an=1'b0;
        bn=1'b0;
    end
end

```

```

end
endmodule

```

絕對值(absolute.v)

```

module absolute(clk, vin, vin_abs);
input clk;
input [15:0] vin;
output reg [15:0] vin_abs=16'd0;
always @(clk)
begin
    if(vin[15]==1'b1) vin_abs=~vin+16'd1;
    else vin_abs=vin;
end
endmodule

```

指撥開關調整 AD 值(ad_dip.v)

```

module ad_dip(clk, adin, adout, pin_1, pin_2, pin_3, ad_plus);
input [15:0] adin;
input clk, pin_1, pin_2, pin_3, ad_plus;
output reg [15:0] adout;
reg [2:0] x=3'd0;
always @(negedge clk)
begin
    if(ad_plus==1'b0)
    begin
        adout=adin+16'd788;
        //t=1'b1;
    end
    else
    begin
        x[2]=~pin_1; //利用開關 5~7 調整 offset 值, 7 是 MSB
        x[1]=~pin_2;
        x[0]=~pin_3;
        case(x)
            3'b000: adout=adin+16'd788;
            //補償 MAX121 的 offset, 輸入 0V 時, AD 值 FCEC=788,
            3'b001: adout=adin+16'd1200;
            3'b010: adout=adin+16'd1230;
            3'b011: adout=adin+16'd1260;
            3'b100: adout=adin+16'd1290;
            3'b101: adout=adin+16'd1320;
            3'b110: adout=adin+16'd1350;
            3'b111: adout=adin+16'd1380;
        endcase
    end
end
endmodule

```

指撥開關調整 KI 值(ki_dip.v)

```

module ki_dip(clk, ki, pin_1, pin_2, pin_3);
input clk, pin_1, pin_2, pin_3;

```

```

output reg [15:0] ki;
reg [2:0] x=3'd0;
always @(negedge clk)
begin
    x[2]=~pin_1;
    x[1]=~pin_2;
    x[0]=~pin_3;
    case(x)
        3'b000: ki=16'd10;
        3'b001: ki=16'd50;
        3'b010: ki=16'd100;
        3'b011: ki=16'd200;
        3'b100: ki=16'd300;
        3'b101: ki=16'd500;
        3'b110: ki=16'd700;
        3'b111: ki=16'd900;
    endcase
end
endmodule

```

指撥開關調整 KP 值(kp_dip.v)

```

module kp_dip(clk, kp, pin_1, pin_2, pin_3);
input clk, pin_1, pin_2, pin_3;
output reg [15:0] kp;
reg [2:0] x=3'd0;
always @(negedge clk)
begin
    x[2]=~pin_1;
    x[1]=~pin_2;
    x[0]=~pin_3;
    case(x)
        3'b000: kp=16'd50;
        3'b001: kp=16'd100;
        3'b010: kp=16'd150;
        3'b011: kp=16'd200;
        3'b100: kp=16'd300;
        3'b101: kp=16'd500;
        3'b110: kp=16'd800;
        3'b111: kp=16'd1000;
    endcase
end
endmodule

```

60Hz 三角函數查表(sin_table.v)

```

module sin_table(sin, cos, sin_counter);
output reg [15:0] sin=16'd0, cos=16'd0;
input [8:0] sin_counter;
always @ (sin_counter)
begin
    case(sin_counter)

```

```

9'd 1 :begin sin=16'd 0 ; cos=16'd 32767 ; end
9'd 2 :begin sin=16'd 308 ; cos=16'd 32766 ; end
9'd 3 :begin sin=16'd 617 ; cos=16'd 32761 ; end
9'd 4 :begin sin=16'd 925 ; cos=16'd 32754 ; end
9'd 5 :begin sin=16'd 1233 ; cos=16'd 32744 ; end
9'd 6 :begin sin=16'd 1541 ; cos=16'd 32731 ; end
9'd 7 :begin sin=16'd 1849 ; cos=16'd 32715 ; end
9'd 8 :begin sin=16'd 2157 ; cos=16'd 32696 ; end
9'd 9 :begin sin=16'd 2464 ; cos=16'd 32674 ; end
9'd 10 :begin sin=16'd 2772 ; cos=16'd 32650 ; end
.....(略)
9'd 504 :begin sin=16'd 32761 ; cos=16'd 64896 ; end
9'd 505 :begin sin=16'd 32753 ; cos=16'd 64588 ; end
9'd 506 :begin sin=16'd 32743 ; cos=16'd 64280 ; end
9'd 507 :begin sin=16'd 32730 ; cos=16'd 63972 ; end
9'd 508 :begin sin=16'd 32714 ; cos=16'd 63664 ; end
9'd 509 :begin sin=16'd 32694 ; cos=16'd 63356 ; end
9'd 510 :begin sin=16'd 32673 ; cos=16'd 63049 ; end
9'd 511 :begin sin=16'd 32648 ; cos=16'd 62741 ; end
default: begin sin=16'd 0 ; cos=16'd 32767 ; end
endcase
end
endmodule

```

400Hz 三角函数查表(Hz_sin_table.v)

```

module Hz_sin_table(sin, cos, sin_counter);
output reg [15:0] sin=16'd0, cos=16'd0;
input [8:0] sin_counter;
always @ (sin_counter)
begin
case(sin_counter)
9'd 1 :begin sin=16'd 0 ; cos=16'd 32767 ; end
9'd 2 :begin sin=16'd 2057 ; cos=16'd 32702 ; end
9'd 3 :begin sin=16'd 4107 ; cos=16'd 32509 ; end
9'd 4 :begin sin=16'd 6140 ; cos=16'd 32187 ; end
9'd 5 :begin sin=16'd 8149 ; cos=16'd 31738 ; end
9'd 6 :begin sin=16'd 10126 ; cos=16'd 31163 ; end
9'd 7 :begin sin=16'd 12062 ; cos=16'd 30466 ; end
9'd 8 :begin sin=16'd 13952 ; cos=16'd 29648 ; end
9'd 9 :begin sin=16'd 15786 ; cos=16'd 28714 ; end
9'd 10 :begin sin=16'd 17557 ; cos=16'd 27666 ; end
.....(略)
9'd 91 :begin sin=16'd 19260 ; cos=16'd 40288 ; end
9'd 92 :begin sin=16'd 17557 ; cos=16'd 39026 ; end
9'd 93 :begin sin=16'd 15786 ; cos=16'd 37869 ; end
9'd 94 :begin sin=16'd 13952 ; cos=16'd 36821 ; end
9'd 95 :begin sin=16'd 12062 ; cos=16'd 35887 ; end
9'd 96 :begin sin=16'd 10126 ; cos=16'd 35069 ; end
9'd 97 :begin sin=16'd 8149 ; cos=16'd 34372 ; end
9'd 98 :begin sin=16'd 6140 ; cos=16'd 33797 ; end
9'd 99 :begin sin=16'd 4107 ; cos=16'd 33348 ; end
9'd 100 :begin sin=16'd 2057 ; cos=16'd 33026 ; end

```



```
default: begin sin=16' d 0          ;   cos=16' d 32767    ;   end
endcase
end
endmodule
```

