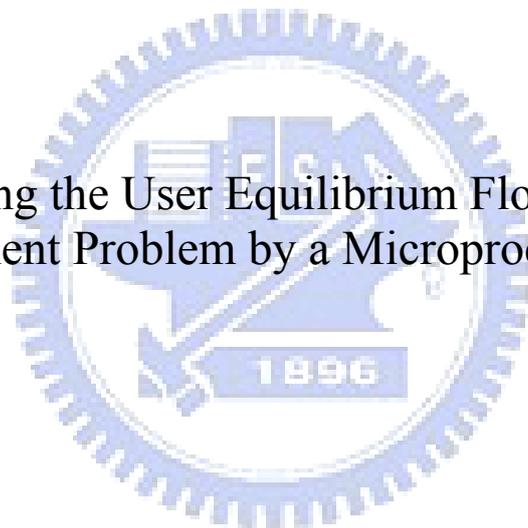


國立交通大學
運輸科技與管理學系碩士班

碩士論文

以微處理機系統決定交通路網之使用者均衡流量

Determining the User Equilibrium Flow of the Traffic
Assignment Problem by a Microprocessor System



研究生：謝智翔

指導教授：黃寬丞 博士

中華民國九十八年七月

以微處理機系統決定交通路網之使用者均衡流量

Determining the User Equilibrium Flow of the Traffic Assignment Problem by
a Microprocessor System

研究生：謝智翔

Student : Chih-Hsiang Hsieh

指導教授：黃寬丞

Advisor : Kuancheng Huang



Submitted to Department of Transportation Technology and Management
College of Management
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Transportation Technology and Management

July 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年七月

以微處理機系統決定交通路網之使用者均衡流量

學生：謝智翔

指導教授：黃寬丞 博士

國立交通大學

運輸科技與管理學系 碩士班

摘要

交通指派問題發展已久，相關研究也相當多，而「使用者均衡」是交通指派問題中一個基本的模式。先前有研究以電子電路的概念，利用「電流」來比擬「車流」，求解使用者均衡問題。理論上，以量測電流代替運算車流的方式來得到車流量的資料，速度上應該較快，但是，這僅僅是軟體上的模擬運算，實際硬體電路是否可行卻是未知，因實作時需要考量的因子更為繁雜。本研究透過小型微處理器控制整個電路的運作，並發展出一套適合實作使用的電路演算法，以實現微處理機系統之實際硬體製作。初步的實驗顯示，實作流量值與理論流量值之間差距不大，說明實作電路的可行。

關鍵字：微處理器、電路實作、實作誤差

Determining the User Equilibrium Flow of the Traffic Assignment Problem by a Microprocessor System

Student: Chih-Hsiang Hsieh

Advisor: Dr. Kuancheng Huang

Department of Transportation Technology and Management
National Chiao Tung University

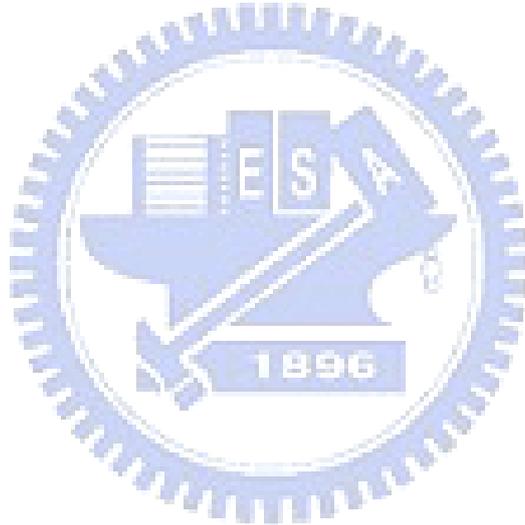
Abstract

The traffic assignment problem (TAP) has been researched for a long time, and the user equilibrium is one of the basic model of it. There are some researches using electrical current to simulate the traffic flow to solve the user equilibrium problem. By measuring electrical current substitute for computing traffic flow, it is faster to get the data of the flow theoretically. This is only a simulation on the software, and we don't know the physical circuit can be implemented or not, because there are more complicated factors should be considered. This study controls the whole circuit operation by a small microprocessor, and establishes a physical circuit algorithm. The result shows that the difference between physical flow and theoretical flow is small, and it can illustrate the feasibility of physical circuit.

Keywords: microprocessor, physical circuit, physical circuit error

誌 謝

不知不覺就要畢業了，當初決定要轉考運管所是下了很大的決心，畢竟是自己沒有接觸過的領域，只憑著一股「有興趣」的意念就跳進來了。首先感謝指導教授黃寬丞老師，在這兩年給了我做研究上的許多建議，讓我的論文可以如期完成。接下來再感謝口試委員許巧鶯老師、黃家耀老師在口試時所給予的許多中肯建議，讓我受益匪淺。在這裡，認識了許多好朋友，除了實驗室的同學之外，也有其他實驗室的(廢話)，感謝同實驗室景堯、兆哲、丞博、郁英，還有已畢業的老人展宗，以及各個實驗室的好夥伴們(族繁不及備載)，這是一段很快樂的時光，謝謝大家的幫忙以及照顧。還要特別感謝的是我的女朋友—榆萱，這些年來的日子謝謝你陪伴在我的身邊，在我難過或失意的時候安慰著我，在我開心的時候和我一起分享著喜悅，希望未來的日子我們也可以一直的走下去。最後要感謝的是我的父母親，感謝在我決定轉考的時候支持我，這些年來也不斷地給我鼓勵與包容，期許未來的我以後可以帶著你們的支持繼續走下去。



目錄

中文摘要.....	III
英文摘要.....	IV
誌謝.....	V
目錄.....	VI
圖目錄.....	VIII
表目錄.....	IX
第一章 簡介.....	1
1.1 交通路網指派問題簡介.....	1
1.2 使用者均衡.....	2
1.3 使用者均衡與電路之相關性.....	2
1.4 路段成本函數與歐姆定律.....	3
1.5 實際電路製作之重要議題.....	5
第二章 文獻回顧.....	7
2.1 交通指派演算法回顧.....	7
2.2 電路模擬演算法回顧.....	8
第三章 電路模擬演算法與實際電路演算法.....	10
3.1 子系統之電路模擬演算法.....	11
3.2 子系統之之實際電路演算法.....	12
3.3 系統電路模擬演算法.....	13
3.4 系統之實際電路演算法.....	15
第四章 實作的誤差與影響.....	16
4.1 量化誤差的定義.....	16
4.2 電阻值量化的影響.....	17
4.3 元件的誤差.....	18
4.4 量化誤差與元件誤差的綜合影響.....	19
4.5 元件的物理限制.....	21
第五章 微處理機系統之電子電路實作.....	23
5.1 系統架構圖與元件介紹.....	23
5.2 子路網之實際電路製作.....	25

5.2.1 單起訖點對之電路製作.....	27
5.2.2 兩起點對一訖點之實際電路製作.....	29
5.3 系統電路實際製作.....	31
5.4 系統容量測試.....	33
5.5 運算時間之估算.....	37
第六章 結論與建議.....	38
參考文獻.....	39
附錄一 微處理器(P89V51RD2)腳位圖及說明.....	40
附錄二 數位電阻 AD8400 接腳圖及說明.....	42
附錄三 數位/類比轉換器(ADC0804)腳位圖及說明.....	43
附錄四 移位暫存器(74164)腳位圖及真值表.....	44



圖目錄

圖 1.1	雙路段路網示意圖	2
圖 1.2	非線性成本函數圖	4
圖 1.3	歐姆定律關係圖	4
圖 1.4	兩電阻並聯電路圖	5
圖 3.1	全路網系統示意圖	10
圖 3.2	子路網系統示意圖	10
圖 3.3	雙曲線逼近成本函數示意圖	12
圖 4.1	電腦模擬測試路網圖	17
圖 4.2	電腦模擬測試電路圖	18
圖 5.1	超級終端機顯示介面	25
圖 5.2	電子電路架構圖	26
圖 5.3	單起訖點對之電路實作圖	27
圖 5.4	兩起點對一訖點之測試路網	29
圖 5.5	兩起點對一訖點之電路實作圖	30
圖 5.6	兩起點對兩訖點之測試路網(各 OD 間流量=1).....	32
圖 5.7	兩起點對兩訖點之測試路網(各 OD 間流量=15).....	34
圖 5.8	起點 O_1 至訖點 D_1 之通過路段.....	36
圖 5.9	KEIL C 軟體偵錯介面圖	37
圖 A.1	微處理器 P89V51RD2 頂視圖.....	40
圖 B.1	數位電阻 AD8400 頂視圖	42
圖 C.1	數位/類比轉換器 ADC0804 頂視圖	43
圖 D.1	移位暫存器 74164 頂視圖	44

表目錄

表 4.1	數位電阻階數與實際阻值間的關係(以最大值為 $10K\Omega$ 為例).....	16
表 4.2	量化階數不同對於目標值之影響.....	18
表 4.3	三十個數位可變電阻之設定電阻值與實際表示阻值.....	20
表 4.4	量化誤差與元件誤差合併考慮之影響.....	21
表 5.1	單起訖點例題隨機更換電阻之 15 組解.....	28
表 5.2	兩起點對一訖點之路段成本參數.....	31
表 5.3	兩起點對一訖點之理論與實際電路之各路段流量.....	31
表 5.4	兩起點對兩訖點之路段成本參數(各 OD 間流量=1).....	33
表 5.5	兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=1).....	33
表 5.6	測試路段之路段成本參數(各 OD 間流量=15).....	34
表 5.7	兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=15).....	34
表 5.8	兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=20).....	35
表 5.9	兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=30).....	35
表 5.10	兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=40).....	36
表 A.1	P89V51RD2 接腳說明.....	40
表 B.1	AD8400 腳位圖及說明.....	42
表 C.1	ADC0804 接腳說明.....	43
表 D.1	移位暫存器 741647 真值表.....	44

第一章 簡介

1.1 交通路網指派問題簡介

近年來，由於都市快速的發展，「運輸規劃」這門學問已經成為一種顯學，因為都市網路的日益複雜，使得做好規劃的工作充滿了挑戰性。在運輸規劃的過程中，最主要有以下四個步驟：

1. 旅次產生(trip generation)，是指將整個路網分成許多個區塊，並且由這些區塊作為產生旅次的起點或是作為旅次的訖點。
2. 旅次分配(trip distribution)，是指各區塊間形成一對對起訖點(origin-destination pair, OD pair)間之運量。例如，由區塊 i 出發往區塊 j 的流量可以 Q_{ij} 表示。
3. 運具選擇(mode choice)，考慮每一對起訖點在運輸活動可以選擇使用的運輸模式，例如使用運具 m ，其對應流量可以 Q_{ijm} 表示。
4. 交通指派(traffic assignment)，考量各起訖點與各運具之路徑(path)來分配旅次的計劃過程。其核心問題在於探討旅行者在某一路網中，會如何從起點選擇適合的路徑到達目的地。若以 k 來代表到達訖點的路徑之一，其對應流量可以 Q_{ijmk} 表示。

其中，交通指派問題是我們所關注的焦點。而在交通路網的流量模式中，使用者均衡(User Equilibrium，簡稱為 UE)與系統最佳化(System Optimization，簡稱為 SO)是兩個最基本的靜態交通路網模式，前者的任一起訖點間之旅行成本間相同，而後者的全路網成本總和將會最低，不過，本研究所最關注的基本概念為「使用者均衡」。在使用者均衡的狀況下，路網中的各個使用者，會選擇對於自己而言成本最低的路徑行走，也就是說，在均衡狀態下，對於任何起訖點來說，路徑的旅行成本將會相同。使用者均衡的概念，對於整個路網系統來說是非常重要的，旅行者希望能夠以最短的時間抵達其目的地，近來更由於科技的發展，智慧型運輸系統(Intelligent Transport Systems，簡稱為 ITS)的發展日趨成熟，車輛導航系統的路徑規劃甚至也開始考慮道路流量的大小，讓使用者可以最快速的方式抵達目的地。除此之外，政府部分也希望可以對交通壅塞的路段進行拓寬或是其他改善，或是對於道路經施工或是遊行等等原因封閉，進行車流量的進一步預測與執行相對應的措施，這些都與使用者均衡的概念息息相關。

1.2 使用者均衡

在一個交通路網中，使用者會傾向於選擇對自己而言，成本最低的路徑行走。舉圖 1.1 的簡單的單一起訖點與雙路段路網為例，路段 1 與路段 2（即 Link 1 與 Link 2）都擁有自己的路段成本函數，兩成本函數不相同。且路段流量高時，路段的行駛時間會越長（因較容易塞車），也就是成本越高；路段流量低時，路段的行駛時間越短，也就是成本越低。使用者在本路網中，將會選擇行駛時間較短的路段。舉例來說，在無車流量時，行駛於路段 1 耗時 10 分鐘，行駛於路段 2 耗時 5 分鐘，則當有一駕駛要進路本路網時，自然會選擇路段 2 行走，路段 2 的車流量將越來越高，行駛的時間也會越來越長。當在路段 2 行駛的時間超過了 10 分鐘時，選擇路段 1 行駛似乎更好的選擇。久而久之，兩路段便會達到一均衡（equilibrium）狀態，也就是不論行駛於路段 1 或路段 2 的耗費時間皆相同，這就是所謂的使用者均衡現象。

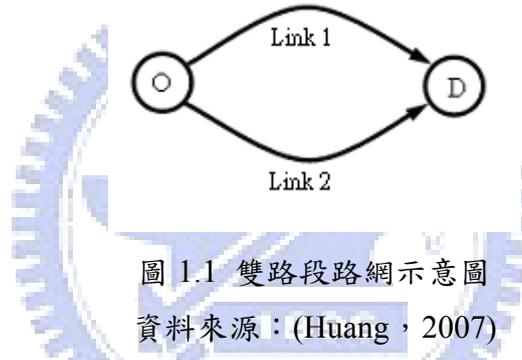


圖 1.1 雙路段路網示意圖
資料來源：(Huang, 2007)

在使用者均衡的狀態下，對於路網中的任何起訖點對，其所使用任意一條路徑的旅行時間皆相同，且旅行時間都要比該起訖對任一未使用的路徑都還要低或是相同，此即為使用者均衡的定義。

1.3 使用者均衡與電路之相關性

我們發現，交通指派問題的使用者均衡與一般電子電路間，存在著許多相同點。交通路網與電子電路之間不僅是網路結構相似，還有其他許多的相關性，分別簡介如下：

1. 在如前述的並聯電路中，電流分配是根據分支電路之電阻值而定，且並聯處兩端電壓會相等，這類似於交通路網中，車流分配是根據路段成本函數而決定，且路段的旅行時間相等，亦即所謂使用者均衡的原理。
2. 在串聯的電路中，電路的總電壓降為電阻各端點之分壓降值加總，類似於交

通路網中，路段的總旅行時間為各路段旅行時間的加總。

3. 在串並聯混合電路中，電路的各節點將會滿足克希何夫電流定律 (Kirchhoff's current law)，也就是節點之流入電流會等於流出電流，類似於交通路網中的流量守恆。且電路中起訖點不同路徑之總電壓降，會等於起訖點之電壓差，類似於交通路網中在使用者均衡時，各路徑之旅行時間相同。

1.4 路段成本函數與歐姆定律

路段成本函數的概念，代表著使用者行駛於某段道路上，所需耗費的成本。「成本」二字的涵義有很多，但其中最廣泛應用的就是時間的成本，旅行時間越長，代表著成本越高，在完全無車流時，路段會有一個自由車流的成本；而當路段的車流量越來越大時，將會造成壅塞的現象，因此旅行時間也會越來越長。在這裡，我們常使用美國公路局(U.S. Bureau of Public Roads，簡稱為 BPR)所發展出的路段成本函數(BPR function)，函數如式(1-1)：

$$t_a(x_a) = T_a \times \left(1 + 0.15 \times \left(\frac{x_a}{C_a}\right)^4\right) \quad (1-1)$$

x_a ：路段 a 上的流量

T_a ：路段 a 之固定成本

C_a ：路段 a 之道路容量

$t_a(x_a)$ ：路段 a 的時間成本函數

按式(1-1)，我們發現路段的成本函數並非線性的關係，而是以四次方的函式做遞增的動作，因此，當流量 x_a 超過容量 C_a 之後，成本 t_a 會非常急劇的上昇。圖 1.2 即為典型的 BPR 函數圖形，此為 $T_a=1$ 、容量 $C_a=50$ 的成本函數圖形，我們可以發現，在流量小於 50 之前，因流量尚未超過路段限制容量，故成本上升幅度較緩；當流量超過 50 以後，已超越了本身路段之容量限制，故發現成本非線性地急速上升。

而在電子電路的領域中，我們所關注的最重要法則為歐姆定律(Ohm's law)，此指在同一個導體中，經過導體兩端的電流與兩端的電壓成正比，與導體的電阻成反比，公式如(1-2)，其中 I 為電流， R 為電阻， V 為電壓：

$$I = \frac{V}{R} \quad (1-2)$$

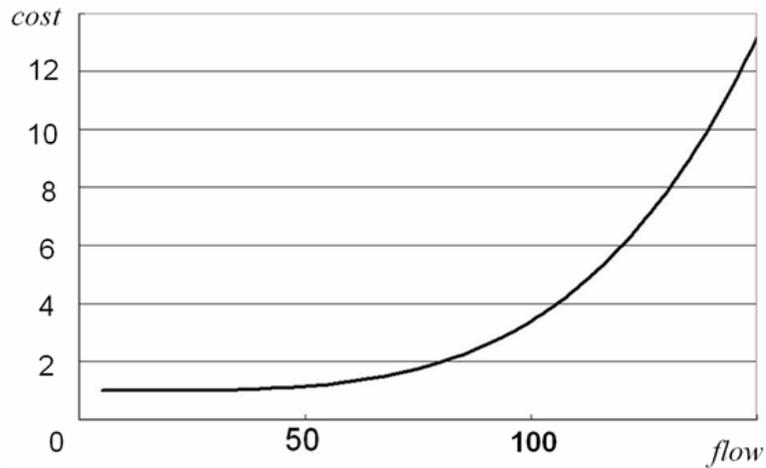


圖 1.2 非線性成本函數圖

資料來源：本研究整理

由(1-2)式可知，歐姆定律是一線性的關係。在電阻相同的情況下，電壓增大，電流就會增大；電壓減小，電流也跟著減小。現在假定有一個 25Ω 的電阻，其兩端有電壓差，故電阻將會有電流通過，而其電壓-電流關係如圖 1.3，橫軸單位為伏特(V)，縱軸單位為安培(A)，該圖之電壓與電流之間關係為線性。

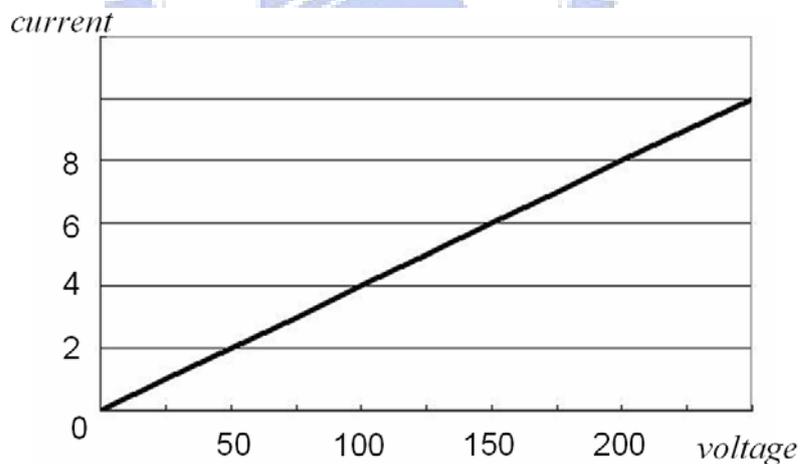


圖 1.3 歐姆定律關係圖

資料來源：本研究整理

現在應用歐姆定律至較為一般性的電路上，以圖 1.4 之兩電阻電路為例，因兩電阻為並聯，所以兩電阻兩端之電壓差為相同， I_1 為流經 R_1 之電流， I_2 為流經 R_2 之電流，所以歐姆定律也可以寫成如(1-3)的公式：

$$V = I_1 R_1 = I_2 R_2 \quad (1-3)$$

路段成本函數是非線性的函數，且函數不會通過原點；歐姆定律卻是一個線性的函數，且函數會經過原點，因此成本函數與歐姆定律存在著差異性。儘管在 1.3 節中，我們發現了許多交通指派問題與實際電路之間的關連性，但是兩者間網路的建構還是存在有許多的相異點。

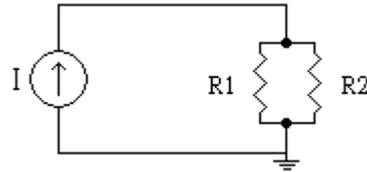


圖 1.4 兩電阻並聯電路圖

資料來源：本研究整理

1.5 實際電路製作之重要議題

過去已有學者發展出一套遞迴的模擬演算法，將線性的歐姆定律應用到非線性的成本函數之上，並運用其作電路流量均衡等的計算，但是，這僅是對其中的電路問題進行軟體上的模擬而已，實際上的硬體系統是否可行卻是個未知數。假如電路能夠真正的實作出來，將可進一步驗證電路模擬演算法之可行性，我們可以藉由實際的量測電路，立刻得出諸如電壓、電流等各種資料，來推算出實際的路段成本、流量等等資料，可以說是相當的方便快捷，故這就是我們實作電路的目的。

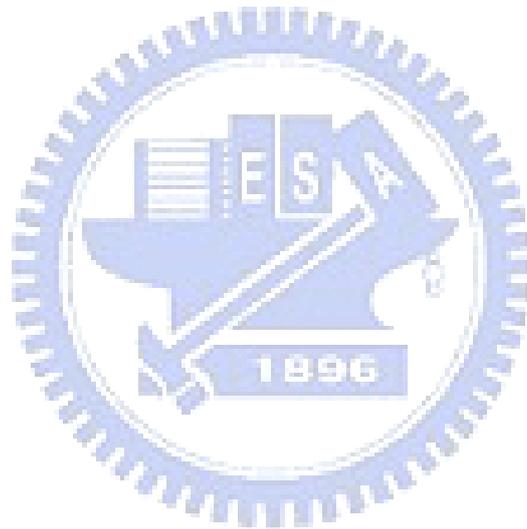
理論上，以「量測」代替「運算」的方式來得到車流量的資料，速度上應該會快上許多，因為運算需要時間，而量測的資料幾乎是可以立即得到，且利用直接量測的方式取得資料，不會普通演算法一樣，隨著網路規模的增大，使運算時間變的更長。

經由模擬的數值，都是較為理想化的結果。不過，在實作電路上面，需要考慮的因子非常多，不能僅透過先前的演算法來完成。在實作電路上，可能會遇到以下的幾項問題：

1. 量化誤差：元件在數位值與類比值之間的量化轉換造成的誤差。
2. 元件誤差：各元件(如電阻)之間設定值的誤差。
3. 元件本身限制：元件或硬體系統本身工作條件上先天的物理限制。

本研究中，我們將利用一核心的小型微處理器控制整個電路中電阻的運作，再將各個路段間的電壓值，透過類比/數位轉換器來轉換，並傳送回微處理器，微處理器根據傳回值做運算，運算完畢後回傳電阻值以完成迴圈，其中存在著許多上述的問題有待克服，主要的誤差(量化誤差、元件誤差等)會是在我們所使用的數位可變電阻身

上，且它還會有本身最大電阻值的限制；而類比/數位轉換器也會產生部分的量化誤差，這些問題會對先前的演算法造成一定程度的影響，本研究需要克服這些問題，才能真正地將理論電路實際製作出來。



第二章 文獻回顧

本章中，我們要先就傳統的交通指派演算法，以及學者所提出的電路模擬演算法作一般性的介紹。而在後面的章節，我們將會以電路模擬演算法為基礎做延伸及後續的研究工作。

2.1 交通指派演算法回顧

在運輸指派問題中，最重要的概念就是使用者均衡，根據 Sheffi (1985)的數學規劃模式，可表示如下。其他還有許多關於運輸指派問題研究的文獻，可參考 Patriksson (1994) 及 Bell & Iida (1997) 的研究。

$$\min z(x) = \sum_a \int_0^{x_a} t_a(\omega) d\omega \quad (2-1)$$

$$\sum_k f_k^{rs} = q_{rs} \quad \forall r, s \quad (2-2)$$

$$x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs} \quad \forall a \quad (2-3)$$

$$f_k^{rs} \geq 0 \quad \forall k, r, s \quad (2-4)$$

其中：

x_a ：路段 a 上的流量

q_{rs} ：起點 r 至訖點 s 的交通流量

f_k^{rs} ：在路徑 k 上，起點 r 至訖點 s 的交通流量

$t_a(\omega)$ ：路段 a 在流量為 ω 時耗費之旅行成本

$\delta_{a,k}^{rs}$ ：二元參數，起訖點 r, s 的路徑 k 包含路段 a 時為 1，否則為 0。

目標式(2-1)為各路段的成本函式在有流量下時之積分式總和，限制式(2-2)為流量守恆限制式，限制式(2-3)代表各個路段的流量為所有起訖點路徑有經過該路段之總和，限制式(2-4)表示流量恆大於零。

而在求解交通指派問題的演算法中，最常被使用的就是 Frank-Wolfe 演算法，參考 Sheffi (1985)的著作，簡介如下：

步驟 0：初始化。初始流量為 0 時，執行全有全無指派(All-or-nothing)，求得初始流量 $\{x_a^1\}$ ，遞迴次數 $n=1$ 。

步驟 1：更新。設定 $t_a^n = t_a(x_a^n), \forall a$ ， x_a^n 為路段 a 在遞迴次數 n 時之流量。

步驟 2：方向尋找。以 t_a^n 為依據，執行全有全無指派(All-or-nothing)，這可

以產生一個輔助流量的集合 $\{y_a^n\}$ 。

步驟 3：線性搜尋。對目標式做線性搜尋

$$\min \sum_a \int_0^{x_a^n + \alpha(y_a^n - x_a^n)} t_a(\omega) d\omega, 0 \leq \alpha \leq 1, \text{ 以求得最佳 } \alpha_n$$

步驟 4：移動。設定路段 a 下一次的流量 $x_a^{n+1} = x_a^n + \alpha_n(y_a^n - x_a^n), \forall a$

步驟 5：收斂測試。如已達成收斂條件，即停止，目前的 $\{x_a^{n+1}\}$ 流量即為均衡的路段流量集合。否則，設定遞迴 $n = n + 1$ ，並返回步驟 1 演算。

由於在 Frank-Wolfe 演算法中，大部分的時間都在解決最短路徑問題(Patriksson, 1994)，所以當路網規模越來越大時，演算法的效率便會變差。

2.2 電路模擬演算法回顧

根據文獻，已有多位學者利用電子電路觀念求解交通指派問題，Sasaki & Inouye (1974)利用電壓類比旅行成本，以及電流類比車流量的觀念，再運用歐姆定律，並增加了通過節點的時間納入考慮，以求解交通指派問題，但仍僅限於求解線性成本函數之問題。

Wang and Zhang (2005)根據克希荷夫定律(Kirchhoff's Law)建立一套非線性的系統，來計算交通指派問題中之使用者均衡，而該問題是由數學軟體 MATLAB 來求解，並與漸增法(Incremental assignment)比較結果，由於該研究是依賴數學軟體來解題，並沒有強調其演算法對於解交通指派問題的效用，因此能否解決大型的交通路網問題是個未知數。

Cho et al. (2006)利用了二極體這種非線性的電子元件的特性，以分段逼近的方式，模擬一般成本函數與車流量之間的非線性關係。換句話說，將路段中非線性的成本函數，以電阻及二極體來分段成幾條線性的成本函數，並利用此電路直接計算出電流與電壓，以求解較為一般性的交通指派問題。儘管如此，該方法仍然存在著諸多現實的限制。二極體有一個約 0.6V~0.7V 的導通電壓，而當網路變得較為複雜之後，許多二極體元件串聯的結果，電壓可能需要增至數十甚或數百伏特才能將二極體導通，這在實際應用上會有些困難，因為過大的電壓可能造成實際系統過大的負荷。

Huang et al. (2007) 參考前面學者的研究，選擇了以純電阻電路作為基本的模擬電路，再透過遞迴修正電阻值的方式，來逼近非線性的成本函數。與Cho et al.(2006)的方法做比較，此方法不以二極體做模擬，僅利用純電阻模擬，這讓模擬電路的實作變得較為可能，因為整個電路變的比較單純，前者直接利用非線性之電子元件近似非線性成本函數，且電路兩端電壓差過大，讓實作變得較為困難。鄭兆哲(2009)針對了

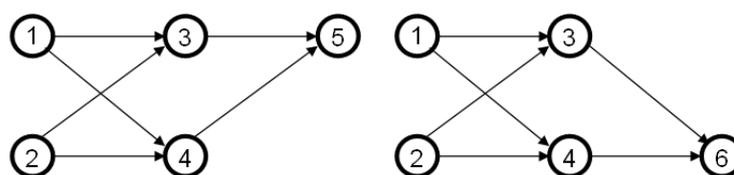
先前演算法做了改良，改善了先前演算法的收斂問題，且其效率更好。先前演算法尚未處理路段雙向流量的問題，該演算法將其加以改良以實現路段雙向流量問題的求解。

本研究主要是根據鄭兆哲(2009)的文章做延伸的實作，實作上將會遇到許多實際的問題，在1.5節中已有提及，並根據該研究演算法的架構，發展出了一套實際電路製作時所使用的演算法，該法是針對實作時所產生的各項誤差以及限制所應運而生的，故接下來的章節將會一一介紹。



第三章 電路模擬演算法與實際電路演算法

第二章所介紹的電路模擬演算法中，我們將會利用鄭兆哲(2009)的研究繼續做延伸的工作。按照先前文獻所言，電路模擬時需注意不可在同一電路中擁有多起點與多訖點，因為電子本身不認識路，而所流動之方向僅是由電壓降決定，故當同一電路同時存在多起點與多訖點，則電流將會無法分配到正確的需求點，所以模擬時必須先將全路網系統拆解成電流分配不產生衝突的子路網系統(Subsystem)，這些子網路系統為多起點對一訖點之網路。以圖 3.1 為例，我們須將此兩起點對兩訖點之路網，拆解成如圖 3.2 的兩個子路網，而這兩個子路網都是兩起點對一訖點之路網。本研究採用單起點對多訖點的子路網系統作為子問題的基礎，因為這種路網在實際電路的製作上較為簡單。



據此，我們在 3.1 節將會簡介子系統之電路模擬演算法(Circuit Simulation Algorithm for Subsystem，簡稱 CSA-Sub)，並在 3.2 節推導出電路子系統實作時，所使用的子系統之實際電路演算法(Physical Circuit Algorithm for Subsystem，簡稱 PCA-Sub)，其與 Huang et al. (2007)的研究相比較，增加了許多製作實際電路時需要考慮的部份，這對於電路的實作工作將會有很大的幫助，3.3 節會介紹全系統之電路模擬演算法(Circuit Simulation Algorithm for system，簡稱 CSA-Sys)，該演算法的結構

與 Frank-Wolfe 演算法類似，但電路模擬演算法是將電子視為旅行者，讓電子去找尋自己的路徑到訖點，故沒有 Frank-Wolfe 演算法中”找尋最短路徑”的步驟。3.4 節介紹了實作時使用的全系統實作電路演算法(Physical Circuit Algorithm for system, 簡稱 PCA-Sys)，這是根據 3.3 小節之演算法而來，與 3.2 節相同的是，同樣增加了實作時的考量因素在內。

3.1 子系統之電路模擬演算法(CSA-Sub)

使用前述Huang et al. (2007)演算法之概念，將該演算法分別說明如後。現在沿用前章的符號定義，並將 i_a 與 v_a 分別設為路段 a 的電流與電壓值， R_a 為路段 a 之電阻值，成本函數為 $t_a(x_a)$ ，路段成本為 c_a ， n 為遞迴次數，則子系統之實際電路演算法，其運算步驟如下(鄭兆哲，2009)：

步驟 0：初始化。將子路網之各起點模擬成直流電流源 I_{rs} ，其電流值設為起點至訖點之流量，即 $q_{rs} = I_{rs}$ ；另外，將路網訖點模擬成接地點。每個路段都個別模擬成電阻，以 R_a^n 表示第 n 次迴圈的電阻值。在初始時，也就是 $n=0$ 時， R_a^0 為 1。

步驟 1：啟動電路，計算出各路段在電阻值 R_a^n 時的電流值 i_a^n 。

步驟 2：逼近，根據已測得的 i_a^n 與 v_a^n 逼近到成本函數上，藉以得到流量 x_a^n 與旅行時間 c_a^n

步驟 3：利用歐姆定律，計算下一次各路段之電阻值，亦即 $R_a^{n+1} = c_a^n / x_a^n$

步驟 4：將流量代回目標式(1)中，若前後兩次目標值的變化量在要求範圍內，即為收斂，並停止計算；若不是，則回到步驟 1 繼續計算。

本法主要是透過遞迴修正電阻值的方法，求得電路流量的解，而逼近的過程主要是透過步驟 2 來完成。逼近的方式有許多種，根據研究，共有三種不同的逼近方法，但以雙曲線來逼近是最穩定的，可以避免使用其他方式逼近時，需求量大而不收斂的情況，故本研究將以雙曲線逼近的方式做後續的研究工作。該法大略敘述如後：將電流與電壓的乘積($i_a^n \times v_a^n = P$)定為一雙曲線，並用此求解其與成本函數的交點，以求得路段流量與旅行成本。

圖 3.1 為雙曲線逼近成本函數之示意圖，一開始成本函數上之解為 A 點，可以據此計算出電阻 R^n ，電路啟動後電流通過，可量測出電路的均衡解 B 點(i_a^n 與 v_a^n)，將電流與電壓做乘積，可得到一條雙曲線，再利用此雙曲線將答案逼近到成本函數上面的新解 C 點(x_a^n 與 c_a^n)，接著算出 C 點之電阻 R^{n+1} ，如此不斷地反覆求解。

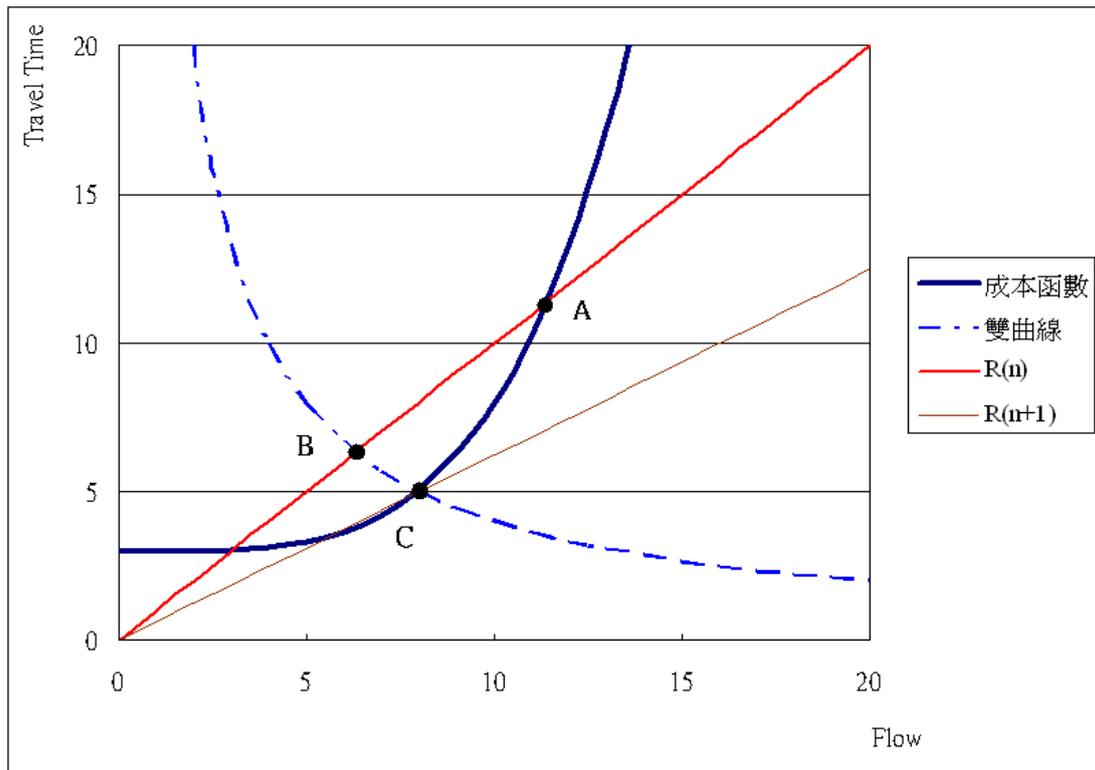


圖 3.3 雙曲線逼近成本函數示意圖

資料來源：(鄭兆哲，2009)

3.2 子系統之實際電路演算法(PCA-Sub)

根據前述演算法，我們發展出一套子系統之實際電路演算法。先前的演算法在演算時，沒有考慮到電阻值「量化誤差」的問題，因為在做電腦模擬時，數值都相當的精確，但實際電路製作時，需要考慮量化的誤差，沒有辦法設定成理論的電阻值一定等於實際的電阻值，這是因為元件本身的限制使然，無法做到百分之百的精確是必然的。

另外，除了量化的影響之外，我們可能會碰到遞迴過程中，運算時的電阻值 R_a^n 超過實體電阻最大值 R_{\max} 的限制，現在令演算法每次迴圈計算時其最大電阻值為 R_0^n ，即 $R_0^n = \max(R_a^n, \forall a)$ ，我們必須將所有電阻值作等比例的轉換，亦即同乘以 R_{\max} / R_0^n ，讓所有電阻值都不超過 R_{\max} 。基於以上各點，我們所發展出的實作電路演算法步驟如下：

步驟 0：將子路網之各起點 r 模擬成直流電流源 I_{rs} ，電流值設為該起點至訖點之

流量，各起點電流值之比會等於實際各路段流量 q_{rs} 之比(亦即 $\frac{q_{rs}}{I_{rs}} = k$ ，

且對每一個子路網，皆用同一個比值 k)；另外，將路網訖點模擬成接地點。每個路段都個別模擬成電阻，以 R_a^n 表示第 n 次迴圈的路段 a 電阻值。在初始時，也就是 $n=0$ 時， R_a^0 為 $1K\Omega$ ，最大阻值 $R_O^0 = 1K\Omega$ 。

步驟 1：電阻值轉換，設各路段所得之新電阻值為 S_a^n (即 $S_a^n = R_a^n \times \left(\frac{R_{\max}}{R_O^n}\right)$)。

步驟 2：將 S_a^n 量化，得新量化後之電阻值 Q_a^n 。

步驟 3：啟動電流源電路，取得此時各路段實際電流量 i_a^n 與電壓值 v_a^n 。

步驟 4：根據路段電流量 i_a^n ，計算實際交通路網之車流量 f_a^n ，即

$$f_a^n = i_a^n \times k, \forall r。接下來利用電壓值 v_a^n 計算出實際之旅行成本，即 $C_a^n = v_a^n \times k \times \frac{R_O}{R_{\max}}$ ，計算出 $f_a^n \times C_a^n = P$ 之雙曲線。$$

步驟 5：求取雙曲線與成本函數之間的交點，交點之流量為 x_a^n ，成本為 c_a^n 。利用歐姆定律，計算下一次各路段 a 之電阻值，也就是 $R_a^{n+1} = c_a^n / x_a^n$ ，並求取下一次迴圈時之 R_O ，即 $R_O^{n+1} = \max(R_a^{n+1}, \forall a)$

步驟 6：將流量代回目標式中，若前後兩次目標值的變化量在要求範圍內，即為收斂，並停止計算；若不是，則回到步驟 1 繼續計算。

在步驟 0 中，為了簡化初始值的計算，一般先將所有之 R_a^0 設為全部相同，此處之電阻初始值皆設為 $1K\Omega$ 。

步驟 1 中，假如所使用的元件最大值為 $10K\Omega$ ，則我們就會把最大的電阻值設為 $10K\Omega$ 。在遞迴的過程中，阻值可能會大於 $10K\Omega$ ，但是我們的數位可變電阻最大值僅有 $10K\Omega$ ，無法顯示大於 $10K\Omega$ 的部份，所以必須以此做等比例的切換。這樣做對於各路段的電流值不會有影響，因為只要各電阻間的阻值比是固定的，其所通過的電流比例也一定會是固定的。

步驟 4 中，必須將電路的實際電流量，以適當比例轉換成子路網的車流量。除此之外，我們也要利用所取得之電壓值，轉換成在該路段上的旅行時間成本，並使用這些流量與成本的資料，與成本函數做逼近。步驟 4 即是流量轉換以及成本轉換的動作。

3.3 系統電路模擬演算法(CSA-Sys)

上兩節所提出之演算法，可以量測計算出單一起訖對的均衡流量，但一般交通路網往往都是多重起訖點對的。且交通路網的特性與電路特性並非完全相同，比如說，交通路網中可以有雙向的車流，但在路網中，電壓差決定的電流的方向，所以，某些狀況下不能直接利用電路模擬路網。在多重起訖點之路網中，使用者可以旅次分配辨認

自己的起訖點，但是電流本身是無法分別起訖點的。所以，前述方法在改良以後，學者提出修正後的全系統電路模擬演算法，並假設路網有 Π 個旅次訖點，因此能夠拆解成 Π 組多起點對單訖點之子路網系統，演算法如下(鄭兆哲，2009)：

步驟 0：初始化，將路網拆解成子路網系統 π ，對所有子路網系統 π 將 O-D 之起點模擬成接地點，所有訖點模擬成直流電流源，電流值設為該訖點的需求流量(q_{rs})，末端直接接地。每個路段都個別模擬成電阻，並將初始值電阻值 R_a^0 設為 1 後便啟動電路。最後將所子路網系統得到的流量合併成為解集合 $x_a^0, \forall a$ (亦為 \mathbf{x}^0)，當作初始解。

步驟 1：轉換道路成本，我們將每個子路網的中路段的成本函數，設定為 $u_a(\omega) = \max(t_a(0), t_a(x_a^n) + t_a'(x_a^n)(\omega - x_{a,\pi}^n))$ ，其中 $x_{a,\pi}^n$ 為子路網系統 π 經過路段 a 之流量。

步驟 2：找尋方向，根據新的成本函數 $u_a(\omega)$ ， $\forall \pi \in \Pi$ 執行 CSA-Sub，以合併所有 CSA-Sub 的解當作搜尋方向 $y_a^n, \forall a$ 。

步驟 3：線性搜尋，利用 $\min_{0 \leq \alpha \leq 1} \sum_a \int_0^{x_a^n + \alpha(y_a^n - x_a^n)} t_a(\omega) d\omega$ 找出最適步幅 α^n 。

步驟 4：找尋新解，設定新解為 $x_a^{n+1} = x_a^n + \alpha^n (y_a^n - x_a^n), \forall a$ 。

步驟 5：測試收斂，若滿足所設定的收斂條件，就視同收斂並停止計算；若不是，則回到步驟 1 繼續計算。

從上述方法中可發現，FWA 和 CSA-sys 主要的差別在於步驟 0、步驟 1 與步驟 2。在步驟 0 中，不採取前述 Frank-Wolfe 演算法的全有全無指派，是因為在全有全無指派的時候，將會有指派流量為 0 狀況，使得電路上之電流為需為 0，這會造成電阻值的無法計算(按照歐姆定律，電流等於 0 會使電阻變得無限大)。CSA-sys 法雖然都將起始電阻設定為 1，但通常可以得到一個較好的起始解，對未來的收斂也不會有影響。

在步驟 1 與步驟 2 中，Frank-Wolfe 演算法主要是利用前次流量所得的成本做為一常數，並藉此進行全有全無指派；而 CSA-sys 法將成本函數轉換成一階近似導函數，根據新的成本函數和前次流量所得的成本利用 CSA-sub 進行指派，這種利用非線性成本函數，取代線性成本函數，以尋找最短路徑的解法，類似於 Regularized Frank-Wolfe Algorithm (Migdalas, 1984)，主要可避免 Frank-Wolfe 演算法找尋可行解方向時會找到邊界解(boundary solutions)的缺點。

3.4 系統實際電路演算法(PCA-Sys)

根據 3.3 節之演算法，下面將介紹全系統之實際電路演算法。在 3.2 節中已說明，實作使用的演算法要考慮的因子較多，故我們將對 3.3 節多重起訖點之電路模擬演算法做修正，使其可以在實際電路中完整地呈現。系統之實作電路演算法介紹如下：

步驟 0：初始化，將路網拆解成子路網系統 π ，對所有子路網系統 π 將 O-D 之起點模擬成接地點，所有訖點模擬成直流電流源，電流值設為該訖點的需求流量 (q_{rs})，末端直接接地。每個路段都個別模擬成電阻，並將初始值電阻值 R_a^0 設為 $1K\Omega$ 。將所子路網系統得到的流量合併成為解集合 $x_a^0, \forall a$ (亦為 \mathbf{x}^0)。

步驟 1：轉換道路成本，現在將每個子路網中路段的成本函數，設定成為 $u_a(\omega) = \max(t_a(0), t_a(x_a^n) + t_a'(x_a^n)(\omega - x_{a,\pi}^n))$ ，其中 $x_{a,\pi}^n$ 為子路網經過路段 a 之流量。

步驟 2：找尋方向，根據新的成本函數 $u_a(\omega)$ ， $\forall \pi \in \Pi$ 執行 PCA-Sub，以合併所有 PCA-Sub 的解當作搜尋方向 $y_a^n, \forall a$ 。

步驟 3：線性搜尋，利用 $\min_{0 \leq \alpha \leq 1} \sum_a \int_0^{x_a^n + \alpha(y_a^n - x_a^n)} t_a(\omega) d\omega$ 找出最適步幅 α^n 。

步驟 4：找尋新解，設定新解為 $x_a^{n+1} = x_a^n + \alpha^n (y_a^n - x_a^n), \forall a$

步驟 5：測試收斂，若滿足所設定的收斂條件，就視同收斂並停止計算；若不是，則回到步驟 1 繼續計算。

此與上小節演算法不同，步驟 0 中，3.3 節之演算法將初始電阻值設定為 1，但是在實際電路上我們無法設定電阻值為 1，故我們將初始值電阻值 R_a^0 設為 $1K\Omega$ 。在步驟 2 中，3.3 節所執行的子演算法為 CSA-Sub，這裡我們需要執行實體電路運作時的子演算法 PCA-Sub，這些是主要的不同之處。

第四章 實作的誤差與影響

4.1 量化誤差的定義

一般我們所採用的數字系統皆為十進位系統(decimal system)，也就是以 0、1、2、...、9 等十個數字作為計數的基底；但是想要控制數位電阻之電阻值，僅能以二進位系統(binary system)的訊號輸出來控制，也就是以 0、1 兩個數字為計數之基底，換言之，任何的十進位數字都須用 0 或 1 來表示，以八位元的二進位系統為例，十進位的 1 代表二進位的 00000001；十進位的 2 代表二進位的 00000010，以此類推。故我們想要控制數位電阻的值，就必須以此二進位的訊號來控制它。表 3.1 顯示電阻的階數變化所顯示的各階電阻值。

表 4.1 數位電阻階數與實際阻值間的關係(以最大值為 10KΩ 為例)

階數(十進位)	階數(二進位)	實際電阻值
0	00000000	0.0000Ω
1	00000001	39.2157Ω
2	00000010	78.4314Ω
3	00000011	117.6471Ω
...
254	11111110	9960.7843Ω
255	11111111	10000.0000Ω

資料來源：本研究整理

3.2 小節中說明，先前演算法在演算時，沒有考慮到電阻值「量化誤差」的問題。何謂量化誤差？量化誤差就是量化的結果與未量化前實際值之間的差值。由於數位可變電阻的特性限制，造成並非想要什麼電阻值都可以得到。舉一個最大值為 10KΩ(即 10000Ω)，且為 8 位元共 256 階變化的電阻為例，其電阻值僅有 256 階的變化而已。所以，當我們在計算時，得出某次遞迴之實際阻值為 50Ω，但我們卻沒有辦法實際將阻值調整至 50Ω，僅能以 39.0625Ω 或 78.125Ω 來顯示，這就是量化的誤差。

顯然地，在一個範圍值內的量化階數越多，量化的誤差會越小。故我們在電路製作時，一定要將量化誤差考慮進去。原先的數位可變電阻之電阻值為 R_a^n ，而各電阻值分別經過量化後，阻值將會轉變成 Q_a^n 。

4.2 電阻值量化的影響

實作電路演算法是根據前述的電路模擬演算法所推衍出來的，因為在實際電路中，所使用的元件有著許多限制，在 3.2 節中已提及，電阻本身存在著量化的誤差，量化誤差對於電路的實作可能會造成影響。

受限於軟硬體的限制，以及操作上的便利性等等，因此我們選擇 256 階的數位可變電阻來做本實作。事實上，我們發現量化的階數對於目標值的影響而言，並不會太過於劇烈，詳細情況將在後面說明。

在電路模擬演算法與實作電路演算法這兩個演算法中，後者新增新的演算步驟。除了前述最大阻值的限制外，步驟 2 將阻值量化，得到一個新的量化電阻值 Q_a^n ，此處就有量化的誤差。也因為這樣，後面的電流與電壓計算，也都有造成一定程度的不精確。

儘管因為量化之故，造成了演算法會提早收斂，但是，所求得的解與理論值之間，差異可說是非常的微小，這顯示利用本演算法，量化所造成的誤差對於最佳解而言，影響並不明顯。但在電阻有 256 階的情況下是如此，那麼，我們現在想要探討的是，量化的階數大小，對於答案的精確度是否影響很大呢？以常理判斷之，當電阻的量化階數越少的時候，答案應該會更加的不精確，因為量化後所得之電阻值，離實際的阻值應該會相差的更多，所以，我們現在使用電腦模擬的方式，並以圖 4.1 的範例為例題，此小型範例可以模擬成圖 4.2 的路網，並套入 3.2 節的演算法做電腦的模擬，用以探討量化階數的變化對於實際阻值以及目標值的影響大小，結果如表 4.2：

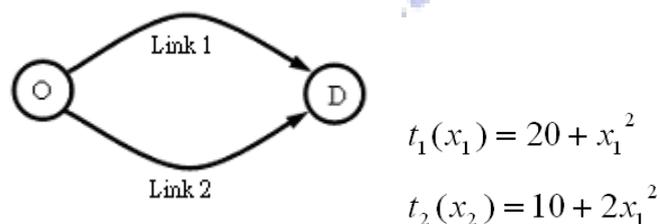


圖 4.1 電腦模擬測試路網圖

資料來源：(Huang, 2007)

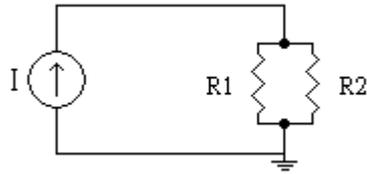


圖 4.2 電腦模擬測試電路圖

資料來源：本研究整理

表 4.2 量化階數不同對於目標值之影響

階數	遞迴次數	x_1	x_2	目標值	目標值誤差
		11.540	8.460	1231.331	
256	*19	11.511	8.489	1231.353	0.00182%
128	*18	11.575	8.425	1231.357	0.00200%
64	13	11.513	8.487	1231.351	0.00165%
32	11	11.494	8.506	1231.389	0.00473%
16	7	11.635	8.365	1231.588	0.02091%

資料來源：本研究整理

註：*號表示該次運算結果，會發生如同 3.2 節所述之收斂值不唯一的現象。在 256 階的實驗中，第 20 次遞迴之解為(11.562, 8.438)，目標值 1231.345，誤差為 0.00117%；在 128 階的實驗中，第 19 次遞迴之解為(11.524, 8.476)，目標值 1231.337，誤差為 0.00050%

令人驚異的發現，儘管阻值是經過量化之後的電阻值，但是其對於各路段之流量解的影響卻是相當的有限，因為對於解的影響不大，使得各目標值也與理論目標值之間差距甚微；而隨著階數的減少，目標值的誤差雖然越來越大，但是影響幅度有限。

4.3 元件的誤差

電阻除了量化的誤差之外，本身還存在有其他的誤差。以市售的一般型電阻為例，每一顆電阻都只有一個固定的電阻值，即所謂「定值電阻」，但其理論電阻值與實際電阻值之間往往存在著誤差，這是因為現實的電阻器本身除電阻外，還擁有著微量的電感或電容，使其阻值與理想的電阻器有所差異。一般電阻的誤差約在 5% 左右，這也就是說電阻的最大值不固定，即所謂的元件誤差。

電子元件的各種標示值與實際值之間常常存在著誤差。同樣地，我們所使用的數位可變電阻也有著類似的誤差，因為每一顆元件的最大值都不一樣，甚至是實驗當時的溫度、溼度等變化，對於元件的精準度都有影響。以理論最大值為 10KΩ 的數位可變電阻來說，據實際量測的結果，此型數位可變電阻的最大值約在 9.5KΩ ~ 10.5KΩ 之

間，而最大值與其量化後每一階的距離有關係，電阻最大值較大，電阻每階的間距也較大。我們實際的測量了 30 個數位可變電阻，且每個電阻分別設定小、中、大三種不同的電阻值，來驗證是否每一個電阻間存在著差異，結果如表 4.3。

在這裡我們計算了變異係數的大小。在一般的情形下，要比較多組資料的差異，不能只比較標準差的大小，需要一種相對的測度值作為比較的標準。變異係數的定義如(4-1)式：

$$CV = \left(\frac{s}{\bar{X}} \right) \times 100\% \quad (4-1)$$

變異係數就是一種相對測度值，其中 \bar{X} 為算術平均數， s 為樣本標準差。這裡發現，隨著設定的電阻值越高，雖然實際電阻值的標準差越來越大，但是其變異係數並沒有明顯的變化。由此試驗我們可以得知，各電阻樣本之間確天生存在著差異性。

4.4 量化誤差與元件誤差的綜合影響

既然電阻間存在著差異，則它對於實際電路求解的影響會有多少？我們想利用電腦模擬的方式，合併考量前一小節量化誤差與本節電阻間之元件誤差，探討其對於目標解的影響。現在利用電腦隨機取樣兩個電阻值，範圍在 $9.5K\Omega \sim 10.5K\Omega$ 之間，我們得到 $9.7K\Omega$ 與 $10.4K\Omega$ ，並以此作為兩電阻之最大阻值，以此為基礎，做電腦的模擬運算。接下來，再次使用圖 4.1 的範例為例題，且將之模擬成圖 4.2 的路網。

表 4.4 為模擬之後的結果，可以發現，相較於前面量化階數對目標值的影響，合併考量兩種誤差之後，其對於目標值的影響似乎更大，且也同樣是量化的階數越少時，目標值的誤差會越大，故我們在實際製作電路上，必要更加注意其所造成之影響。

而除了電阻之外，經由類比/數位轉換器將類比電壓轉換成數位訊號，同樣地也要將電壓的量化誤差計算進去，但是，經過實驗的發現，電壓的誤差影響也並不大，下一章將有實際的演算範例。

表 4.3 三十個數位可變電阻之設定電阻值與實際表示阻值

編號\階數	10	100	200
1	0.42	3.77	7.5
2	0.44	3.98	7.87
3	0.44	4.00	7.93
4	0.45	4.01	7.91
5	0.43	3.88	7.60
6	0.44	4.01	7.97
7	0.43	3.90	7.70
8	0.43	3.89	7.61
9	0.40	3.54	6.92
10	0.44	3.85	7.58
11	0.39	3.55	6.91
12	0.43	3.81	7.58
13	0.43	3.85	7.55
14	0.41	3.77	7.42
15	0.43	3.87	7.52
16	0.42	3.82	7.50
17	0.43	3.90	7.52
18	0.40	3.61	7.11
19	0.41	3.70	7.32
20	0.40	3.62	7.18
21	0.42	3.79	7.53
22	0.41	3.65	7.23
23	0.43	3.91	7.85
24	0.43	3.90	7.76
25	0.43	3.84	7.64
26	0.44	3.98	7.87
27	0.43	3.88	7.66
28	0.43	3.85	7.50
29	0.42	3.85	7.59
30	0.41	3.72	7.43
平均	0.424	3.823	7.542
理論值	0.43	3.94	7.84
標準差	0.0145	0.1308	0.2737
變異係數	3.43%	3.42%	3.62%

資料來源：本研究整理

表 4.4 量化誤差與元件誤差合併考慮之影響

階數	遞迴次數	x_1	x_2	目標值	目標值誤差
		11.540	8.460	1231.331	
256	*24	11.628	8.372	1231.553	0.0180%
128	*16	11.678	8.322	1231.872	0.0439%
64	14	11.664	8.336	1231.774	0.0360%
32	*8	11.975	8.025	1236.693	0.4354%
16	*4	12.242	7.758	1245.280	1.1329%

資料來源：本研究整理

註：*號表示該次運算結果，會發生如同 3.2 節所述之收斂值不唯一的現象。在 256 階的實驗中，第 25 次遞迴之解為(11.653, 8.347)，目標值 1231.695，誤差為 0.0296%；在 128 階的實驗中，第 17 次遞迴之解為(11.628, 8.372)，目標值 1231.553，誤差為 0.0180%；在 32 階的實驗中，第 9 次遞迴之解為(11.357, 8.643)，目標值 1232.282，誤差為 0.0773%；在 16 階的實驗中，第 5 次遞迴之解為(10.973, 9.027)，目標值 1240.513，誤差為 0.7457%。

4.5 元件的物理限制

在演算法迴圈進行的時候，可能會遇見其遞迴電阻值，大於數位可變電阻之最大可顯現值的狀況，以最大值為 10KΩ 的數位可變電阻來說，最多也只能顯示出 10KΩ 左右的電阻值，但是迴圈過程很可能發生電阻值超過 10KΩ 的狀況，乍看之下這會對於實際上的求解產生限制，故在實作的演算法中我們將要考慮這樣的因子。

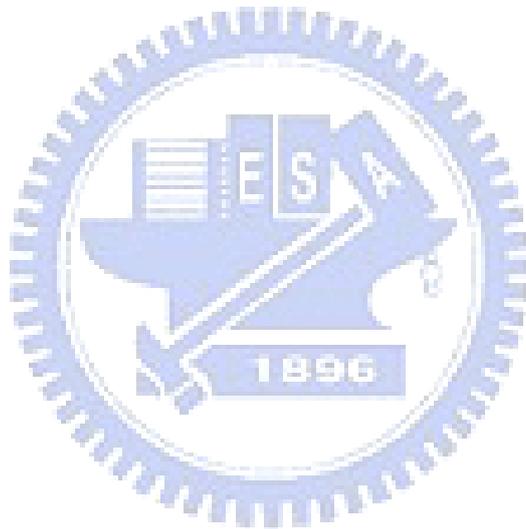
但事實上，這對於我們實際電路的求解並不會產生非常大的影響，因為只要各個電阻間的阻值比是固定的，其所通過的電流比也一定會是固定的。以圖 1.4 的簡單雙路段路網為例，假設在某次遞迴時，兩路段的阻值分別為 12KΩ 與 21KΩ，故兩路段之電阻值比為 12:21，即 7:4。現根據歐姆定律，發現通過電阻之電流比將一定會是 7:4，但是電阻只能顯示最大值 10KΩ，故我們只要調整兩路段的阻值，如 1.2KΩ 與 2.1KΩ，其阻值比是 7:4，則通過之電流比將也會是 4:7，以這樣的方式，就可以克服元件本身限制的問題。

除此之外，當電阻設定值過大，使得整個網路的等效電阻乘以定電流值超過了 V_{\max} ，也就是全網路最大電壓差時，會造成定電流源電流的下降。全網路的總壓降受限於電源供應器本身所供給之電壓限制，不可能超過 5V，此處的 V_{\max} 就等於 5。3.2 的演算法提到了比值 k，該比值的設定是事先估算過的，透過計算，使得實際通過電路路網的定電流值或是電阻值不會過大，讓該電路路網之總壓差不會超過 V_{\max} ，超過的話將會造成定電流的下降。

那定電流值的下降是否會對求解造成影響？答案是不會的，因為就算電流值會下降，其所分配到各個電阻之間的電流值也會同等比例地下降，根據 3.2 節的 PCA-Sub，

只要各路段的電流下降比例一致，我們就可以利用演算法將之修正回來。

除此之外，元件本身所能通過的電流也不能太大。文獻中指出，數位可變電阻所能通過之電流不能超過 5mA，故這對演算法也會有一些限制存在，不過無論電流通過多寡，其各電阻之間通過電流的比例也會一樣的，故我們所發展出的演算法也要針對此作處理。



第五章 微處理機系統之電子電路實作

本章進入了實際製作電路的主題。5.1 節將會介紹整個電路系統的架構圖，還有所使用元件的簡單介紹；5.2 節就是單起訖點對的電路實際製作，以及實驗結果；5.3 節則是兩起點對一訖點的電路製作與結果；5.4 節為較為一般性的多重起訖點電路製作及結果；最後 5.5 節則是增加了路網供需改變後的一些測試結果。

5.1 系統架構圖與元件介紹

為了模擬第三章演算法中，不斷遞迴計算的運算過程，在電路的製作上，我們必須利用單晶片微電腦來做控制。單晶片微電腦，顧名思義就是將各種有關電腦的單元，都製作在同一個晶片上，藉此達成控制的動作。在本實作中，我們使用了 NXP 公司所生產的 P89V51RD2 這種 MCS-51(MCS: Micro Computer System)單晶片來做處理架構的核心，這也是目前世界上非常普遍的一種單晶片微電腦。P89V51RD2 單晶片之說明，如附錄一所示。在本微處理器中，共有 32 個輸入/輸出腳位，我們將會使用 P0.0~P0.7 這八個腳位當做數位電壓值的輸入，而微處理器內部可以將此數位訊號換算成類比訊號值，作為演算法運算之用。運算結束後，所得之電阻值再經由輸入/輸出腳位傳至數位可變電阻(digital potentiometer，簡稱 digipot)，以更改電阻值。

又由於電阻值必須不斷地變動，數位可變電阻的使用是必須的。我們選用 Analog Devices 公司所生產，型號為 AD8400 的數位可變電阻來實作電路。本元件的最大電阻值為 $10K\Omega$ ，共有 256 階，也就是說，將電阻值 $10K\Omega$ 平均分配 256 階，理論上每 $(10000/256)\Omega$ 為一階，第一階為 $10000/256 = 39.0625\Omega$ ，第二階為 $(10000/256) \times 2 = 78.125\Omega$ ，以此類推。

由於實際電路理論的限制，電阻的第 0 階應為 0Ω ，又根據式(1-1)的歐姆定律，電阻 $R = 0$ 會導致電流會趨近於無限大，過大的電流將會燒毀電阻，故電阻第 0 階時，為了保護元件，本身會有一個約 40Ω 的電阻值，用意為保護電阻，故實際每一階的距離應為 $(10000 - 40)/256\Omega$ 。該元件之腳位說明如附錄二所示，這是一個八腳位的電子元件。

前面有提及，在遞迴的過程中，我們有需要將電路所實際量測的電壓，傳送到微處理器的步驟，這時就需要使用類比/數位轉換器(A/D Converters，簡稱 ADC)，它可以將類比的電壓值轉變成為數位訊號值，因為微處理器的輸入以及輸出，都是使用數位信號，故將類比電壓值轉成數位訊號有其必要性。我們所使用的是 ADC0804 這個型號的 ADC，它能把輸入的類比電壓值，轉換成 256 階的數位電壓值作輸出。較詳

細的 ADC0804 腳位說明如附錄三所示。

因為需要利用微處理器控制的元件數較多，但 P89V51RD2 中也只有 32 個輸入/輸出埠而已，我們將會面臨到腳位不夠的問題，這時就需要利用 74164 這個 IC。這是一個八位元的移位暫存器，我們主要是利用它來控制各個元件的 CS(chip select)腳位，該腳位可以控制元件為啟動或是關閉。一般而言，當 CS 為低電位，也就是 CS=0 時，元件啟動；反之，元件關閉。我們僅須適當調整 74164 的輸入部分，在輔以相對應的時脈觸發，便可以達成依序分別啟動元件的目的，如此做將可以解決微處理機腳位不足的問題。詳細的 74164 腳位與真值表，如附錄四所示。

在程式的撰寫上面，所使用的是德國 Keil 公司所開發出的 μ Vision 軟體，這是一個整合性的軟體，可以用在 C 語言的撰寫以及組合語言的應用上。此軟體把專案的管理、程式原始碼的撰寫、編譯、偵錯；模擬等等功能都整合在一起，且還內涵許多 MCS-51 單晶片的系統和周邊的設備環境設定等等，目前也是全球使用最廣泛的單晶片程式撰寫工具。以下對於電路的實際流程做說明：

1. P89V51RD2 內設定數位電阻值，將前次迴圈所得之電阻值，轉換成數位信號，透過 P1.1 腳位輸出，依序將值輸出至各個數位可變電阻內。途中須用 P1.6 腳位輸入二位元值到 74164 內，讓 74164 可控制數位可變電阻的 CS 腳位，當 CS=0 時，電阻將會啟動。之後，電阻將會顯示量化後之阻值。
2. 數位可變電阻的電阻值皆已設定完畢，電流流經電阻會產生電壓差，此電壓差的類比訊號，在 CS=0 為低電位時，可藉由 ADC 之 $V_{in}(+)$ 及 $V_{in}(-)$ 腳位感應，而 CS 的控制同樣需藉由 74164 達成；接下來控制 WR，當正時脈邊緣觸發時，類比電壓值開始轉換成數位電壓值，RD=0 時數位電壓值資料開始透過 DB0~DB7 之腳位輸出，不過此處 RD 腳位會直接接地，如此較為簡便。
3. 數位輸出信號由 P89V51RD2 之 P0.0~P0.7 腳位接收，且微處理器內部會將數位信號計算回類比電壓值，再藉由演算法的運算，可得出下一次迴圈時所需之電阻值。
4. 在演算法運算的時候，可以透過 P3.0 腳位的串列埠接收，以及 P3.1 的串列埠發射，將各種數值(如電壓值、電流值.....)經由 MAX232 元件之轉換，發送至個人電腦，而個人電腦可藉由超級終端機此程式來接收資料，並將其顯示於螢幕上觀看。

圖 5.2 所代表的是整個系統的架構圖，圖中被方框框起的數位電阻及類比/數位轉換器部份，所代表的就是整個實際的交通路網。從圖中可以瞭解，微處理器(P89V51RD2)可以輸出阻值，再藉由 74164 控制數位電阻，使數位電阻顯示相對應之電阻值。而流經數位電阻的電流造成了壓降，其壓降就由類比/數位轉換器(ADC0804)

來負責測量。各個 ADC0804 所得之電壓資料，傳送至微處理器，讓微處理器可以藉由演算法，求出下一次遞迴時，需要輸出至數位電阻之電阻值。

在電路演算的過程中，運算是藉由微處理機的控制達成的，欲得知計算時各變數的值，可將電路與個人電腦，每當微處理機計算出某變數之值，便可將其傳送至電腦上顯示。個人電腦端的接收平台是利用 Windows 內建的超級終端機完成，圖 5.1 為超級終端機的基本顯示介面。

在進行實作前，先在考慮有量化誤差的情況下，做電腦上的模擬演算，以觀察誤差對於最佳解、以及目標值的影響有多少。我們在每一次的迴圈中，所求出之理論的電阻值，都要將他表示成經過數位量化後，實際上數位可變電阻所能顯示的阻值，且電壓經過 ADC0804 轉換的誤差也要考慮進去。



圖 5.1 超級終端機顯示介面

資料來源：本研究整理

5.2 子路網之實際電路製作

本節中將會進入實際製作的主題，首先我們利用 3.2 節中提及的子系統之實際電路演算法(PCA-sub)來製作實際電路。5.2.1 的實作路網較為單純，僅含有一組起訖對與兩個路段；而 5.2.2 的路網則包含了兩起點與一訖點，總共有六個路段。

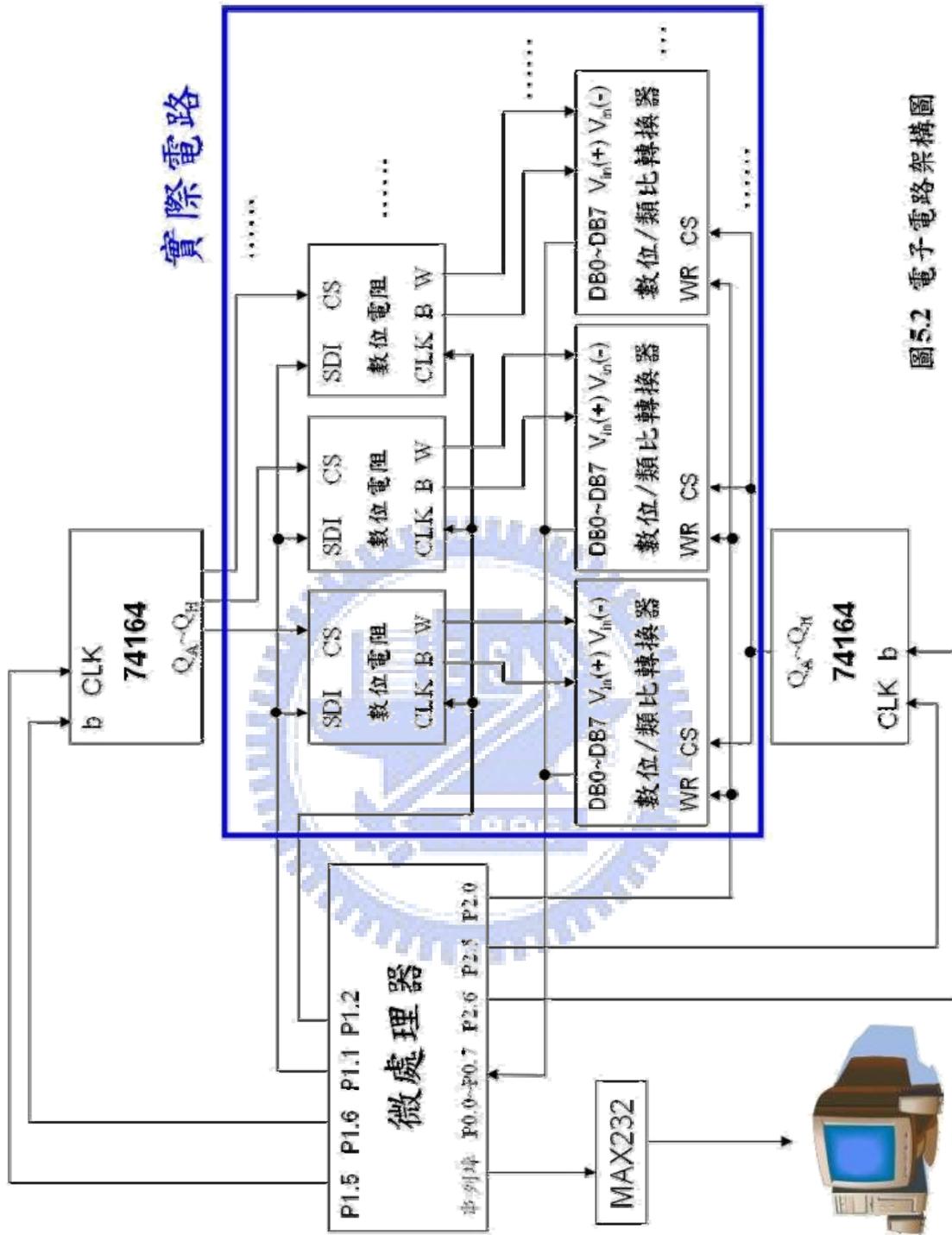


圖5.2 電子電路架構圖

5.2.1 單起訖點對之電路製作

現在考慮一單 OD 點對的路網，為了簡化計算的過程，我們所挑選的電路，在遞迴過程中均沒有超過 $10K\Omega$ 的現象產生。我們選擇如圖 1.4 之路網，並隨機挑選兩個數位可變電阻，作為路網中的路段，其成本函數如下：

$$c_1(x_1) = 1 + 6x_1^2$$

$$c_2(x_2) = 2 + x_2^2$$

$$q_{rs} = 1$$

$$\text{目標式 } z(x) = x_1 + 2x_2 + 2x_1^3 + \frac{1}{3}x_2^3$$

理論上，本範例最佳解應為 $(i_1, i_2) = (0.4633, 0.5367)$ ，目標值為 1.78712。而有經由量化過程的系統最佳解為 $(i_1, i_2) = (0.4641, 0.5359)$ ，目標值為 1.78713，詳細計算過程不再贅述。

此處之 OD 流量為 1，作如此設定的原因是因為實際硬體電路的限制，電流不可以超過 $5mA$ 之故，所以先利用低電流的電路來驗證電路實作的可行性。實際電路如圖 5.3：

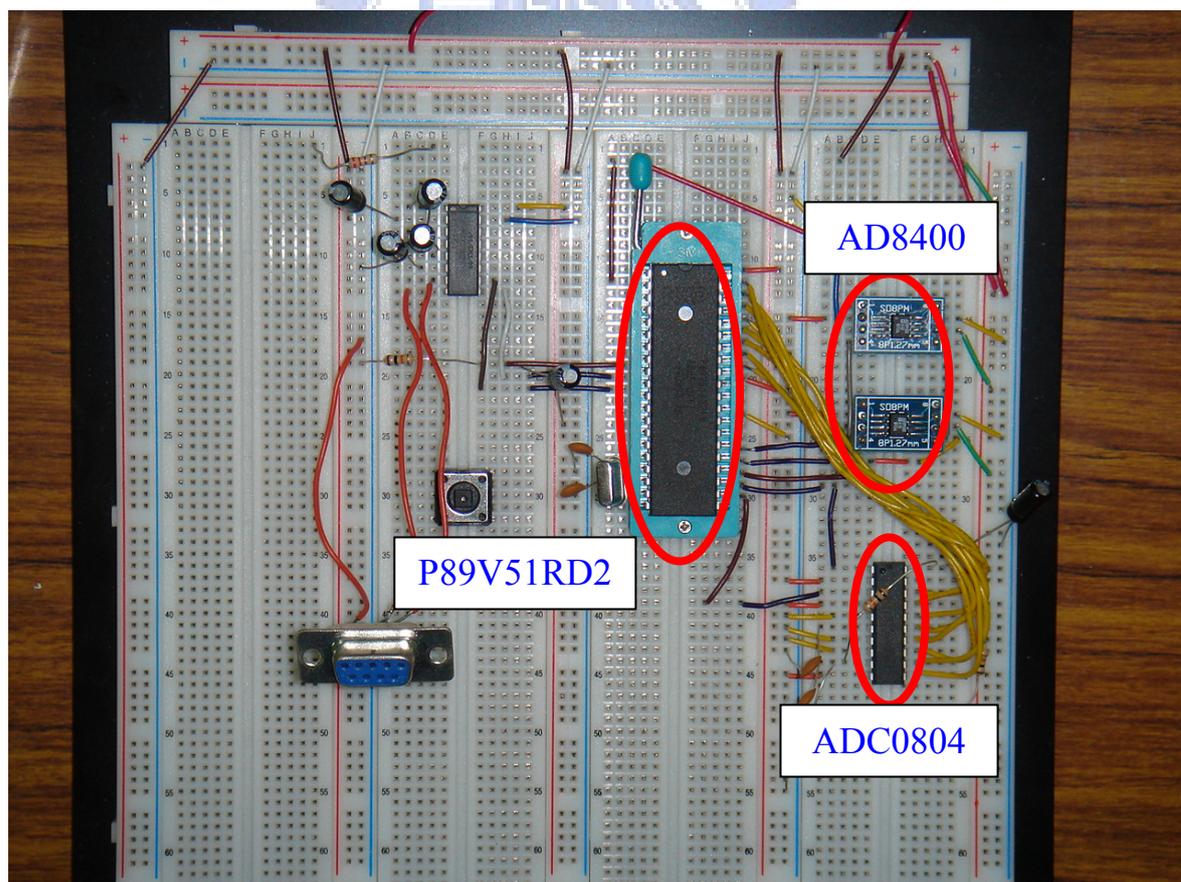


圖 5.3 單起訖點對之電路實作圖

資料來源：本研究整理

解說：

1. P89V51RD2：主要微處理器，程式在此進行運算，可將電阻值以串列方式傳至數位可變電阻，再接收 ADC 所傳過來之並列電壓值做運算。
2. AD8400：數位可變電阻，可依傳入之串列訊號的改變而改變其電阻值。
3. ADC0804：類比/數位轉換器，將類比的電壓值轉變成數位且並列之值輸出至微處理器。

本題之理論上最佳解為 $(i_1, i_2) = (0.4633, 0.5367)$ ，電阻值 $(R_1, R_2) = (4.938, 4.263)$ ；而實際電路所測量出之結果為 $(i_1, i_2) = (0.461, 0.539)$ ，電阻值 $(R_1, R_2) = (4.93, 4.22)$ ，實做電路之結果與理論結果相去不遠。

由 4.3 節得知，每顆電阻所擁有的最大值不同，連帶使得階數的電阻值也略有不同，導致就算某兩顆電阻其階數相同，經由量化後的實際電阻值也可能不同，現在我們想知道，這樣的狀況對於實際求解是否有影響。現在我們仍然測試圖 4.3 之例題，但現在隨機挑選 30 個電阻作更換，再求取本例題之解。所得到的 15 組解如表 5.1：

表 5.1 單起訖點例題隨機更換電阻之 15 組解

組別	I_1	I_2
1	0.461	0.539
2	0.461	0.539
3	0.456	0.544
4	0.457	0.543
5	0.449	0.551
6	0.452	0.548
7	0.458	0.542
8	0.465	0.535
9	0.452	0.548
10	0.457	0.543
11	0.460	0.540
12	0.458	0.542
13	0.456	0.544
14	0.455	0.545
15	0.457	0.543

資料來源：本研究整理

經由上述的實驗結果發現，元件的誤差等影響，對於實際電路求解來說影響不大，且各電阻通過之電流與理論值之間的差距，大約也都不超過 5%。透過這樣的小型範例測試，說明元件的誤差及量化的影響應為有限。前述之電路實做結果，電阻值

的解也與理論值有一些差距，究其原因，可能如下：

1. 量化的誤差

如前一小節所述，將前一步驟所計算得出之電阻值轉成數位信號，並將其輸入至可變電阻值，將會有量化誤差產生，且使用 ADC0804 量測電壓時，同樣也有量化的誤差，因這兩者都是 256 階的 IC，以 ADC0804 來說，最大可顯示之電壓為 5V，共有 256 階，所以大約每 0.02V 為一階，故某些電壓值無法顯示是必然的。

2. 元件本身的誤差與演算法的限制

前面已說明過，我們所使用的數位可變電阻，雖然號稱最大值為 10KΩ，但其實際最大值隨著元件的不同，甚至是實驗當時的溫度、溼度等不同，而會有所變化。如前所述，數位可變電阻的最大值可能從 9.5KΩ ~ 10.5KΩ，可將近有 1KΩ 之差距，但是，我們在撰寫微處理器內部程式碼時，卻沒有將所有元件的最大電阻值都考慮進去來計算，而是一律使用最大電阻值為 10KΩ 計算。最大可能之電阻值的不同，使得電阻每一階的距離也稍有不同，所以，當我們認為在某次迴圈下，將要給某電阻值一個數位信號，使之轉換成類比的電阻值時，電阻所實際顯現出來之電阻值，卻可能與預期該得到之值之不同，這將會對於演算的準確性造成影響，惟其影響較有限。

5.2.2 兩起點對一訖點之電路實際製作

儘管實際電路擁有的誤差，但是其最終結果還算令人滿意，結果的誤差也沒有很大，所以，我們想要製作更大的電路繼續來驗證電路實驗的可行性，接下來我們挑選了如圖 5.4 的路網圖，該路網有兩起點、一訖點，各起點至訖點之流量均為 1，實際製作之電路如圖 5.5：

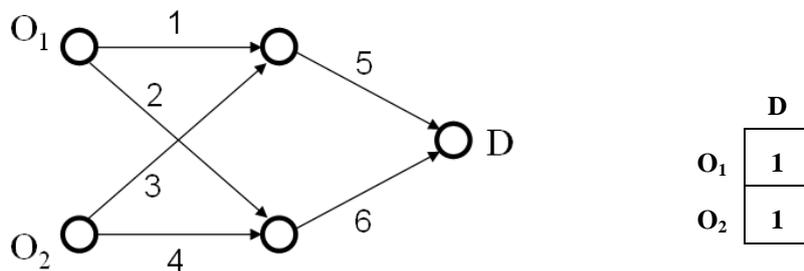


圖 5.4 兩起點對一訖點之測試路網

資料來源：本研究整理

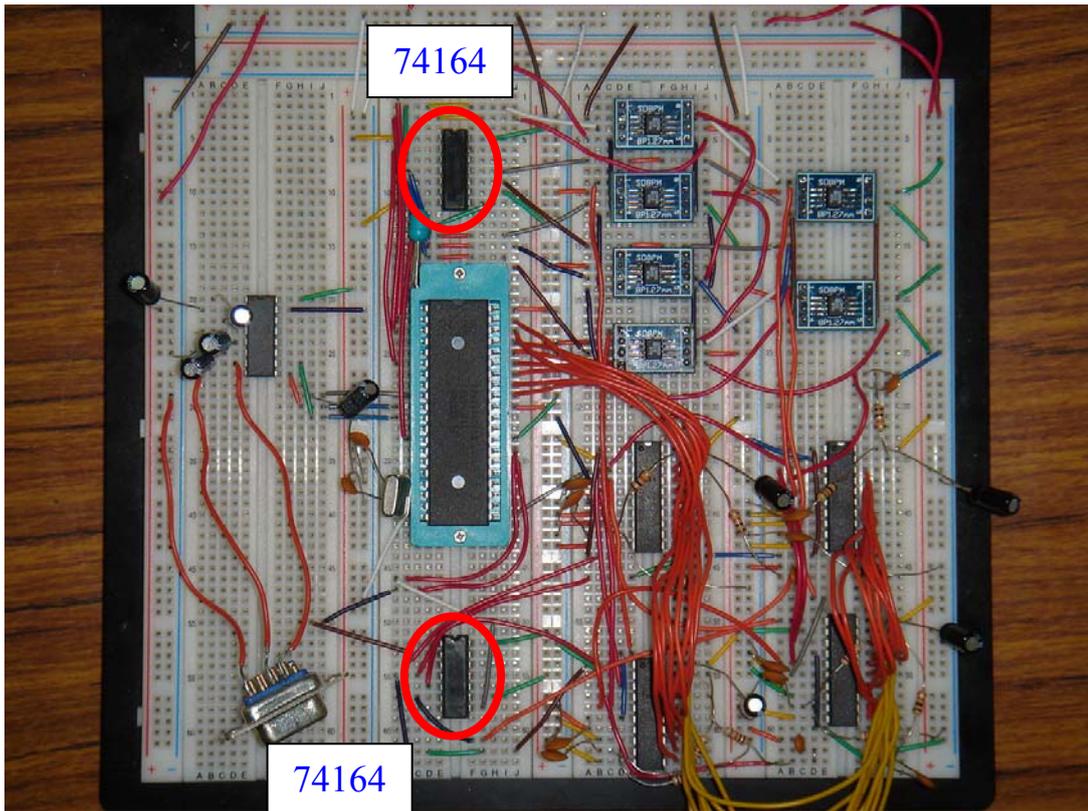


圖 5.5 兩起點對一訖點之電路實作圖

資料來源：本研究整理

本電路中，除了原本的 P89V51RD2、ADC0804、及數位電阻等元件外，另外增加了 74164 這個 8 位元的移位暫存器(架構圖如附錄)，其主要用途是將所輸入之串列資料轉換成為 8 位元的並列資料輸出。而原因是因為當微處理器所需要控制的電阻越來越多時，將會使微處理器本身的腳位不敷使用。以 P89V51RD2 為例，總共只有 40 個腳位，而實際能用來作為輸入或輸出之腳位，更是只有 32 個，所以我們透過此移位暫存器，依序的控制數位可變電阻以及類比/數位轉換器(ADC)，以達到節省腳位的目的。而本路網各路段之成本函數可如表 5.2 所示，數值計算之解如表 5.3，電路實際測量所得之解如表 5.4：

結果分析

所得之電路實做結果與實際值同樣有一段差距，但是差距並不很大，流量的最佳解與理論之最佳解之間，大約有 5%以內的誤差，此誤差就結果來看，比之前的單 OD 網路誤差要來的更大一些。原先我們認為，前述的量化誤差、元件誤差等等，可能隨

著路網的增大，誤差之間可能相互抵銷，但事實並非如此，後續將會對此議題做更深入的研究。

表 5.2 兩起點對一訖點之路段成本參數

$$\text{成本函數 } t_a(x_a) = p_1 + p_2 x_a^4$$

路段 a	p_1	p_2
1	0.8	0.50
2	0.6	0.60
3	1.0	0.45
4	1.0	0.40
5	1.2	0.20
6	1.0	0.70

資料來源：本研究整理

表 5.3 兩起點對一訖點之理論與實際電路之各路段流量

路徑	理論流量	實際流量
1	0.356	0.368
2	0.644	0.632
3	0.700	0.721
4	0.300	0.279
5	1.057	1.080
6	0.943	0.920

資料來源：本研究整理

5.3 系統電路實際製作

將前一小節的路網繼續擴展，現在希望製作多起訖點之實際路網。根據 3.3 節的多重起訖點之電路模擬演算法，多起訖點演算法在計算時，是將全路網拆解成多個單起點對多訖點的子路網，並將所有子路網所得到的流量合併，再經由道路成本的轉換、搜尋至最適步幅等等的步驟，求得新解。我們所測試的路網圖結構如圖 5.7，這是一個擁有 6 節點，以及 8 路段的問題，且各個起訖對之間的流量均為 1：

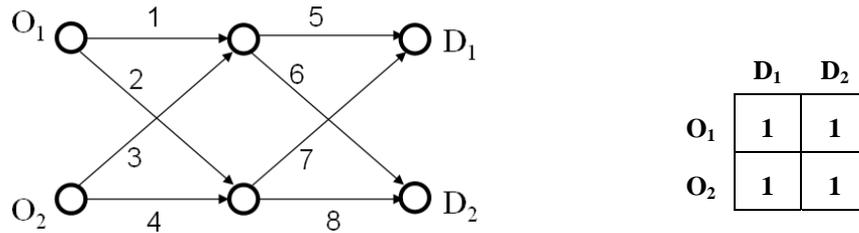


圖 5.6 兩起點對兩訖點之測試路網(各 OD 間流量=1)

資料來源：本研究整理

按照之前的多重起訖點之電路模擬演算法，很快的發現了實際電路上的問題，導致我們沒有辦法實際的利用 8 個可變電阻來代表 8 個路段。由於在演算法中，是將全路網拆解成多個單起點對多訖點的子路網，那麼，理論上沒有使用到的路段應該不會有任何電流量的通過，但是實際上這是不可能的。倘若我們一開始就將所有的電阻都安置在電路板上，那麼，演算法中未使用的路段仍會有電流的通過。以圖 5.7 為例，假設我們所設定的子路網為 1-2-5-6-7-8 與 3-4-5-6-7-8，則當我們在為 1-2-5-6-7-8 路網進行演算時，因為路段 3、4 是與其相接的，則勢必會有電流通過路段 3 與 4，使得演算時所得到的子路網電流狀況，必不能代表當時實際的子路網電流。當然，理論上也可以將未使用路段電阻設定為無限大，如此一來依據歐姆定律，這些路段的流量就會變成 0，但是實際上我們是沒有辦法將電阻設定為無限大的，在我們的例子中，電阻值最大僅能設定成 10KΩ。

那麼，又該如何解決在計算子路網流量時，未使用路段卻也有流量通過的情況發生呢？以圖 5.7 為例，在計算流量時，我們會將其拆成 1-2-3-4-5-7 與 1-2-3-4-6-8 兩個子路網，這兩個子路網的架構為對稱，且與圖 5.5 架構類似。事實上，演算法計算時，我們的確只會用到子路網來求取當時相對應的電流、電壓、電阻等數值，總流量的計算是在所有子路網流量都求得並相加後，再經由線性搜尋而得，故當我們實際製作路網時，實際上也只需要接出如圖 5.5 的子路網，再經由微處理機的運算就可以了。所製作出的實際電路如圖 5.6，這和兩訖點對一訖點的實際電路是一樣的。本路網各路段之成本函數可如表 5.4 所示，數值計算與電路實際測量之解如表 5.5：

表 5.4 兩起點對兩訖點之路段成本參數(各 OD 間流量=1)

$$\text{成本函數 } t_a(x_a) = p_1 + p_2 x_a^4$$

路段 a	p_1	p_2
1	0.40	0.25
2	0.50	0.40
3	0.60	0.50
4	0.30	0.30
5	0.60	0.25
6	0.25	0.25
7	0.25	0.30
8	0.40	0.30

資料來源：本研究整理

表 5.5 兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=1)

路段 a	理論流量	實際流量
1	1.106	1.086
2	0.894	0.914
3	0.841	0.913
4	1.159	1.087
5	0.864	0.936
6	1.083	1.106
7	1.136	1.064
8	0.917	0.894
目標值	3.7285	3.7428

資料來源：本研究整理

所得之電路實做結果與實際值仍有一段差距，同樣地，差距也不大，流量的最佳解與理論之最佳解之間，大約有 5%以內的誤差。並且，理論的目標值為 3.7285，與實作電路的目標值 3.7428 之間的誤差也不超過 1%。

5.4 系統容量測試

先前的試驗證明，利用電路求解流量問題的解應為可行，現在我們想擴大 OD 間的流量，以求解較為一般性的交通指派問題，因為一般的指派問題，不太可能發生起訖點流量僅為 1 的狀況，現在仍沿用圖 5.7 之路網，但是改變各 OD 間之流量如圖 5.8：

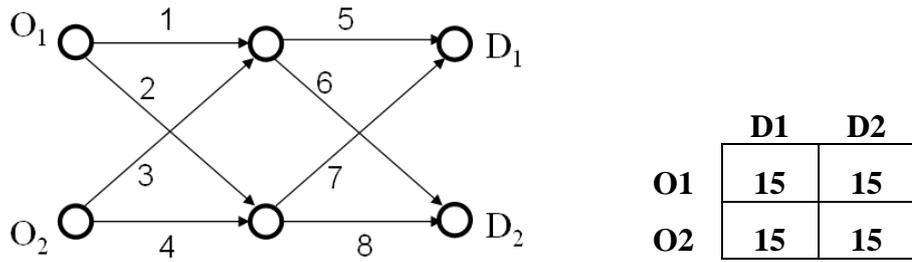


圖 5.7 兩起點對兩訖點之測試路網(各 OD 間流量=15)

資料來源：本研究整理

我們設定了一組新的路段之成本函數，如下表 5.6，每條路段都有其自由車流旅行時間與容量限制。以路段 1 為例，自由車流旅行時間為 2，其道路容量限制為 15。最佳解流量與電路計算解如表 5.7：

表 5.6 測試路段之路段成本參數(各 OD 間流量=15)

$$\text{成本函數 } t_a(x_a) = T_a \left(1 + 0.15 \left(\frac{x_a}{C_a} \right)^4 \right)$$

路段 a	T_a	C_a
1	2	15
2	2	30
3	3	15
4	3	25
5	2	10
6	2	10
7	3	15
8	3	15

資料來源：本研究整理

表 5.7 兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=15)

路段 a	理論流量	實際流量	誤差百分比
1	14.223	15.321	7.72%
2	15.777	14.678	-6.96%
3	13.549	13.575	0.19%
4	16.451	16.424	-0.17%
5	14.036	14.802	5.46%
6	13.736	12.926	-5.90%
7	15.964	15.197	-4.80%
8	16.264	17.173	5.59%
目標值	315.270	315.814	0.17%

資料來源：本研究整理

接下來，我們將會更改 OD 的流量做測試。根據定義上的公式，當流量開始超過道路本身的限制流量時，成本將會急劇的上昇，現在我們想研究這會不會對我們流量的求解有影響。現在我們分別加入各 OD 流量在 20、30、40 時的情形，讓道路的流量逐漸超過道路容量，成本函數如同前表 5.6，所得之理論解與實際測試解分別如表 5.8、5.9、5.10：

表 5.8 兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=20)

路段 a	理論流量	實際流量	誤差百分比
1	17.807	16.924	-4.96%
2	22.193	23.075	3.97%
3	16.340	17.007	4.08%
4	23.660	22.992	-2.82%
5	17.379	16.987	-2.25%
6	16.768	17.854	6.48%
7	22.621	23.012	1.73%
8	23.232	22.145	-4.68%
目標值	455.096	456.219	0.25%

資料來源：本研究整理

表 5.9 兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=30)

路段 a	理論流量	實際流量	誤差百分比
1	25.951	24.432	-5.85%
2	34.049	35.567	4.46%
3	23.476	24.404	3.95%
4	36.524	35.595	-2.54%
5	25.221	24.496	-2.87%
6	24.206	23.113	-4.52%
7	34.779	35.504	2.08%
8	35.794	36.886	3.05%
目標值	979.529	984.278	0.48%

資料來源：本研究整理

表 5.10 兩起點對兩訖點之理論與實際各路段之流量(各 OD 間流量=40)

路段 a	理論流量	實際流量	誤差百分比
1	34.414	33.001	-4.11%
2	45.586	46.998	3.10%
3	31.067	30.254	-2.62%
4	48.933	49.745	1.66%
5	33.429	34.025	1.78%
6	32.052	30.658	-4.35%
7	46.571	45.974	-1.28%
8	47.948	49.341	2.91%
目標值	2368.712	2373.124	0.19%

資料來源：本研究整理

由實驗結果發現，各 OD 間流量從 15 慢慢增加到 40，實際流量與理論流量間的差距大部份都在 5% 以內，至於目標值之間的差距更是都在 1% 以內。除此之外，理論上本例中，容量較高的路段流量增加的速度較快。以 O_1D_1 之起訖點對為例，從 O_1 到 D_1 其通過路徑可以選擇路段 1 與 5 或路段 2 與 7，示意如圖 5.9：

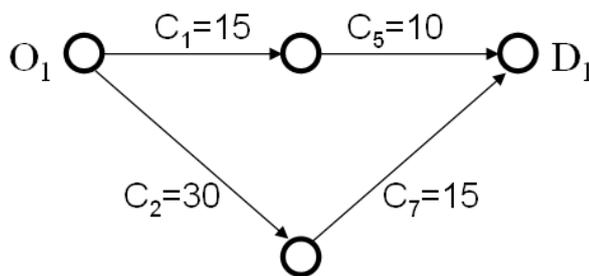


圖 5.8 起點 O_1 至訖點 D_1 之通過路段

資料來源：本研究整理

當起訖點間需求增加時，通過 1、5 路徑的時間成本就會越來越大，因為越來越接近道路的容量，故流量會轉移到 2、7 路徑，因為該兩路段之容量皆較大。而我們的實作結果顯示，實作電路確實也符合這樣的現象，實作的結果與理論結果事實上也相去不遠，而且也有表現出需求增加時，流量會向較高容量路段移轉的性質。

5.5 運算時間之估算

本研究的一中心假設為「量測」比「運算」來的迅速。當我們在求解最短路徑問題時，隨著節點與路段的增加，其計算時間會呈現一非線性的成長趨勢，但是如果利用直接以量測方式取得資料，量測時間與路段總數將會呈現一線性關係，這樣的特性使得在求解大型路網問題上，「量測」的優勢會比「運算」來的明顯，因為量測速度會比僅以運算要快上許多。

現在我們利用圖 4.4 的小型路網驗證此一假設之真實性，我們將會把最短路徑的 Floyd - Warshall 演算法、以及計算全有全無指派之演算法寫入微處理器內，此處，微處理器的運算速度，將會與實際利用微處理器控制各元件，並取得電流資料的時間做比較(即 PCA-Sub 的步驟 2 與步驟 3)。Keil C 軟體內有一偵錯(debug)功能，該功能可以顯示執行程式所需要耗費的時間，該功能界面如圖 5.9。結果得知，Floyd - Warshall 演算法的運算時間約為 0.0955 秒，而利用實際控制並量測電流資料，且演算法遞迴 10 次的時間，大約為 0.0575 秒。

比較後發現，在此小型的六路段路網問題中，利用實際控制並量測的方式，比起最短路徑演算法的運算時間要來的快，又因為最短路徑問題的運算時間，會隨著路段數增加而呈非線性的大幅增加，故我們可以推知，在大型路網問題的求解上，實際的電路控制與量測必定比最短路徑運算要快速許多。

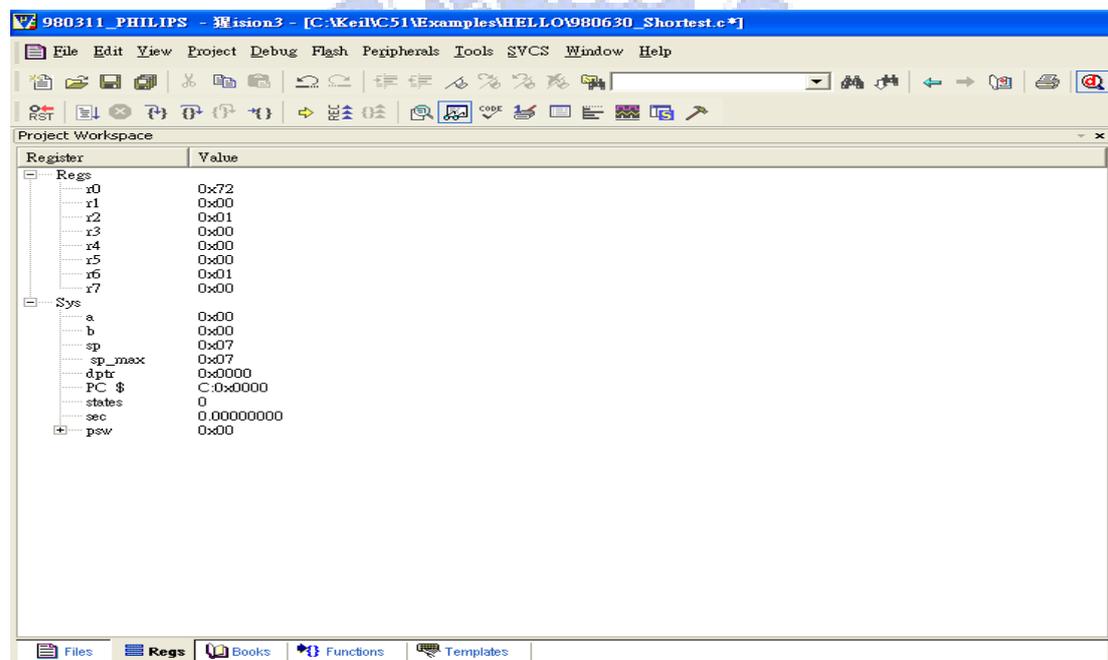


圖 5.9 Keil C 軟體偵錯介面圖

資料來源：本研究整理

第六章 結論與建議

本研究係以一實際電路求解交通指派之問題，而交通指派問題在運輸規劃的領域中是充滿挑戰性的一門學問，其最重要的特性就是使用者均衡的概念。在指派問題中，最常用的是 Frank-Wolfe 演算法，該法大部分時間都在解決最短路徑問題，故路網增大後效率較為不彰。

先前有文獻指出，交通路網的特性與電子電路有相同之處，Cho et al. 使用非線性的二極體元件模擬非線性之道路成本函數，但其限制較多，如要應用在實作上比較困難，而 Huang et al. (2007) 使用了較為單純的純線性元件—電阻，來做理論之推導，成效較好，也讓電路變得較為實際可行。本研究即以 Huang & Cheng 之文獻為出發點，探討實作之可行性並實際製作出電路。理論上以「量測」電流代替「計算」車流的方式，應該是較為迅速，因為計算較耗費時間，而量測的資料卻是幾乎可以立即得到。

根據 Huang et al. (2007) 的研究，我們推導出了一些實作時會使用的演算法。實作時需要考慮的因子較多，將會遇到量化誤差、元件誤差、以及元件物理限制等問題，這些誤差本身是沒有辦法克服的，但是其對於本電路實做的影響卻是相當有限，當我們的路網從兩路段擴展到六路段、八路段，解答與理論值之間的誤差仍是在容許的範圍內。最後我們改變起訖點間的流量，研究流量改變對於整個電路的影響，發覺其最後所求出之解與理論值之間差距仍舊不大，這說明了本實作的電路在應用上的確具有可行性。

目前我們已初步證明，利用電路來實做交通指派的問題是可行的，但受限於本身硬體電路的限制，目前整個網路僅能發展到處理 8 路段的交通指派問題。由於本實作電路在進行運算時，將會使用到大量的浮點數運算，浮點數在宣告變數的時候，會耗費到大量的記憶體，而我們所使用的一般型微處理器，受限於其本身之記憶體容量，沒有辦法儲存太多的變數，故沒有辦法繼續將路網擴展。未來如可以改善這一狀況(如更換記憶體更大的微處理器)，應可以此為基礎，繼續做出更大型的路網。

除此之外，我們所使用的元件，如數位可變電阻、類比/數位轉換器等，都是 8 位元共 256 階的電子元件，前面提及這會造成求解上的誤差，未來若採用較數較多的元件，如 10 位元 1024 階之數位電阻，對於整個求解的品質相信會有更好的改善。

參考文獻

- 鄭兆哲，「以電路模擬求解非線性成本之交通指派問題」，交通大學碩士論文，2009。
- 陳惠貞、陳玄玲，新世代計算機概論，學貫行銷股份有限公司。
- 董勝源，MCS-51 與 Keil C 語言入門實習：使用 Keil Vision 3 及 SimLAB 軟體模擬，宏友圖書公司。
- 蔡朝洋、蔡承佑，單晶片微電腦 8051/8951 原理與應用(C 語言)，全華圖書公司。
- 劉銘中、林琮烈，MCS-51 單晶片原理與設計實務：Keil C 語言版，儒林圖書公司。
- 賴麒文，C 與 8051 單晶片設計實務：使用 Keil C，文魁資訊股份有限公司。
- Beckmann, M., McGuire, C. B. and Winston, C. B. (1956) *Studies in the Economics of Transportation*, Yale University Press, New Haven, CT.
- Bell, M.G.H. and Iida, Y. (1997) *Transportation Network Analysis*. John Wiley & Sons, New York, the U.S.A.
- Cho, H., Hwang, M., Huang, H. and Chen, W. (2006) A nonlinear cost function formulation for electrical circuit simulation of traffic assignment, *WSEAS Transactions on Electronics*.
- Huang, K., Cheng, C. and Chuang, T. (2007) Determining the User Equilibrium Flow of the Traffic Assignment Problem by Circuit Simulation, *Proceedings of the 2nd International Conference on Transportation Logistics*, Shenzhen, China.
- Patriksson, M. (1994) *The traffic assignment problem models and methods* (Utrecht, the Netherlands:VSP).
- Patriksson, M. (2004) Algorithms for computing traffic equilibria, *Networks and Spatial Economics*, 4, pp. 23–38.
- Sasaki, T. and Inouye, H. (1974) Traffic assignment by analogy to electric circuit, *Proceeding of the 6th International Symposium on Transportation and Traffic Theory*, pp. 495-515
- Sheffi, Y. (1985) *Urban transportation networks* (Englewood Cliffs, New Jersey, the U.S.A.: Prentice Hall).
- Wang, D. H. and Zhang, H. X. (2005) New traffic assignment method on circuit theory for urban planning, *Journal of Urban Planning and Development*, 131, pp.30-38.
- Wardrop, J. G. (1952) Some theoretical aspects of road traffic research, *Proceedings of the Institute of the Civil Engineers*, Part II, pp. 325-378.

附錄一 微處理器(P89V51RD2)腳位圖及說明

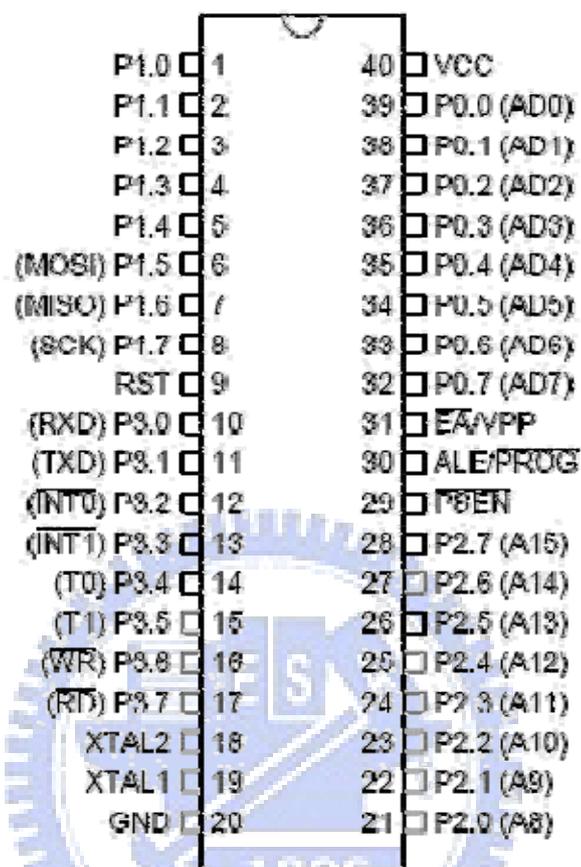
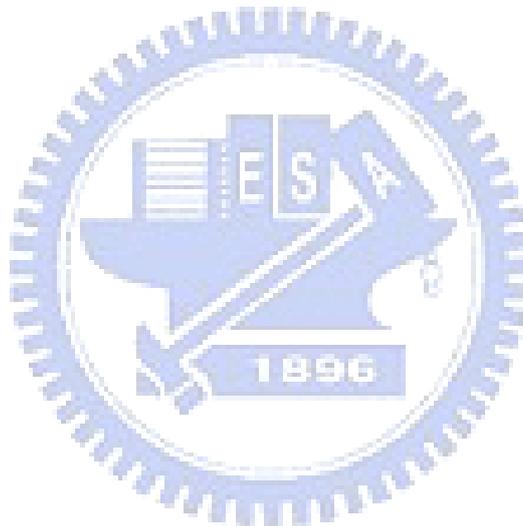


圖 A.1 微處理器 P89V51RD2 頂視圖

表 A.1 P89V51RD2 接腳說明

腳名	腳位	接腳說明
P1.0 / T2	1	IO 埠，T2 為 Timer 2 外部計數輸入(52 以上)
P1.1 / T2EX	2	IO 埠，T2EX 為 Timer 2 外部觸發輸入(52 以上)
P1.2-1.7	3-8	IO 埠
RESET	9	系統重置輸入，RESET=1 會令系統重置
P3.0 / RXD	10	IO 埠，RXD 為串列埠接收
P3.1 / TXD	11	IO 埠，TXD 為串列埠發射
P3.2 / $\overline{INT0}$	12	IO 埠， $\overline{INT0}$ 為外部中斷 0
P3.3 / $\overline{INT1}$	13	IO 埠， $\overline{INT1}$ 為外部中斷 1
P3.4 / T0	14	IO 埠，T0 為 Timer 0 外部計數輸入
P3.5 / T1	15	IO 埠，T1 為 Timer 1 外部計數輸入
P3.6 / \overline{WR}	16	IO 埠， \overline{WR} 為外部資料記憶體寫入控制

P3.7/ \overline{RD}	17	IO 埠， \overline{RD} 為外部資料記憶體讀取控制
XTAL2	18	石英晶體振盪輸出
XTAL1	19	石英晶體振盪輸入
V _{SS}	20	電源接地
P2.0-2.7/A8-15	21-28	IO 埠，A8-15 為外部記憶體位址
PSEN	29	外部擴充 ROM 控制信號
ALE	30	外部位址拴鎖致能，ALE=0，P0 腳為 A0-7 輸出
EA	31	EA=0 不使用內部 ROM，EA=1 使用內部 ROM
P0.0-0.7/AD0-7	39-32	IO 埠，AD0-7 為外部記憶體位址/資料
V _{DD}	40	電源電壓



附錄二 數位電阻(AD8400)腳位圖及說明



圖 B.1 數位電阻 AD8400 頂視圖

表 B.1 AD8400 接腳說明

腳名	腳位	接腳說明
B1	1	電阻端點 B1
GND	2	接地
\overline{CS}	3	晶片選擇輸入，低電位時 IC 啟動
SDI	4	串列資料輸入
CLK	5	串列時脈輸入，正緣觸發
V _{DD}	6	電源電壓
W1	7	Wiper 電阻
A1	8	電阻端點 A1

附錄三 數位/類比轉換器(ADC0804)腳位圖及說明

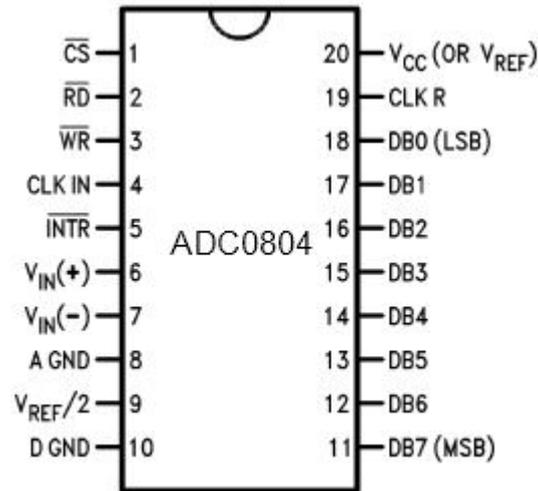


圖 C.1 數位/類比轉換器 ADC0804 頂視圖

表 C.1 ADC0804 接腳說明

腳名	腳位	接腳說明
\overline{CS}	1	晶片選擇輸入，低電位時 IC 啟動
\overline{RD}	2	低電位時，輸出已轉換好之數位資料
\overline{WR}	3	由 0 至 1 時，進行 A/D 轉換
CLK IN	4	時脈輸入
\overline{INT}	5	低電位時，表轉換完成
$V_{IN(+)}$	6	類比電壓輸入
$V_{IN(-)}$	7	類比電壓輸入
A GND	8	接地
$V_{ref}/2$	9	參考電壓
D GND	10	接地
DB7~DB0	11~18	數位輸出
CLK R	19	CLK IN 反相輸入
V_{CC}	20	電源電壓

附錄四 移位暫存器(74164)腳位圖及真值表

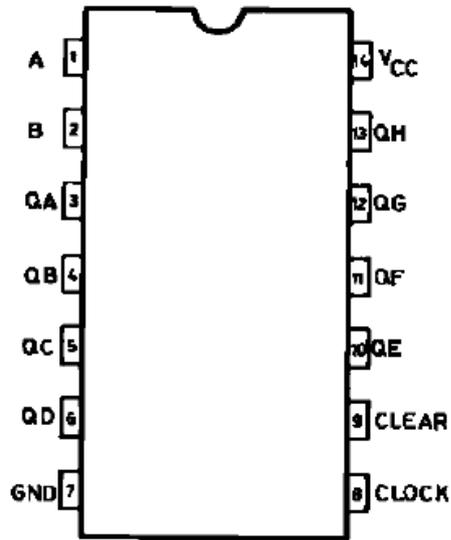


圖 D.1 移位暫存器 74164 頂視圖

表 D.1 移位暫存器 74164-真值表

INPUTS				OUTPUTS			
$\overline{\text{CLEAR}}$	CLOCK	SERIAL IN		QA	QB	QH
		A	B				
L	X	X	X	L	L	L
H		X	X	NO CHANGE			
H		L	X	L	QAn	QGn
H		X	L	L	QAn	QGn
H		H	H	H	QAn	QGn

X: Don't Care