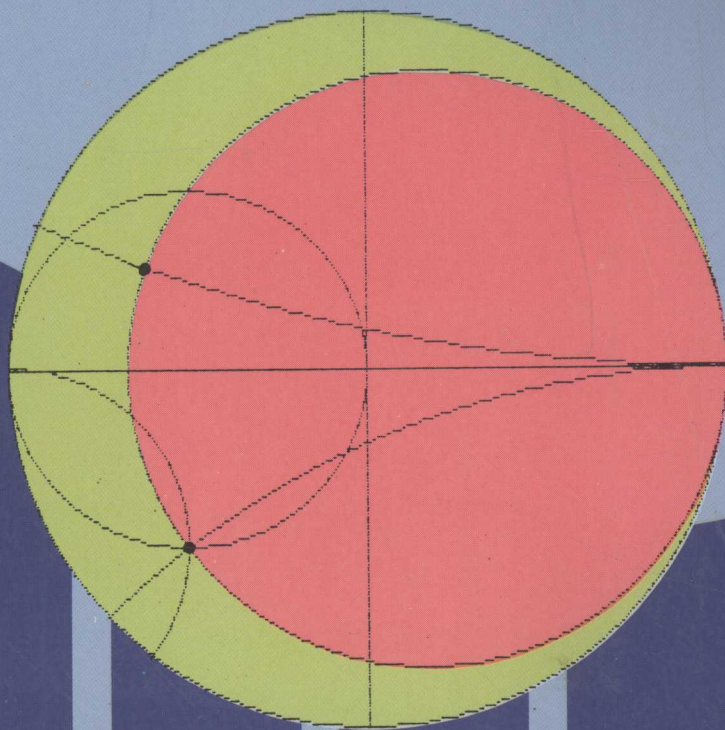


用 C 語言分析 微波電晶體放大器

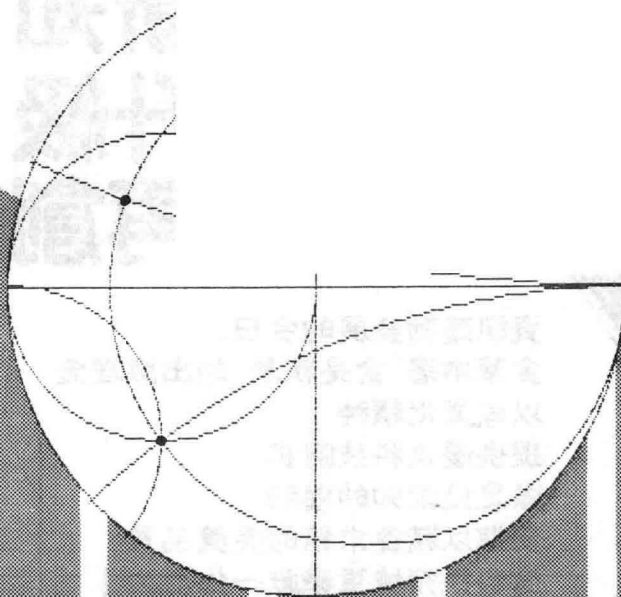
戴奎生 編著



全華科技圖書股份有限公司 印行

用 C 語言分析 微波電晶體放大器

戴奎生 編著



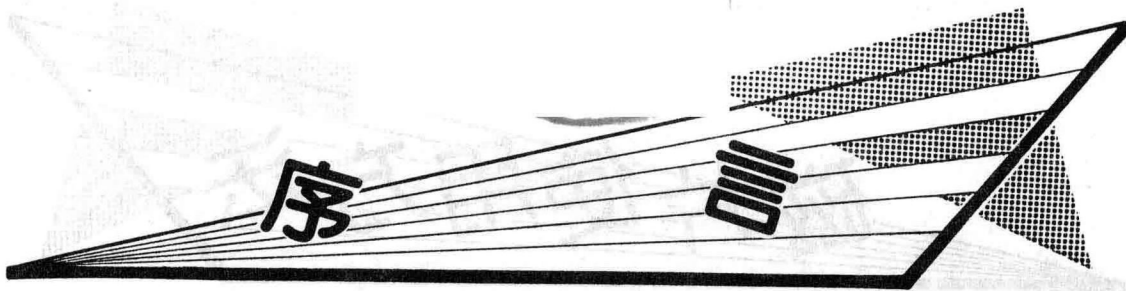
全華科技圖書股份有限公司 印行

我們的宗旨：

**提供技術新知
帶動工業升級
為科技中文化再創新猷**

資訊蓬勃發展的今日，
全華本著「全是精華」的出版理念
以專業化精神
提供優良科技圖書
滿足您求知的權利
更期以精益求精的完美品質
為科技領域更奉獻一份心力！

為保護您的眼睛，本公司特別採用不反光的米色印書紙！！



由於C語言具有強有力的繪圖函式庫 (graphics lib)、繪圖驅動程式工具 (.BGI) 及標準函式庫 (standard library) 等，使得以C語言編寫的應用程式，用以分析以S參數為基礎的微波電晶體電路及討論阻抗匹配問題所使用的史密斯圖 (Smith charts) 等複雜問題，得以簡化而處理迅速。

在我國台灣地區，理工科工程人員的個人電腦擁有率，成長急速，而C語言之用於電子工程界，尤其對高頻電子電路特性的分析、阻抗匹配問題的分析、網路的設計等，亦與日俱增，廣受歡迎。

本書所附原始程式檔 (附加檔名為 .C 者) 為開放式檔案，讀者可依自己設計需求，予以更改使用；應用程式可執行檔 (附加檔名為 .exe 者) 可直接在 DOS 提示訊息下鍵入檔名即可使用，應用方便簡捷，不受工作環境限制，隨時可提供讀者對高頻電路分析及設計的清晰概念。

本書之偏校均於公餘為之，尤其文稿圖表之整理、編寫，得內人芷琴女士之助與全華科技圖書公司的支援，始可如期呈獻於讀者面前，除深致謝意外，尚祈各界先進不吝賜教，尤盼專業軟體人員能鼎力相助，將本書應用程式製成套裝軟體，亦為對電子工程界一大貢獻！

戴奎生 謹識於台北

磁片使用方法

本書含有應用程式磁片兩片；一片為原始程式檔，即附加檔名為 .C 者，
例如：史密斯阻抗——導納圖原始檔：

Smithzy.c

讀者欲自行連結原始檔成為可執行檔，必須編輯該原始檔的專案檔 .prj
方可。編輯專案檔的方法為：

<原始檔名稱> + <GRAPHICS.LIB>

例如，史密斯阻抗——導納圖的專案檔的編輯方法：

Smithzy.c

graphics.lib

取檔名為：smithzy.prj 即可。

另一片為全書應用程式的可執行檔 .exe，該檔可直接在 DOS 提示訊息
下鍵入檔名即可執行，使用極為方便。讀者可依需要選擇其中一片或兩片。

設在 A 磁碟機下，請鍵入：

A><Table 1><return>

螢幕顯示磁片內容如下所示：

1. Herc.bgi:繪圖公用程式
2. Table1:顯示檔案內容
3. Smithzy:史密斯阻抗--導納圖
4. Select2:等阻圖/等抗圖/等導圖/等納圖/反射係數圖
5. Selpac2:並聯C/串聯L---匹配電路
6. Secpal2:並聯L/串聯C---匹配電路
7. Secpal3:並聯L/串聯C---匹配電路($G < 1$)
8. Selpac3:並聯C/串聯L---匹配電路($G < 1$)
9. Secpal5/Selpac5:並聯C/串聯L 或 並聯L/串聯C---匹配電路($G > 1$)
10. RLCmat1:數種典型RLC匹配網路
11. Convert1:四分之一波長線分析
12. Convert2:波長圖與史密斯圖解法
13. Line2/Line4:由輸入反射係數或輸入導納,設計匹配線
14. Line5:由輸入反射係數或輸入導納,微條線物理長度設計
15. Line6:計算微條線的特性阻抗
16. Sparm1:穩定條件及穩定圖
17. Sparm2:單向性功率增益圖
18. Sparm3:改進功率增益的匹配法
19. Sparm4:單向性最大功率增益的匹配法
20. Sparm5:潛在不穩定的匹配法
21. Sparm7:雙向功率增益圖繪圖法
22. Sparm8:由已知S參數決定微波電晶體的各項特性及雜訊指數

例如,若已知微條線材質參數為:

$$\epsilon_r = 2.23$$

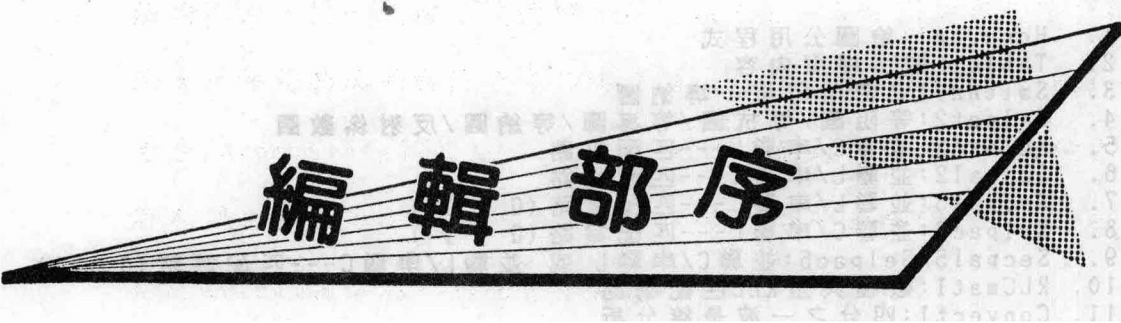
$$h = 0.031 \text{ 吋}$$

$$W = 0.096 \text{ 吋}$$

欲計算該微條線的特性阻抗,依Table1所顯示,可選用可執行檔
Line 6.exe,請在DOS提示訊息下鍵入:

A><Line 6><return>

即可依螢幕顯示鍵入已知條件而獲解答。



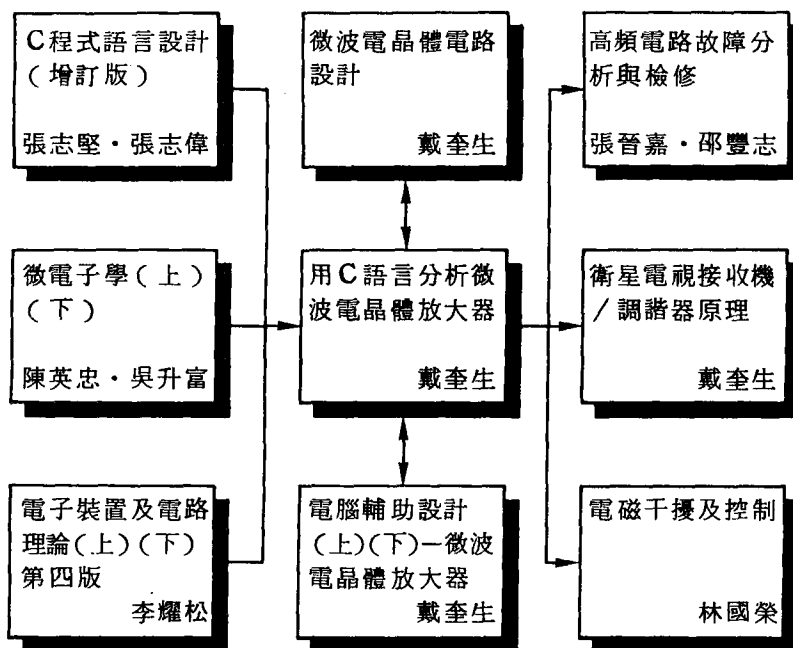
編輯部序

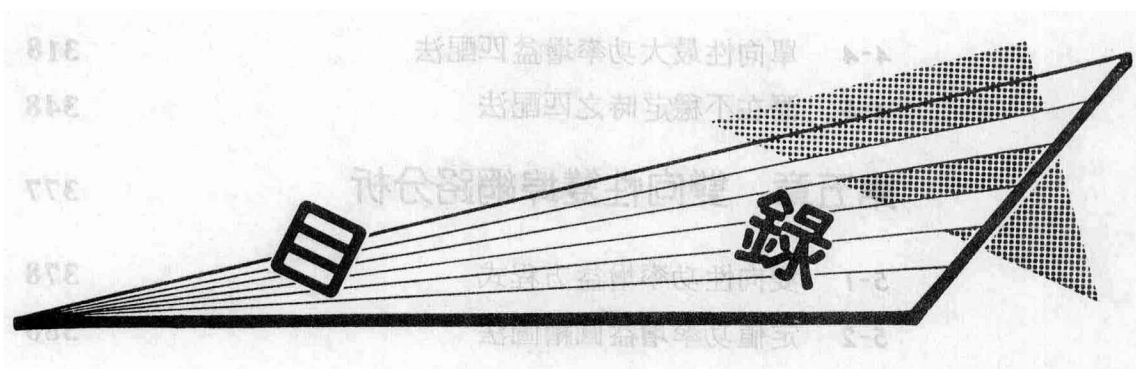
「系統編輯」是我們的編輯方針，我們所提供給您的，絕不只是一本書，而是關於這門學問的所有知識，它們由淺入深，循序漸進。

本書應用C語言分析以S參數為基礎的微波電晶體電路及阻抗匹配問題的史密斯圖，予以分析、歸納、程式化，將深奧理論及複雜運算簡化，循序漸進、逐步探討，配合必要的教學說明及圖表，使讀者瞭解設計原理，而能自行設計高頻電路。適合做為電子專業工程人員工具書及學生自習、業餘者進修的重要參考書。

同時，為了使您能有系統且循序漸進研習相關方面的叢書，我們以流程圖方式，列出各有關圖書的閱讀順序，以減少您研習此門學問的摸索時間，並能對這門學問有完整的知識。若您在這方面有任何問題，歡迎來函連繫，我們將竭誠為您服務。

流程圖





第一章 史密斯圖繪圖法	1
第二章 RLC阻抗匹配網路的設計	35
2-1 串聯電感 - 並聯電容 L 型	35
2-2 串聯電容 - 並聯電感 L 型	51
2-3 輸入為導納式或阻抗式：RLC 匹配網路	70
2-4 當 $G > 1$ 時輸入導納式或阻抗式匹配法	99
2-5 數種典型 RLC 匹配網路設計	131
第三章 輸電線或微條線長度設計	143
3-1 四分之一波長線分析	143
3-2 波長圓與史密斯圖解法	159
3-3 微條線長度設計	174
3-4 應用程式使用說明	194
3-5 微條線的物理長度	222
第四章 應用 S 參數設計電晶體放大器	245
4-1 穩定性及穩定圓的判別	245
4-2 單向性功率及定值增益圓討論	264
4-3 改進功率增益的匹配法	288

4-4	單向性最大功率增益匹配法	318
4-5	潛在不穩定時之匹配法	348
第五章 雙向性雙埠網路分析		377
5-1	雙向性功率增益方程式	378
5-2	定值功率增益圓繪圖法	380
第六章 功率增益及雜訊指數		421
6-1	雙埠網路雜訊分析	421
6-2	定值雜訊指數圓	423
6-3	可用功率增益式及操作功率增益式分析	426
6-4	定值操作功率圓與定值可用功率圓之比較	430
6-5	雜訊指數與功率增益的選擇	436
附錄一：中英文名詞對照表		492
附錄二：繪圖索引		494

史密斯圖繪圖法

應用C語言分析在微波電子領域中比較複雜的輸電線（transmission line）、阻抗匹配（impedance match）及電晶體S參數分析等問題，仍以史密斯圖（smith chart）為最佳途徑。為瞭解本書所用史密斯圖繪圖法，首先對該圖的C語言繪圖法，予以分析。

史密斯圖係由反射係數平面（reflection coefficient plane）及阻抗平面（impedance plane）組成，兩者之間的關係為：

$$P = \frac{Z - 1}{Z + 1} \quad (1-1)$$

式中 Z ：正規化阻抗（normalized impedance）

P ：反射係數（reflection coefficient），亦為一複數

設一阻抗 $Z = R + jX$ ，由式（1-1）表示則為：

$$\begin{aligned} P &= \frac{(R + jX) - 1}{(R + jX) + 1} = \frac{[(R - 1) + jX][(R + 1) - jX]}{[(R + 1) + jX][(R + 1) - jX]} \\ &= \frac{(R^2 + X^2 - 1) + j(2X)}{(R + 1)^2 + X^2} = P_A + jP_B \end{aligned} \quad (1-2)$$

2 用C語言分析微波電晶體放大器

式中
$$P_A = \frac{R^2 + X^2 - 1}{(R+1)^2 + X^2}$$

$$P_B = \frac{2X}{(R+1)^2 + X^2}$$

消去式(1-2)中的X，得：

$$\left(P_A - \frac{R}{1+R}\right)^2 + P_B^2 = \left(\frac{1}{1+R}\right)^2 \quad (1-3)$$

式(1-3)爲一在 P_A 、 P_B 平面內之圓系方程式，以 P_A 爲實軸， P_B 爲虛軸，圓心在 $P_A = \frac{R}{1+R}$ 及 $P_B = 0$ 之處，其半徑爲 $r = \frac{1}{1+R}$ ，R愈大則r愈

Draw Constant Resistance Circles
Resistance Circles : Range from 0 to 2 step 0.1 & 3 to 10 step 1

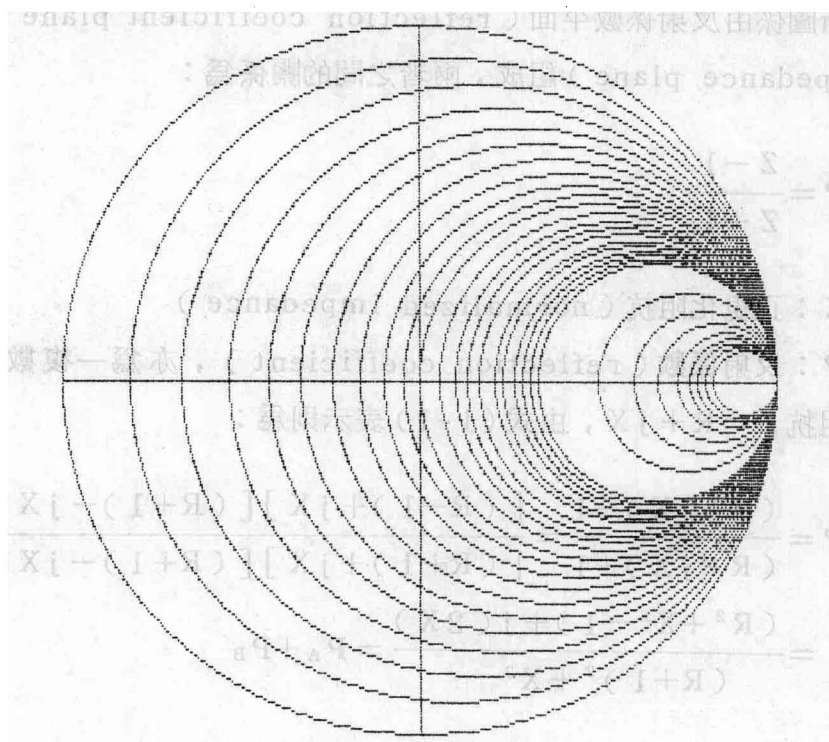


圖 1-1 等阻圓系

小，所繪之圓愈小； R 為零時， $r = 1$ 。式 (1-3) 即為繪史密斯圖中的等阻圓 (constant resistance circles) 公式。設等阻圓半徑為 resist ，自 $\text{resist} = 0$ 到 $\text{resist} = 2.0$ 之間的增量為 0.1 ；自 $\text{resist} = 3$ 到 $\text{resist} = 10$ 之間的增量為 1.0 之等阻圓系如圖 1-1 所示，表 1-1 為繪出該等阻圓系之 C 程式。

表 1-1 resist.c 列印

```

/* Draw a Resistance circles */
/*      resist.c              */
#include <graphics.h>

main()
{
    int graphdriver=DETECT, graphmode;
    float    b,b1,a,resist;
    int      x,y;
    int      xasp,yasp;
    long      xlong;
    double    sin(double),cos(double);

    initgraph(&graphdriver, &graphmode,"");

    line(90,200,500,200);
    line(300,60,300,400);
    circle(300,200,180);

    /* draw Resistance circles */
    outtextxy(160,20,"Drawing Resistance Circles");
    outtextxy(140,30,"Range from 0.0 to 2.0 step
                                0.1 & 3.0 to 10");

    getaspectratio(&xasp,&yasp);
    xlong=(420L*(long)yasp)/(long)xasp;
    rectangle(50,18,(int)xlong,50);

    for (resist=0;resist<=2.1;resist=resist+0.1) {
        a=1+resist;
        b1=1/a;
        b=b1*180;
        y=200;
        x=480-b;
        circle(x, y, b);
    };
    for (resist=3;resist<=10;resist=resist+1){
        a=1+resist;
        b1=1/a;
        b=b1*180;
        y=200;
        x=480-b;
        circle(x,y,b);
    };
}

```

4 用C語言分析微波電晶體放大器

表 1-1 (續)

```
getch();  
closegraph();  
return(0);  
}
```

設等導圓半徑為 conduct，自 conduct = 0 到 conduct = 2.0，增量為 0.1 及 conduct = 3.0 到 conduct = 10.0，增量為 1 之等電導圓系，如圖 1-2 所示，表 1-2 為繪出等電導圓系之 C 程式。

消去式 (1-2) 中的 R，可得：

$$(P_A - 1)^2 + \left(P_B - \frac{1}{X}\right)^2 = \left(\frac{1}{X}\right)^2 \quad (1-4)$$

Draw Constant Conductance Circles
Conductance Circles : Range from 0 to 2 step 0.1 & 3 to 10 step 1

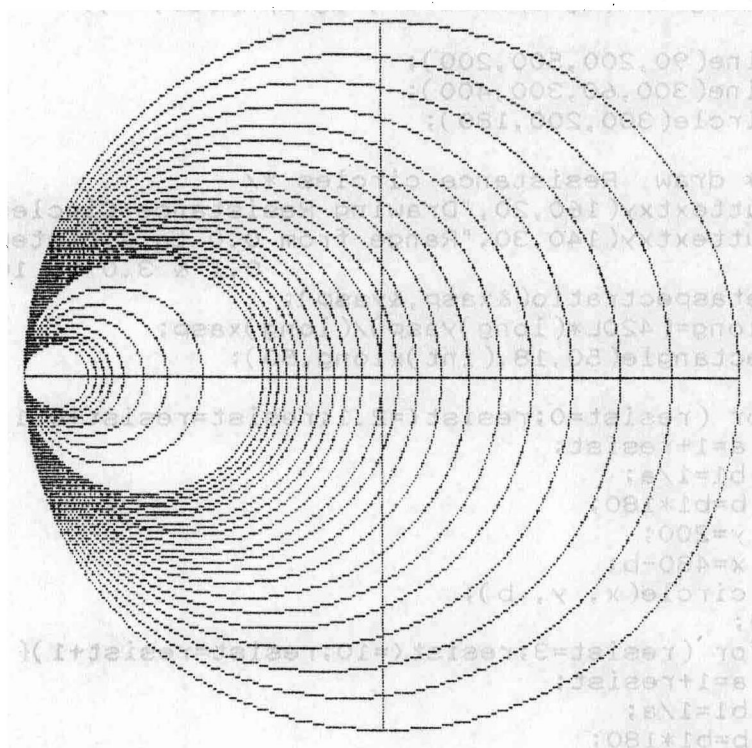


圖 1-2 等電導圓系

表 1-2 列印 conduct.c

```

/* Draw a conductance circles */
/*      conduct.c                      */

#include <graphics.h>

main( )
{
    int graphdriver=DETECT, graphmode;
    float  b,b1,a,conduct;
    int     x,y;
    int     xasp,yasp;
    long     xlong;
    double   sin(double),cos(double);

    initgraph(&graphdriver, &graphmode,"");

    line(90,200,500,200);
    line(300,60,300,400);
    circle(300,200,180);

    /* draw conductance circles */
    outtextxy(160,20,"Drawing Conductance circles");
    outtextxy(140,30,"Range from 0.0 to 2.0 step
                                0.1 & 3.0 to 10");
    getaspectratio(&xasp,&yasp);
    xlong=(420L*(long)yasp)/(long)xasp;
    rectangle(50,18,(int)xlong,50);

    for (conduct=0;conduct<=2.1;conduct=conduct+0.1) {
        a=1+conduct;
        b1=1/a;
        b=b1*180;
        y=200;
        x=120+b;

        circle(x, y, b);
    };
    for (conduct=3;conduct<=10;conduct=conduct+1){
        a=1+conduct;
        b1=1/a;
        b=b1*180;
        y=200;
        x=120+b;
        circle(x,y,b);
    };

    getch( );
    closegraph( );
    return(0);
}

```

6 用C語言分析微波電晶體放大器

式(1-4)爲一在 P_A 、 P_B 平面內之圓系方程式，圓心在 $P_A = 1$ 及 $P_B = \frac{1}{X}$ 處，以不同的 X 值代入，即可繪出大小不同之圓，此即爲繪史密斯圖中的等電抗圓(constant reactance circles)或等電納圓系(constant susceptance circles)的公式。

設等電抗圓半徑爲 react ，自 $\text{react} = 0$ 到 $\text{react} = 2.0$ ，增量爲 0.1 及 $\text{react} = 3.0$ 到 $\text{react} = 10.0$ ，增量爲 1.0 之等電抗圓系如圖1-3所示，表1-3爲繪出該圓系之C程式。

Draw Constant Reactance Circles
Reactance Circles : Range from 0 to 2 step 0.1 & 3 to 10 step 1

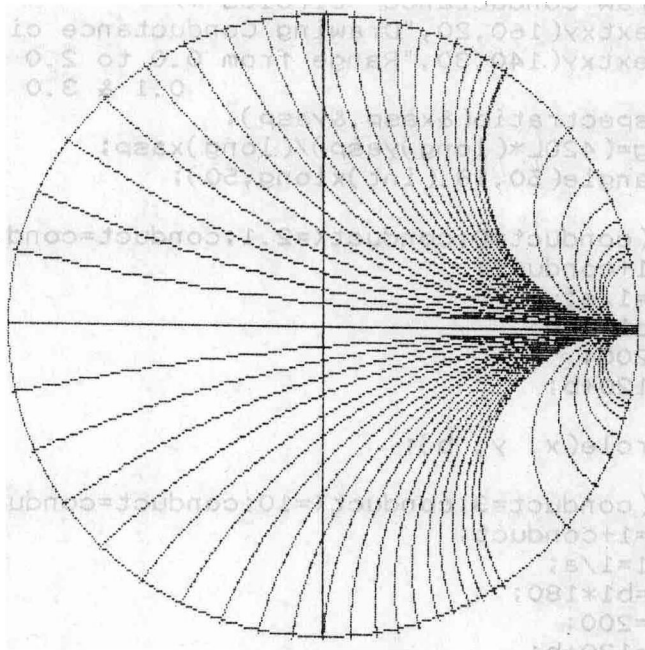


圖 1-3 等電抗圓系

表 1-3 列印 react.c

```

/* Draw reactance circles */
/*      react.c      */

#include <graphics.h>

main( )
{
    int graphdriver=DETECT, graphmode;
    float    b,b1,a,react;
    int      x,y;
    int      xasp,yasp;
    long     xlong;
    double    sin(double),cos(double),atan(double);

    struct    viewporttype;
    int       left,top,right,bottom;
    int       clip;
    initgraph(&graphdriver, &graphmode,"");

    line(140,200,460,200);
    line(300,80,300,320);
    circle(300,200,160);

    /* draw Reactance circles */
    outtextxy(180,2,"Drawing Reactancee Circles");
    outtextxy(150,12,"Range from 0.0 to 2.0 step
                                0.1 & 3.0 to 10.0");
    getaspectratio(&xasp,&yasp);
    xlong=( 420L*( long)yasp)/( long)xasp;
    rectangle( 50,0,( int)xlong,30);
    setviewport(140,80,460,320,1);

    for (react=0.1;react<=2.1;react=react+0.1) {
        b1=1/react;
        b=b1*160;
        a=atan(160/b)*3.14159265/180;
        y=120-b*cos(a)*0.75;
        x=320;

        arc(x, y, 135,290,b);
    };
    for (react=3;react<=10;react=react+1){
        b1=1/react;
        b=b1*160;
        a=atan(160/b)*3.14159265/180;
        y=120-b*cos(a)*0.75;
        x=320;
        arc(x,y,110,250,b);
    };
}

```