Device Outline Optimization for PCB Design

I. Introduction

電子產品設計程序中電路板設計階段需依零件外型進行限制區域之設計，電子零件限制區域主要用來檢查電路板上零件匹配的問題，限制區域過小將影響零件插件，但因受電路板空間限制，其區域亦不可過大，所以限制區域的設計需根據零件外型設計一個最佳的範圍。

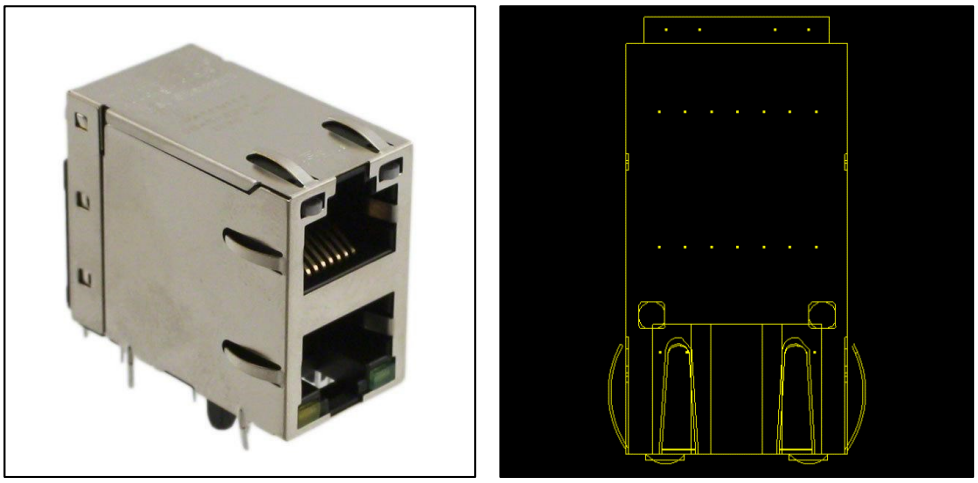


圖 1 3D 外型投影至 2D 畫布示意圖

一般零件 3D 外型投影至 2D 畫布後所呈現的資料為多組無法輕易辨識的線段資料，其中可能包含線段不連續、重疊、偏移等路徑失真的情形，故設計限制區域前須進行元件外廓最佳化，使資料得以明確描述一個封閉的零件外觀，再根據此外型進行演算（**圖 2**）得到最佳的限制區域範圍。

在 EDA 工具中 2D 畫布普遍所支援的繪製精確度有限（通常為小數點後四位），導致基於畫布資料的圖形演算功能均受到此畫布的精確度限制，而這將導致圖形演算結果無法準確的被建立於 2D 畫布中，進而導致結果無法符合繪製需求。（**圖 3**）

此類型的圖形演算技術可廣泛運用於電子零件限制區域最佳化、自動化製圖等各種基於圖形關係之需求，藉此提升電子產品設計於前期元件資料建構之準確性、穩定性、效率等。

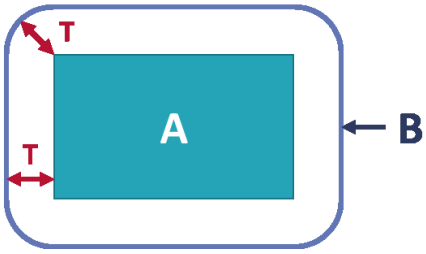


圖 2 假定於限制區域設計時，需要讓所有物件距離 A 範圍皆保持 T 值，可使用幾何等距離放大處理此需求，即 B 範圍。

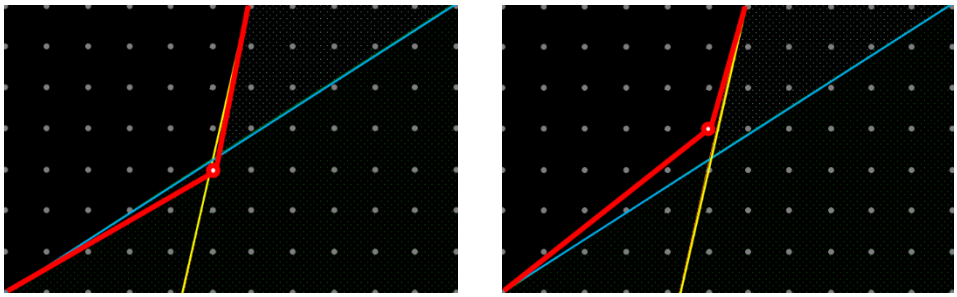
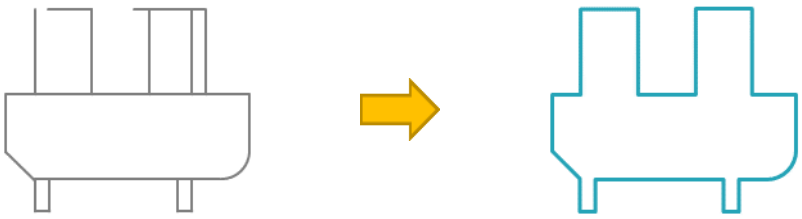


圖 3 圖中灰色點表示畫布最小精確度之繪圖限制，即描述線段之所有資料點（起點終點或是弧線圓心）皆需於灰色點上。舉例由黃色線段與藍色線段計算交點，因 EDA 工具中 2D 畫布精確度限制，導致無法透過數值處理取得演算結果的資料點，而須透過幾何關係進行判斷，例如當演算交點之目的為取得可包覆兩圖形之範圍時，資料點應取用位於兩集合之外且距離交點最近者，如右圖。

II. Problem Description

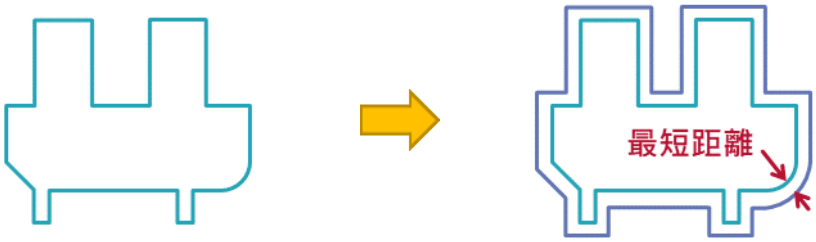
i. 元件外廓最佳化

輸入之路徑資料包含許多線段，其線段因精度轉換失真以致其無法成為一封閉路徑，所以於該步驟需進行線段補償使其成為一封閉路徑，而單一補償線段長度不可大於 2；此外，輸入路徑中亦包含描述零件細節之線段，於此步驟須將其移除，而移除資料線段長度不可大於 2。



ii. **圖形等距擴大**

計算一封閉路徑，使其與(i)所得之結果皆保持最短距離大於等於指定數值。



iii. **凹槽填補**

由(ii)所得結果針對其凹槽部分進行填補，使其路徑之凹槽間隙皆大於指定數值。

III. Example of Input / Output Files

i. **Format**

- Operation, 問題目錄 (指此檔案將敘述的資料標籤)；
- Data, 資料開始 ex. Data,Q1;表示 Q1 資料開始;
- End, 資料結束 ex. End,Q1;表示 Q2 資料結束;
- Expand, 等距放大尺寸;
- NotchSize, 凹槽間隙;
- Line, 線段起點(x), 線段起點(y), 線段終點(x), 線段終點(y);
- Arc, 線段起點(x), 線段起點(y), 線段終點(x), 線段終點(y), 弧線圓心(x), 弧線圓心(y), 弧線半徑(r);

ii. **Input**

Operation,Q1,Q2;

Data,Q1;

Expand,1;

NotchSize,3;
Line,13,-6,0,-6;
Line,0,-6,0,-8;
Line,0,-8,1,-8;
Line,1,-8,1,-6;
Line,4,6,4,0;
Line,4,0,8,0;
Line,8,0,8,6;
Arc,15,-4,13,-6,13,-4,CW;
Line,-2,0,-2,-4;
Line,-2,-4,0,-6;
Line,-2,0,12,0;
Line,0,6,0,0;
Line,4,6,1,6;
Line,11,-6,11,-8;
Line,11,-8,10,-8;
Line,10,-8,10,-6;
Line,15,-4,15,0;
Line,15,0,12,0;
Line,12,0,12,6;
Line,12,6,8,6;
Line,11,6,11,0;
Line,8,6,7,6;
End,Q1;

Data,Q2;
Expand,0.5;
NotchSize,2;
Arc,-9,-9,-9,-7,-9,-8,CW;
Arc,-9,-1,-9,1,-9,0,CW;
Line,5,-11,-5,-11;
Line,-5,-11,-5,11;
Line,-3,8,-3,-9;
Line,-3,-9,3,-9;
Line,-6,-7,-6,-9;
Line,-6,-9,-9,-9;
Line,-9,-7,-6,-7;
Line,-9,-5,-9,-3;

Line,-9,-1,-9,1;

Line,-8,-1,-8,1;

Arc,-9,-5,-9,-3,-9,-4,CW;

Line,-8,-5,-8,-3;

Line,-6,1,-6,-1;

Line,-6,-1,-9,-1;

Line,-9,1,-6,1;

Line,-6,-3,-6,-5;

Line,-6,-5,-9,-5;

Line,-9,-3,-6,-3;

Line,-9,3,-9,5;

Line,-9,7,-9,9;

Line,-8,7,-8,9;

Arc,-9,7,-9,9,-9,8,CW;

Line,-8,3,-8,5;

Line,3,8,-3,8;

Line,-6,9,-6,7;

Line,-6,7,-9,7;

Line,-9,9,-6,9;

Line,-6,5,-6,3;

Line,-6,3,-9,3;

Line,-9,5,-6,5;

Line,-5,11,-6,12;

Line,-6,12,6,12;

Arc,-9,3,-9,5,-9,4,CW;

Line,-5,11,-2,11;

Line,6,-12,5,-11;

Line,5,11,5,-11;

Line,3,-9,3,8;

Line,1,4,-1,4;

Arc,1,2,-1,2,0,2,CCW;

Arc,-2,11,2,11,0,11,CCW;

Line,5,11,6,12;

Line,2,11,5,11;

Line,9,-9,9,-7;

Line,-5,-11,-6,-12;

Line,8,-9,8,-7;

Arc,9,-7,9,-9,9,-8,CW;

Line,9,-9,6,-9;

Line,6,-9,6,-7;

Line,6,-7,9,-7;

Line,6,12,6,-12;

Line,9,-1,9,1;

Line,9,-5,9,-3;

Line,8,-5,8,-3;

Line,8,-1,8,1;

Line,-9,-9,-9,-7;

Arc,9,1,9,-1,9,0,CW;

Line,9,-1,6,-1;

Line,6,-1,6,1;

Line,6,1,9,1;

Arc,9,-3,9,-5,9,-4,CW;

Line,9,-5,6,-5;

Line,6,-5,6,-3;

Line,6,-3,9,-3;

Line,9,7,9,9;

Line,9,3,9,5;

Line,-8,-9,-8,-7;

Line,8,3,8,5;

Line,8,7,8,9;

Arc,9,9,9,7,9,8,CW;

Line,9,7,6,7;

Line,6,7,6,9;

Line,6,9,9,9;

Arc,9,5,9,3,9,4,CW;

Line,9,3,6,3;

Line,6,3,6,5;

Line,6,5,9,5;

Line,6,-12,-6,-12;

Line,-6,-12,-6,12;

End,Q2;

iii. Output

Operation,Q1,Q2;

Data,Q1;

Line,12,-7,12,-8;
Arc,12,-8,11,-9,11,-8,CW;
Line,11,-9,10,-9;
Arc,10,-9,9,-8,10,-8,CW;
Line,9,-8,9,-7;
Line,9,-7,2,-7;
Line,2,-7,2,-8;
Arc,2,-8,1,-9,1,-8,CW;
Line,1,-9,0,-9;
Arc,0,-9,-1,-8,0,-8,CW;
Line,-1,-8,-1,-6.4142;
Line,-1,-6.4142,-2.7071,-4.7071;
Arc,-2.7071,-4.7071,-3,-4,-2,-4,CW;
Line,-3,-4,-3,0;
Arc,-3,0,-2,1,-2,0,CW;
Line,-2,1,-1,1;
Line,-1,1,-1,6;
Arc,-1,6,0,7,0,6,CW;
Line,0,7,12,7;
Arc,12,7,13,6,12,6,CW;
Line,13,6,13,1;
Line,13,1,15,1;
Arc,15,1,16,0,15,0,CW;
Line,16,0,16,-4;
Arc,16,-4,13,-7,13,-4,CW;
Line,13,-7,12,-7;
End,Q1;

Data,Q2;
Line,6.5,-9.5,6.5,-12;
Arc,6.5,-12,6,-12.5,6,-12,CW;
Line,6,-12.5,-6,-12.5;
Arc,-6,-12.5,-6.5,-12,-6,-12,CW;
Line,-6.5,-12,-6.5,-9.5;
Line,-6.5,-9.5,-9,-9.5;
Arc,-9,-9.5,-10.5,-8,-9,-8,CW;
Line,-10.5,-8,-10.5,8;
Arc,-10.5,8,-9,9.5,-9,8,CW;

Line,-9,9.5,-6.5,9.5;

Line,-6.5,9.5,-6.5,12;

Arc,-6.5,12,-6,12.5,-6,12,CW;

Line,-6,12.5,6,12.5;

Arc,6,12.5,6.5,12,6,12,CW;

Line,6.5,12,6.5,9.5;

Line,6.5,9.5,9,9.5;

Arc,9,9.5,10.5,8,9,8,CW;

Line,10.5,8,10.5,-8;

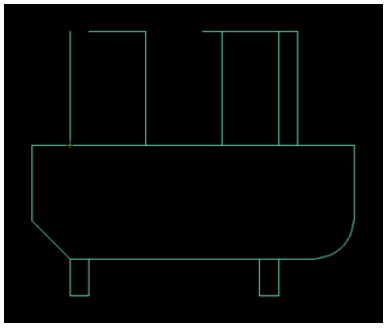
Arc,10.5,-8,9,-9.5,9,-8,CW;

Line,9,-9.5,6.5,-9.5;

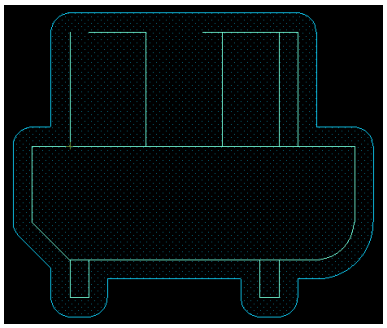
End,Q2;

iv. Reference Diagram

- **Q1**

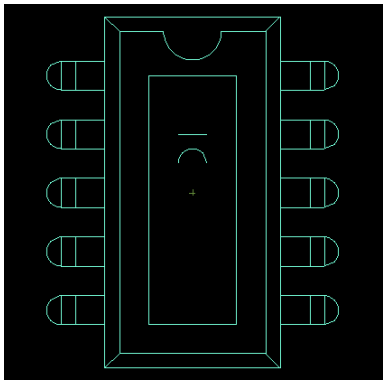


Input Diagram

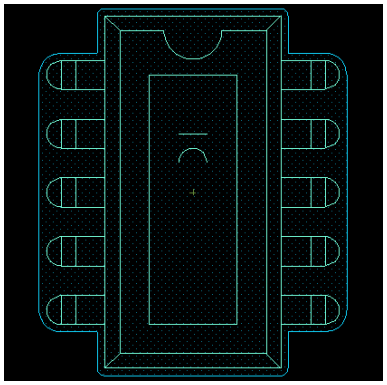


Output Diagram
(藍色路徑)

- **Q2**



Input Diagram



Output Diagram
(藍色路徑)

v. Note

- 所有資料之數值精確度為小數點後四位為有效(即 1.1234)。

IV. Platform

- OS：Linux
- Compiler：gcc / g++

V. Testcases

- 3 Public case (不採計分數)
- 5 Hidden case

VI. Evaluation

i. 符合問題基本要求，若無法符合則不予計分

- 是否為唯一且封閉的路徑
- 所有路徑線段皆與原始資料之最短距離大於等於指定數值(因元件外廓最佳化移除之線段不在此限)
- 凹槽填補區域是否正確

ii. 圖形覆蓋面積較小者為佳

- $Area$ = 指定矩形範圍面積（100×100）
- 圖形覆蓋面積分數 = $\frac{Area - \text{Output 之多邊形面積}}{Area} \times 100\%$

iii. 若圖形覆蓋面積分數相同者，則以程式執行效能分數較佳者為優

- $Time = 300000(\text{ms})$
- 執行效能分數 = $\frac{Time - \text{執行時間(ms)}}{Time} \times 100\%$