

19. 系統模組建構器

杜日富 副教授
聖約翰科技大學 電子工程系

摘要

本文提出了一重新整合的模糊理論結合派翠網的定義模型演算法，此定義模型是真正將派翠網結合L.A.Zadeh所提出的模糊推論法，而使此模糊派翠網提升到真正具有詮釋模擬模糊性控制系統的推論定義模型。

另一方面，藉由此清楚且演算簡單的定義模型，開發出一套具有此定義模型的模擬系統軟體，藉由此一模擬系統可快速幫助設計者完成系統設計前的先前分析工作，以達到增加設計者設計的系統穩定性及正確性的成功率，最重要是借由此模擬系統的完成，來驗證本定義模型是可行的且正確的。

關鍵字：派翠網，模糊理論

1. 前言

模糊理論發展至今，其中又以模糊控制（Fuzzy control）為模糊理論最重要的應用重點項目，而模糊控制是結合了專家經驗與語意式的控制法則，使具有某些仿人工智慧的工程控制與資訊處理系統之功能，完全可不用依賴受控對象的數學模型，就可介由人的操作經驗、知識和受控系統的某些性能而對應出系統所需的控制動作，S.M.Chen[1]等人提出之模糊派翠網定義模型，但所使用的方法都並非是真正結合L.A.Zadeh[9]所提出的模糊推理。而本研究中提出了一重新整合的模糊理論結合派翠網的定義模型，此定義模型是真正將派翠網結合L.A.Zadeh所提出的模糊推理法，而使此模糊派翠網提升到真正具有詮釋模擬模糊性控制系統的推論定義模型[4,5,6,7,8]。

另一方面，藉由此清楚且演算簡單的定義模型，開發出一套具有此定義模型的模擬系統軟體，藉由此一模擬系統可快速幫助設計者完成系統設計前的先前分析工作，以達到增加設計者設計的系統穩定性及正確性的成功率，最重要一點是借由此模擬系統的完成，來驗證本定義模型是可行的且正確的。

2. 模糊理論結合派翠網之定義模型

在本研究中提出了一重新整合的模糊推論嵌入於派翠網的定義模型，此定義模型除了具有派翠網的一些基本運算及特性外，此定義模型還結合了L.A.Zadeh所提出的模糊推理方式，進而達成具有模擬模糊控制系統推論的一圖形化演示的定義模型，並且進階的利用了模糊推理演算法則，修改出了不同型式且更有效於原派翠網的狀態觸發方法及觸發特性。

而本研究中的定義模型之基本架構可分為三個不同的階段型態介面，每一個階段型態的位態（Place）、態變（Transition）、標記（Token）與有向性弧線（Directed Arc）各有不同的規則定義，其中包含有第一階段的模糊化介面區塊、第二階段的模糊法則式推論介面以及第三階段的解模糊化介面區塊，此定義模型的基本架構圖，如圖1所示[2, 3]。

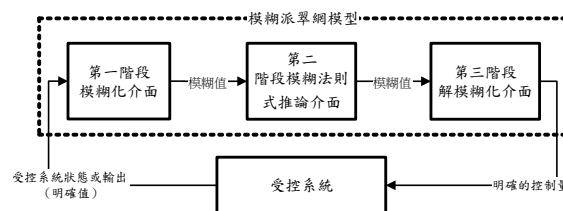


圖1 模糊推論嵌入於派翠網之定義模型的基本架構圖

而此定義模型的三個階段介面區塊的功能分別如下：

1. 第一階段的型態介面 - 模糊化介面；主要功能為將系統輸入之明確值對應到設計者所定義之模糊集合上，因而產生此系統明確值之模糊歸屬函數。相當於模糊控制系統中的模糊化區塊與模糊知識庫區塊中此階段集合相關的模糊歸屬函數資料。
2. 第二階段的型態介面 - 模糊法則式推論介面；主要功能為將模糊化介面所推論出之模糊值，根據設計者所在此區塊描述之規則式，進行模糊規則推論。相當於模糊控制系統中的模糊決策邏輯區塊與模糊知識庫區塊中此階段相關的控制規則資料。
3. 第三階段的型態介面 - 解模糊化介面；主要功能為將模糊法則式推論區塊所推論之模糊值，對應到設計者所定義之模糊集合上，經由解模糊後，因而產生此系統之明確的控制量。相當於模糊控制系統中的解模糊化區塊與模糊知識庫區塊中此階段集合相關的模糊歸屬函數資料。

由上發現，本研究已將模糊控制系統中各功能區塊映射到本研究定義的模糊派翠網模型基本架構區塊上，以下再進一步說明本研究定義的模糊派翠網模型各階段型態的區塊內部之位態、態變、標記及有向連接線的定義為何，如下敘述：

3. 模糊化介面

1. 位態：是用來代表系統所要分析的輸入訊號變數的位置。
2. 態變：是代表將變數對應出在各模糊集合所產生的模糊值的事件激發。
3. 標記：是代表該系統所要分析的輸入訊號變數的位置，已有訊號變數值產生。

4. 有向性弧線：連結位態節點和態變節點間的流向關連（Flow Relation）。

本區塊的定義模型如圖2。

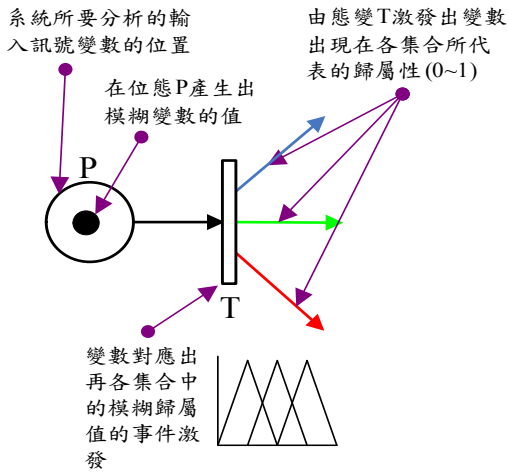


圖2 模糊化介面的位態、態變、標記以及有向性弧線的意義

4. 模糊法則式推論介面

1. 位態：是代表系統各個變數所產生各種集合的位置，其中此區塊的位態又可分為兩種類型的狀態，前提位態以及結論位態二種。
2. 態變：是代表將匯集在態變上的各個集合歸屬值做取小值的事件激發，並且會檢驗此態變上所連接的結論位態有無其他來源路徑連結，若有，檢驗其他同時觸發的其他態變比較後的值，是否此態變上的值是其其他來源路徑中最大值，若是此時其態變才具有真實觸發將值送往結論位態的能力。
3. 標記：是代表系統訊號變數的該模糊集合的位置，已有模糊值產生。
4. 有向性弧線：連結位態節點和態變節點間的流向關連。
5. 截值 (α)：代表有向性弧線上所要准許通過的模糊值範圍。
6. 語氣算子 (h)：代表位態集合上的模糊值的次方值。

本區塊的定義模型如圖3：

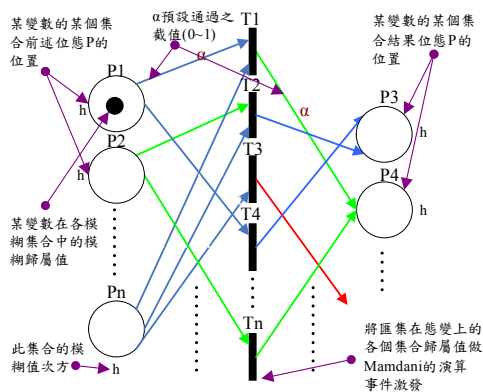
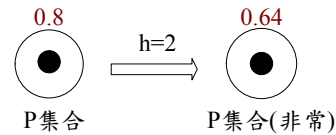


圖3 模糊法則式推論介面的位態、態變、標記、 α 、 h 以及有向性弧線的意義

另外在此階段的型態介面中的位態 (Place)，也可利用位態上的語氣算子 h 來調整出原有的位態集合相對應的加強語氣，語氣算子 h 代表位態集合上的模糊值的次方值，例如：前部分的模糊變數在 P 集合內的歸屬值為 0.8，若這時需要另一種 P 集合的加強語氣，故可使用位態上的語氣算子 h 來調整出位態上集合對應的語氣，而使前部分的模糊變數在新產生的 P 集合(非常)上的歸屬值為 0.64 以供使用，如圖 4。



將 P 集合內的歸屬值加強語氣，對應出新的加強語氣的 P 集合，以供使用(用 h 表示模糊歸屬值的次方，極 4，非常 2，相當 1.25，比較 0.75，有點 0.5，稍微有點 0.25，不變 1)

圖4 模糊法則式推論介面的位態語氣算子的意義

而在此階段的型態介面是可用來描述法則推論的方式，本模型是以 Mamdani 規則推論法做推論，舉例說明若有以下條件規則式：

- R1：IF P_1 AND P_3 THEN P_5
- R2：IF P_1 AND P_4 THEN P_5
- R3：IF P_2 AND P_4 THEN P_6
- R4：IF P_2 AND P_3 THEN P_6

R1~R4 條件規則式用本區塊介面的定義可描述成如圖 5。

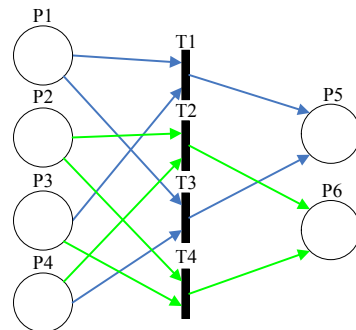


圖5 R1~R4 條件規則式用本區塊介面的定義描述示意圖

其中 P_1 、 P_2 、 P_3 、 P_4 是屬於前題位態的部分， P_5 、 P_6 是屬於結論位態的部分，而在此各模糊集合代表的前題位態 P_1 、 P_2 、 P_3 、 P_4 相互 AND 的運算，本模型是採用 Mamdani 的邏輯積演算 (min 運算法)，接著由於規則式的最後集結方式是屬於 OR 的運算，故規則式集成在屬於結論位態 P_5 、 P_6 上的運算，本模型是採用 Mamdani 的邏輯和演算 (max 運算法)，假設此模型各有向性弧線上的 $\alpha=0$ 和各模糊集合上的 $h=1$ ($\alpha=0$ 表示無條件通過任何歸屬值， $h=1$ 表示模糊集合加強語氣不變) 以及各個前題位態出現的模糊歸屬值分別為 $P_1=0.8$ 、 $P_2=0.4$ 、 $P_3=0.3$ 、 $P_4=0.6$ 。之後經由各態變的 Mamdani 的邏輯積演算 (min 運算法)，可在各態變上產生出各值為 $T_1=0.3$ 、 $T_2=0.4$ 、 $T_3=0.6$ 、 $T_4=0.3$ ，此時規則式的最後集結的結論位態 P_5 、 P_6 ，再經由 Mamdani 的邏輯和演算 (max 運算法)，便可產生出最後的結果 $P_5=0.6$ 、 $P_6=0.4$ ，另一方面，經由此 Mamdani 的演算方法，便

可用來有效的判斷出正確 P_5 、 P_6 的態變觸發路徑分別為 T_3 、 T_2 ，故使具有了態變預測的運算能力，上述運算如圖6、圖7所示。

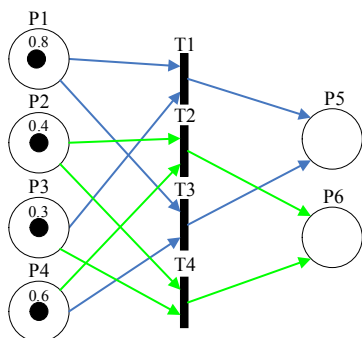


圖6位態的Token為 $P_1=0.8, P_2=0.4, P_3=0.3, P_4=0.6$

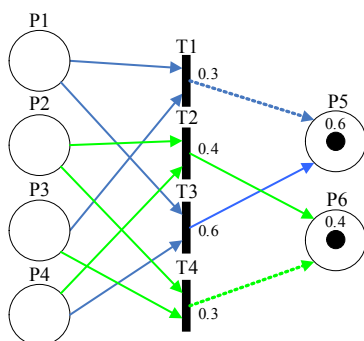


圖7 經由Mamdani規則推論法之結果

解模糊化介面

1. 位態：是代表系統所要分析的輸出訊號變數的位置。
2. 態變：是代表將匯集在態變上的各個集合歸屬值做解模糊（所使用的解模糊方法是用重心解模糊法 [Center of sum defuzzification]），對應出在各模糊集合變數的事件激發。
3. 有向性弧線：連結位態節點和態變節點間的流向關連。
4. 標記：是代表系統所要分析的輸出訊號變數位置，已有輸出訊號變數值產生。

本區塊的定義模型如圖8。

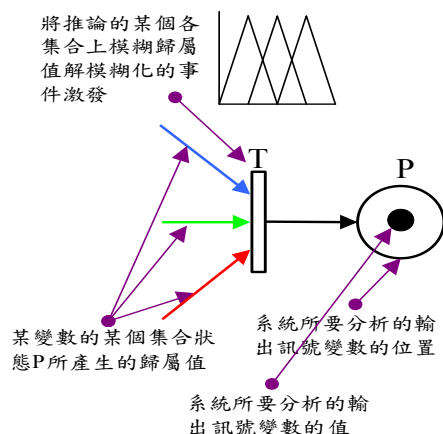


圖8 解模糊化介面的位態、態變、標記以及有向性弧線的意義

5. 模擬系統開發

在現今模糊控制系統的相關軟體分析工具上，也已有公司發展出了相關的模擬軟體，例如有Inform SOFTWARE公司所設計的- fuzzyTECH、Mathworks公司所設計的- Matlab、智控科技公司所設計的- FCDE 等，但這些分析軟體在模糊條件規則式的設立上，是使用了以程式撰寫的方式讓使用者輸入所想建立的模糊條件規則式，故繼承了傳統模糊控制理論的缺點，在規則的分析上欠缺了讓人容易了解的圖形化示意規則以及讓設計者無法在第一時間清楚的了解模糊規則式的推演狀況，也間接的造成了設計者在除錯及修改上許多的困難及困擾。

而本研究利用所提出的定義模型，開發出了一套具有演算及模擬模糊控制系統的分析系統軟體，以提供一套不同型態且有效解決上述軟體缺點的系統分析工具；另一方面，藉由此模擬系統的完成，得以驗證本研究所提出之定義模型是正確且可行的。

6. 模擬系統的開發工具

在模擬系統的開發工具方面，由於現今大多數的個人電腦都是使用微軟公司所開發的作業系統軟體（Microsoft Windows），故為了將來完成後的模擬系統之相容性，所以本研究模擬系統的開發工具，選用的也是由微軟所開發的 Virtual Basic (VB) 程式開發工具，利用VB所提供的良好使用者圖形介面及物件導向程式設計的功能，來建立各項操作的圖件及模擬系統所需的演算功能演示等。

7. 模擬系統的程式架構

本研究的整個模擬系統的程式架構，大至將之分為五大重點程式區塊來組成，分別為以下區塊：1.系統物件圖形與路徑表程式區塊、2.系統模糊相關設定程式區塊、3.系統路徑表判別程式區塊、4.系統模糊推理演示及判別與演算程式區塊、5.人機介面程式區塊，如圖9所示：

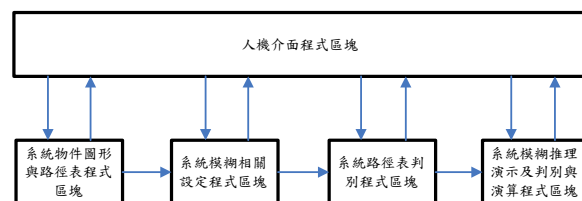


圖9 模擬系統整體的程式架構

而在各程式區塊的大至功能如下敘述：

1. 系統物件圖形與路徑表程式區塊
可利用圖形化動態的方式建立位態、態變、標記、有向性弧線以及語氣因子與截值的設定，以建立整個模糊控制系統的雛型架構及模糊規則式。
2. 系統模糊相關設定程式區塊
可設定整體系統所需的模糊變數以及模糊歸屬函數。
3. 系統路徑表判別程式區塊
經由系統的整體路徑表，可自動演算出整體系統的相關矩陣及數學式。
4. 系統模糊推理演示及判別與演算程式區塊
可動態演示及演算整體系統的推論情形及結果的呈現。

5. 人機介面程式區塊

提供設計者與模擬系統的操作面板介面。

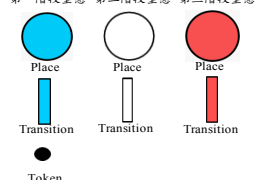
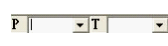
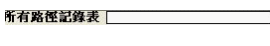
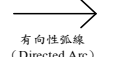
8. 程式區塊運用的撰寫概略說明

在此章節說明各程式區塊所使用的VB方法，並且概略說明區塊中幾個重要的程式流程圖以及副程式：

系統物件圖形與路徑表程式區塊

此區塊是由表單frmDocument所構成的，而在表單裡是使用VB中的影像盒 (Image)、線條工具 (Line)、標籤 (Label)、雙併盒 (ComboBox) 以及列表盒 (ListBox)，來建立整個模糊控制系統的雛型架構及模糊規則式，其中影像盒 (Image) 用來產生三個階段型態的位態 (Place)、三個階段型態的態變 (Transition) 以及標記 (Token) 圖形，線條工具 (Line) 用來產生有向性弧線 (Directed Arc) 圖形，標籤 (Label) 用來產生位態名稱、態變名稱以及第二階段型態推論位態上名稱，雙併盒 (ComboBox) 用來控制所想控制的位態以及態變，列表盒 (ListBox) 用來儲存每條繪製完成的模糊派翠網路徑，表1是各個設計完成的物件完成圖。

表1 各個設計完成的物件圖

<p>影像盒 (Image)</p> <p>第一階段型態 第二階段型態 第三階段型態</p>  <p>Place Place Place</p> <p>Transition Transition Transition</p> <p>Token</p>	<p>雙併盒 (ComboBox)</p> 
<p>列表盒 (ListBox)</p> <p>所有路徑記錄表</p> 	<p>標籤 (Label)</p> <p>位態名稱</p> <p>態變名稱</p> <p>模糊集合名稱</p> <p>線條工具 (Line)</p>  <p>有向性弧線 (Directed Arc)</p>

而其中在路徑表儲存路徑的程式方法上，是應用了如圖10的程式流程方式所設計完成。

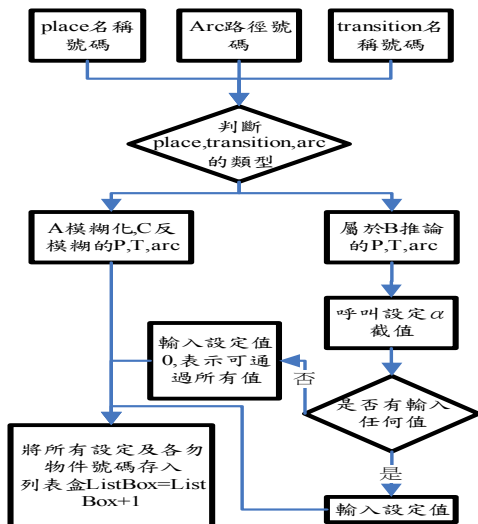


圖10 路徑表儲存路徑的方法程式流程

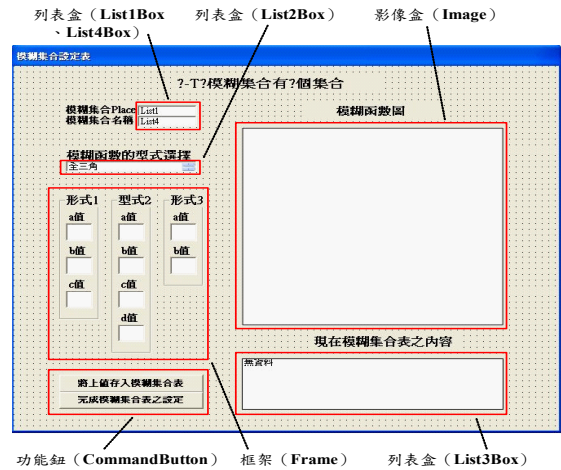


圖11 表單 (FormSet1) 視窗

是以Image1_MouseUp與Image2_MouseUp的事件當作是觸發的條件，而其中Image1代表位態 (Place)、Image2代表態變 (Transition)，而程式中的變數i (代表位態的號碼)、變數j (代表位態的號碼)、變數l (代表Arc的號碼) 以及變數wv1 (代表此條路徑的截值 α)，是將每次由使用者所設計產生的單條路徑，經過編排後儲存至路徑表上，而整個路徑表上的路徑，便可用來表示整個系統模型的各物件間之相互關係。

系統模糊相關設定程式區塊

在此區塊包含了第一階段型態介面的模糊變數及第一 (2輸入)、三 (1輸出) 階段型態介面的態變模糊歸屬函數的設定，其中模糊變數是利用標籤 (Label) 的方式來表示，而第一、三階段態變模糊歸屬函數則是利用2個表單 (FormSet1)、表單 (FormNS) 來設定，而表單 (FormNS) 是用來判斷選擇的態變之模糊歸屬設定是屬於哪個陣列 (2輸入-Set1A1、Set1A2; 1輸出-Set1C1)，之後在判斷出此階段所有連接的位態，將所有判斷的值給表單 (FormSet1) 使用，而表單 (FormSet1) 裡其中包含了用2個列表盒 (List1Box、List4Box) 來表示此態變上所連接的位態集合 (Place set) 及每個位態集合的名稱，用1個列表盒 (List2Box) 來表示模糊函數的型式選擇，用三個框架 (Frame) 來表示模糊函數的型式範圍輸入，用6個影像盒 (Image) 來表示模糊函數的6種型式圖，用列表盒 (List3Box) 來表示所有在此表單輸入的模糊歸屬函數的設定，最後在用2個功能鈕 (CommandButton) 來表示是否要將設定值存入列表盒及是否完成所有設定，而設計完成的表單 (FormSet1) 如圖11所示。

而在此模糊歸屬函數設定，所使用的程式方法，是應用圖12的程式流程方式所設計完成。

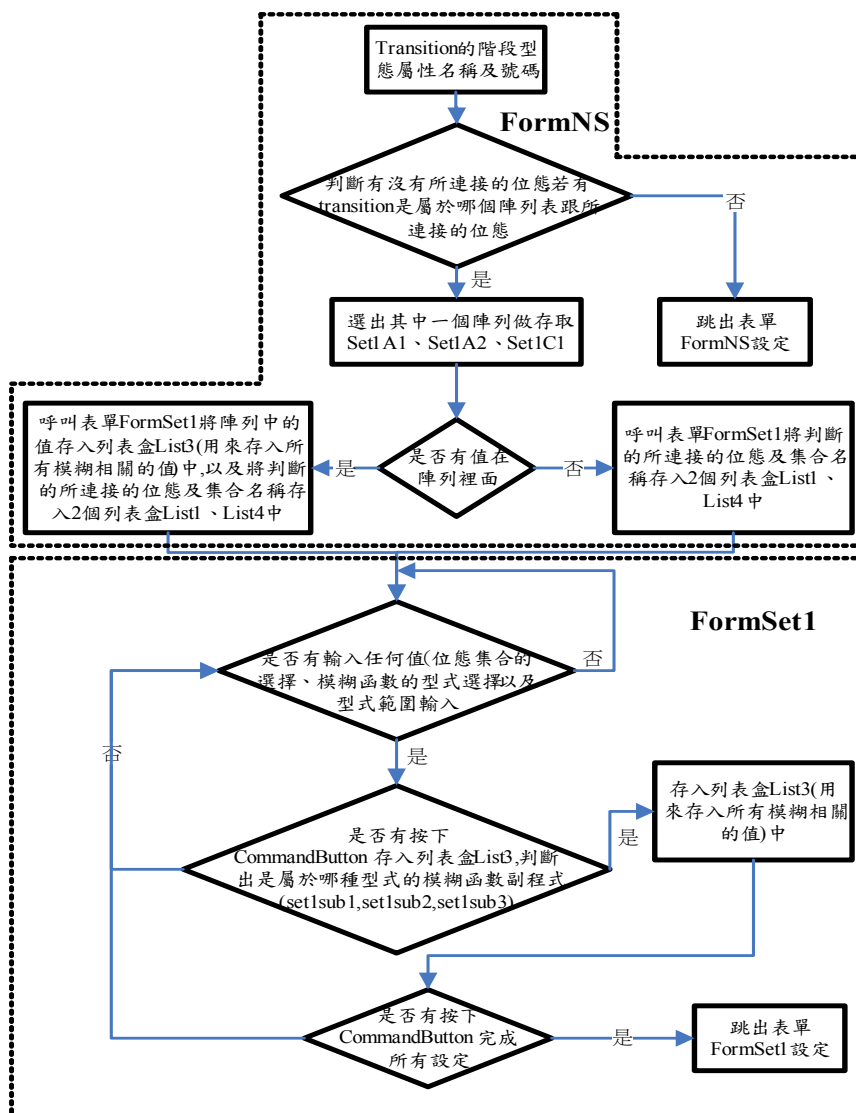


圖12 模糊歸屬函數設定的程式流程圖

而其中幾個比較重要的程式區塊，所使用的程式碼，說明如下：

1. 將判斷的所連接的位態及集合名稱存入2個列表盒List1、List4中。

其中程式中利用了NSPTAY()副程式來判斷所連接的位態及集合名稱，再利用IF判斷出所要給的階段態變選出陣列表Set1A1、Set1A2以及Set1C1，最後再利用FOR迴圈設定給表單（FormSet1），若沒有連接的態變就跳出表單（FormNS），由於第三階段跟第一階段使用的方法一樣，故將之省略，而NSPTAY()副程式跟判斷系統矩陣的方法是一樣的，故在之後內容會提到。

2. 是否有按下CommandButton存入列表盒List3及判斷出所對應的模糊函數型式副程式。

其中程式中利用了CommandButton1_Click()的事件當作是觸發的條件，接著利用Me.List2.Text來判斷出事哪一個模糊函數型式副程式。而在副程式上可分為三種型態的副程式（set1sub1、set1sub2、set1sub3），而就set1sub1的副程式而言，此副程式是用被判斷出為模糊函數值為a,b,c“全三角”、“左梯形”、“右梯形”時用，而在其中先由For迴圈判斷出在列表盒List3上有無相同的模糊集合位態，若有將set1s1d變數設為true，

若無設為false，之後在經由Select Case來判斷要存入哪個陣列表，若為有相同的模糊集合位態就存入原陣列表位子，若無相同的模糊集合位態就存入列表盒List3的下一個陣列表位子。

同理，在另外兩個模糊函數型式副程式（set1sub2、set1sub3）都跟set1sub1的程式差不多，只是在模糊函數值a, b, c的地方不一樣，如set1sub2是用在“全梯形”上，而它的模糊函數值是a, b, c, d，如set1sub3是用在“左三角”、“右三角”上，而它的模糊函數值是a, b。而經由以上三個副程式，便可將每次使用者所輸入之相關設定儲存至列表盒（List3Box）上。

系統路徑表判別程式區塊

在此區塊是由一個PTAY()副程式所構成的，而利用這副程式可以讓整個模糊的系統矩陣產生在表單（FormRC）上或提供模糊控制系統推演的需求，而此程式方法是應用了如圖13的程式流程方式所設計完成。

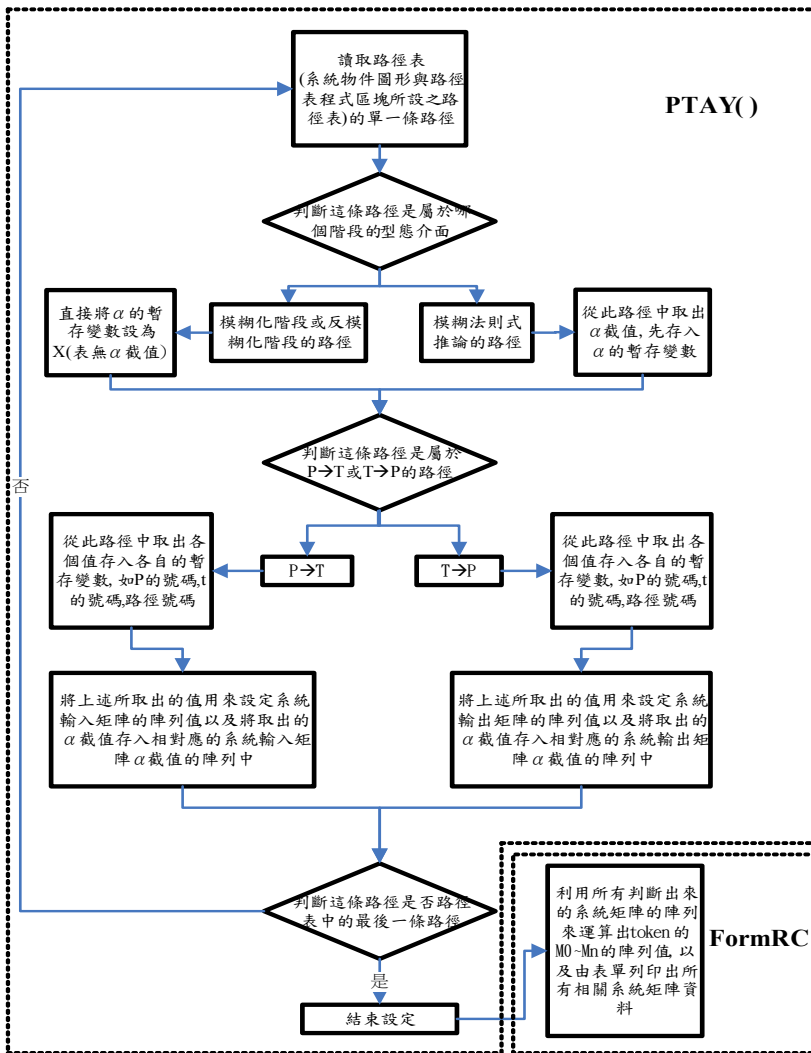


圖13 系統路徑表判別程式區塊的程式流程圖

而在PTAY()的程式碼，一開始程式先將所有所要用來儲存的陣列初值，經判斷設定出來，在利用FOR迴圈去取路徑表List1中各個路徑的值去做分析，在分析的方式是以暫存變數all2，來存每次所取進來的路徑，再利用各種條件IF來分析出此條路徑中的各個值(α截值、P的號碼、t的號碼、路徑號碼)，在經由這些值，以供設定各個系統陣列。另一方面，在列印出值的表單FormRC，用的方法是將剛剛PTAY()副程式分析出的系統陣列值，直接由表單FormRC列印出來，使我們可以清楚的看到整個系統的相關矩陣值。

系統模糊推理演示及判別與演算程式區塊

此區塊可以說是整個模擬系統的核心，而此區塊是由munRun_Click()事件、nRun()副程式以及Timer1物件所構成的。

munRun_Click()事件，是執行程式時所觸發的事件，而其內部的程式是將PTAY()副程式(PTAY()副程式是上述系統路徑表判別程式區塊中的副程式，用來判別出整個系統的各條路徑中的各個值如α截值、P的號碼、t的號碼以及路徑號碼，以供設定出整個系統的矩陣)所產生的系統各矩陣載入後，並且判斷整個模糊系統架構及設定是否有嚴重的錯誤，以及設定一些演算程式所需的初值和暫存陣列、變數等，若系統無重大的錯誤，最後便會將所設定的值傳給nRun()副程式使用。

而nRun()副程式，是用來判斷出每次Token所在之位置和所要執行的路徑及型式，以提供給Timer1物件內部程式做使用。

而Timer1物件，則是此區塊中最重要的推論演示物件程式，而使用

的方法是將nRun()副程式所產生之每次所要執行的路徑及型式，取出後分析，再經由判斷後產生出所要動作的方式及結果，而本區塊程式是利用了派翠網中標記的動態方式及結果來分析整個模糊系統推論的過程，並且間接的利用了模糊值及模糊推論所應用的推論演算法，來修改了派翠網在事件觸發上正確觸發的選擇，有利於提升派翠網分析的效率及能力。整個區塊所使用的程式流程方法如圖14所示。

而其中幾個比較重要的程式區塊，所使用的程式碼，說明如下：

1. nRun()副程式：

一開始先由前判斷出來的fnayP()陣列(專門儲存現在token所停留在Place的位置及數值之陣列)以及fnayT()陣列(專門儲存現在token所停留在Transition的位置及數值之陣列)的情況，利用If條件來判斷現在nayRunY變數是為P→T還是T→P，若為P→T就表示Token現在的位置是停留在Place上，再來就由剛剛所產生的fnayP()陣列裡的狀況，提供FOR迴圈找出各個Place上的Token之後所要連接的路徑，之後將所有FOR迴圈找出的每條路徑存入fnayRun陣列裡。

反之若為T→P就表示Token現在的位置是停留在Transition上，再來就由剛剛所產生的fnayT()陣列裡的狀況，提供FOR迴圈找出各個Transition上的token之後所要連接的路徑，之後將所有FOR迴圈找出的每條路徑存入fnayRun陣列裡。最後再由For nRunW = 1 To L的迴圈以及If nR → 0 Then的條件，來判斷出剛判斷的fnayRun陣列裡是否為無路徑執行或為最終的Token，若為是的情況就結束起動Timer1的演示動作；若為不是的情況就起動Timer1的演示動作。

2. Timer1 P→T的Case Is = 1程式，用來判斷Token在Place上為P→T的三種類型及三種類型所採取的演算和演示的方法：

此程式是前fnayRun陣列裡所儲存的每條路徑都為P→T狀態且當Token在Place上的狀況時，可能會遇到的三種型態路徑，CASE Is = 1代表現在所要演算及演示的Token在Place上，但要演示前要先經過內部程式三種型態路徑的判斷，才決定動作的型式及方式。

一開始先用FOR迴圈將前fnayRun陣列裡所儲存的每條路徑取出來作分析，而這時利用IF條件來判斷每次取出之路徑是屬於哪種型態P→T的路徑，第一種為由模糊化Place至模糊化Transition的型態，而此型態所要採取

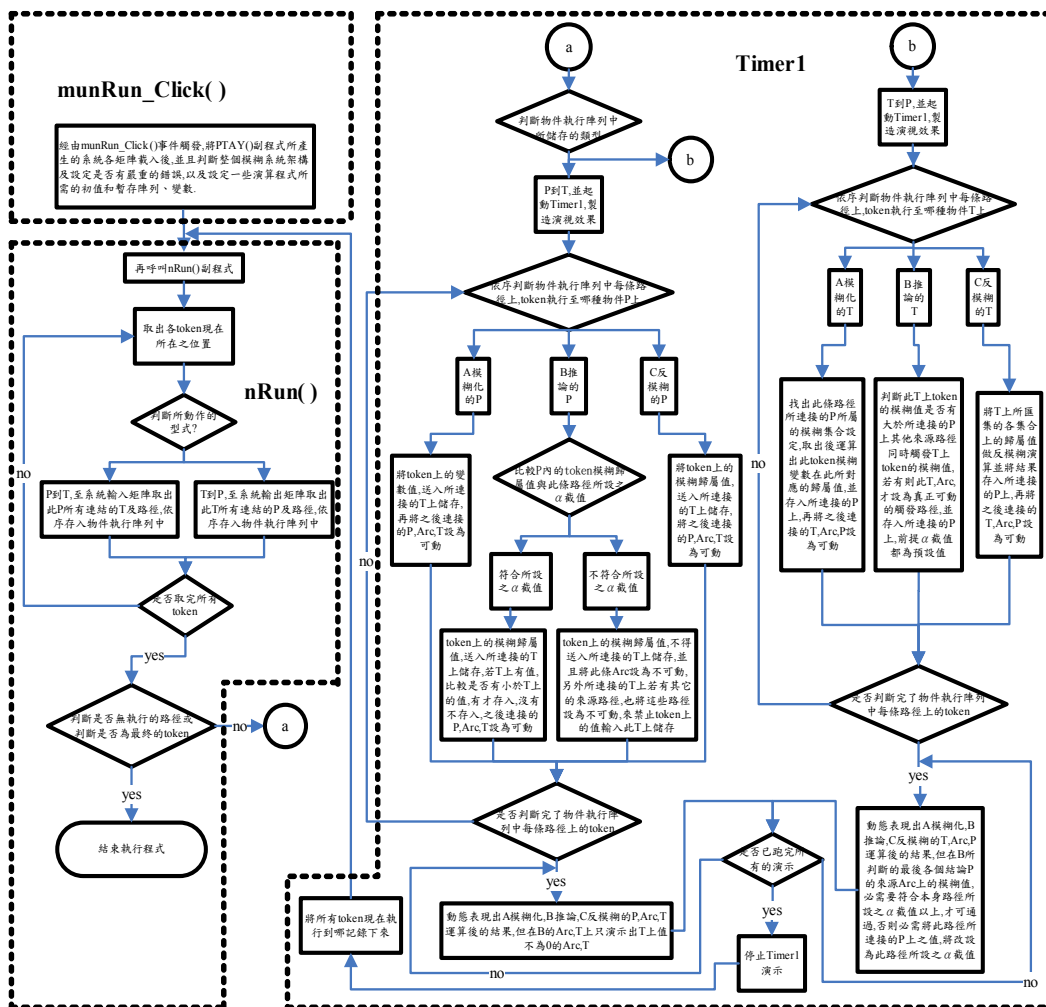


圖14 系統模糊推理演示及判別與演算程式區塊的程式流程圖

的演示及演算方法，只需將模糊化Place上的Token模糊值丟至所連接的Transition上的暫存陣列，決定動作後再作出上演示方法，有做出演示的路徑就在fnayRun陣列裡的該條路徑上設1，表此路徑Place動作過（因此被設為1，是因為Timer1無法將P→T的動作一次動作完畢，故就是要讓Timer1能了解剛剛有動作的是哪個路徑的Place物件，而之後就可根據剛剛有動作的路徑之Place物件去找所要接下去演示動作的Arc以及Transition物件，而Arc物件的動作就是由後面的程式CASE Is = 2去演示，而Ttransition物件的動作就是由後面的程式CASE Is = 3去演示）。

第二種為前提推論Place至推論Transition的型態，而此型態所要採取的演示及演算方法，是先要比較前提推論Place上的Token模糊值是否有無比該條路徑所設之 α 截值大，若無則不去觸發該條路徑所連接的狀態，相對的所連接的狀態上其它相連接的位態也都失去觸發此態變的能力（因為AND的關係，AND的來源必需都要有，才可動作），若有則該條路徑可通過Token模糊值，接著再將通過的Token模糊值，儲存至所連接的Transition上的暫存陣列，但再此必需還要比現在所存入之Token模糊值是否有比現在Transition上的暫存陣列裡的模糊值小（Transition上的暫存陣列裡的模糊值，可能會儲存前路徑所判斷及同樣連接至此Transition上的推論Place上的Token模糊值），此演算

方法相當於Mamdani邏輯積演算（min運算法），由前決定動作後再作出上演示方法，有做出演示的路徑就在fnayRun陣列裡的該條路徑上設1，表此路徑Place動作過。

第三種為結論推論Place至反模糊化Transition的型態，而此型態所要採取的演示及演算方法，不需像模糊化將Place上的Token模糊值丟至所連接的Transition上的暫存陣列，只需作出演示的方法，因為結論推論Place的token模糊值，是屬於單獨各個模糊集合上的推論結果，故等一下只需由所連接的Transition取出Token模糊值即可，跟前兩型態一樣有做出演示的路徑就在fnayRun陣列裡的該條路徑上設1，表此路徑Place動作過。

3. Timer1 T→P的Case Is = 1程式，用來判斷Token在Transition上為T→P的三種類型及三種類型所採取的演算和演示的方法：

此程式是當fnayRun陣列裡所儲存的每條路徑都為T→P且Token在Transition上的狀況時，可能會遇到的三種型態路徑，CASE Is = 1代表現在所要演算及演示的Token在Transition上（而這裡的CASE Is = 1跟前P→T程式的CASE Is = 1是相等的意義，只是差別在於Token所處位置的不同，同理，在這裡也會有CASE Is = 2跟CASE Is = 3的程式，但很明顯的由於是T→P的狀態，所以在此的CASE Is = 3所要觸發的就變成是Place

物件，不再是Transition物件，而CASE Is = 2則不變，所要觸發的還是Arc物件）。

一開始也是用For迴圈將fnayRun陣列裡所儲存的每條路徑取出來作分析，接著利用IF條件來判斷每次取出之路徑是屬於哪種型態T→P的路徑，第一種類型為由模糊化Transition至推論Place（模糊集合）的型態，而此型態所要採取的演示及演算方法，先取出此路徑的模糊化Transition所對應的模糊歸屬函數設定陣列Set1A1、Set1A2（先前在系統模糊相關設定程式區塊中，由使用者所設定的各模糊集合歸屬函數及範圍值），再將模糊化Transition上的Token的模糊值，由Case丟入此路徑所連接的推論Place所屬的模糊集合歸屬函數副程式（全三角、左三角、右三角、左梯形、右梯形、全梯形）去運算，之後判斷此Place所屬的模糊集合是否有無加強語氣h，若有便將運算的結果重新加上h次方後運算，若無則不用重新運算，由上判斷完後，再將運算的結果丟入所連接的推論Place的暫存陣列，最後再作出上演示動作，有做出演示的路徑就在fnayRun陣列裡的該條路徑上設1，表此路徑Transition動作過（跟先前型態P→T用意相同）。

第二種類型為推論Transition至結論推論Place的型態，而此型態所要採取的演示及演算方法，是先比較是否此路徑推論Transition上的Token模糊歸屬值，為所連接的結論推論Place眾多連接路徑中模糊歸屬值最大的一條路徑，若不是，則不做任何動作（表示不送Token至此路徑所連接的結論位態上），而此演算方法相當於Mamdani邏輯和演算（max運算法），另一方面，若為是，則再去比較此路徑推論Transition上的Token模糊歸屬值是否有無比此條路徑所設之 α 截值大，若有，則將Transition上的Token模糊歸屬值丟入此路徑所連接的結論推論Place的暫存陣列，若無，則該條路徑所通過Token模糊歸屬值，變為此路徑所設之 α 截值，最後有做出演示的路徑就在fnayRun陣列裡的該條路徑上設1，表此路徑Transition動作過（跟先前型態P→T用意相同）。

第三種類型為反模糊化Transition至反模糊化Place的型態，而此型態所要採取的方法是重心解模糊法，程式所使用的方式是先取出此路徑反模糊化Transition所對應的模糊歸屬函數設定陣列Set1C1，取出後利用For迴圈將整個反模糊化上的模糊變數範圍分為20次，取出每次的模糊變數來分析，而在每次分析上所使用的方法是用For迴圈將每次取出的模糊變數，去找出此次的模糊變數是屬於哪幾個反模糊化Transition所連接的模糊集合的模糊變數範圍，找到的模糊集合再利用Case把模糊變數代入所對應的模糊歸屬函數副程式，算出模糊歸屬函數值，之後判斷此次的模糊集合是否有無加強語氣h，若有便將運算的結果重新加上h次方後運算，若無則不用重新運算。

由上判斷完後的運算結果，還要經過兩次比對，第一要先比對此次所算出的模糊歸屬函數值是否有大於此模糊集合的推論歸屬值，若沒有才為此算出的模糊歸屬函數值，若有則要將此算出的模糊歸屬函數值設回模糊集合的推論歸屬值（表示不可超過前模糊集合Place上Token所推論的模糊歸屬值）。

第二則是要再比對set1c1nm暫存變數的内部是否已經儲存了有比此次所算出的模糊歸屬值還大的模糊歸屬值，若有則set1c1nm暫存變數内部的值不變，若沒有則由現在算出的模糊歸屬值取代（set1c1nm暫存變數是用來儲存單次模糊變數所找出的模糊集合所算出的模糊歸屬函數值，為什麼要用比對的方式，是因為單次的模糊變數可能會出現在各個模糊集合的範圍內，故經過For迴圈的演算後，我們只取最大的模糊歸屬值，來代表這一次模糊變數的模糊歸屬值），之後再將每一次所分析的模糊變數及它們所算出的模糊歸屬值，做一個總運算。

最後完成計算後，再將值丟入反模糊化Transition所連接的反模糊化Place暫存陣列上，再作出上演示動作，便可達到最後我們所想要得到重心解模糊法的模糊變數值的演示動作。

人機介面程式區塊

此區塊是運用VB中的功能表編輯器、ToolBar、StatusBar以及多重文件介面（MDI），來建立整個人機介面的控制項，使得系統設計者可藉由這些控制項與模擬系統内部的程式區塊來作一溝通的管道使用。如圖15是設計完成的人機介面平台：

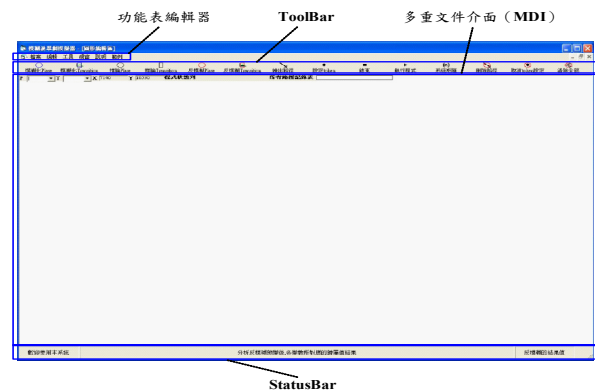


圖15 模擬系統人機介面平台

而在功能表編輯器所產生的功能介面是稱為功能表列，此功能表列是具備了整個模擬系統所需的功能項目，其中包函了檔案、編輯、工具、視窗、說明以及範例等項目，而其中它們各自的内容項目又包括了：

1. 檔案—用來處理繪製的模型檔案，裡面包括了此整個系統模型執行的結果列印、相關系統模型的參數列印以及開新檔案重新繪製系統模型的處理。
2. 編輯—用來修改系統編輯區的物件，裡面包括了有：取消Token的設定(取消系統編輯區上第一階段模糊化Place之模糊變數Token的命令項)、刪除路徑(刪除系統編輯區上Place跟Transition之路徑或Transition跟Place之路徑的命令項)以及刪除全部物件。
3. 工具—用來繪製系統編輯區裡所需的物件及對繪製完畢的系統所下的命令項目，裡面包括了有：模糊化Place(產生第一階段模糊化位態的命令項)、模糊化Transition(產生第一階段模糊化態變的命令項)、推論Place(產生第二階段推論位態的命令項)、推論Transition(產生第二階段推論態變命令項)、反模糊Place(產生第三階段反模糊化位態的命令項)、反模糊Transition(產生第三階段反模糊化態變的命令項)、繪出路徑(繪出系統編輯區上Place跟Transition之路徑或Transition跟Place之路徑的命令項)、設定Token(產

生系統編輯區上第一階段模糊化Place之模糊變數Token的命令項)、系統矩陣(產生整個系統模型的矩陣表的命令項)、執行程式(將整個系統模型做動態演示推論的命令項)。

4. 視窗—用來更改視窗顯示的方式,裡面包括了有:重疊顯示、水平並排、垂直並排、排列圖示。
5. 說明—用來說明此模擬系統的相關資訊。
6. 範例—裡面有幾個範例,可供使用者學習。

而在ToolBar所產生的功能介面是稱為快捷鈕列,此快捷鈕列是將系統編輯區裡較常常使用的功能表列裡的功能取出,變成一快捷鈕列,方便使用者能快速的完成系統模型的繪製,而其中包括了有:模糊化Place、模糊化Transition、推論Place、推論Transition、反模糊Place、反模糊Transition、繪出路徑、刪除路徑、產生token、取消token、系統矩陣、執行程式、清除全部以及結束程式。

而在多重文件介面(MDI)是將系統物件圖形與路徑表程式區塊(表單frmDocument)的程式功能整合在一起,將此部分稱為系統編輯區,利用此編輯區域來繪製整個系統的模型,而在此系統編輯區除了前(表單frmDocument)的程式功能外,再多加了三個標籤(Label),用來顯示當時系統所有的狀態情況及滑鼠在系統編輯區上所在的X、Y區域位置,使設計者方便了解一些系統的相關狀態及編輯物件時所需的區域位置。

而在StatusBar所產生的功能介面是稱為系統模糊結果狀態列,用來顯示整個系統模型經由模糊推論後所得的最後相關模糊參數值,使設計者能快速的了解整個最後的相關模糊參數值。

9. 模擬系統產生之推論演示說明及結果分析

設定完成後,若想要知道整個模糊派翠網模型的相關系統矩陣及各Place、Transition物件在系統中所代表之意思,可由按下快捷鈕列的“系統矩陣”的按鈕,便可了解。按下後會出現模糊派翠網矩陣表如圖16。

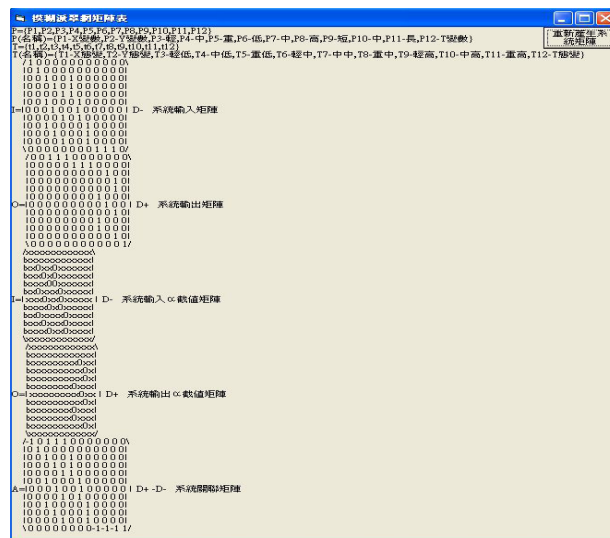


圖16 模糊洗衣機控制系統之模糊派翠網矩陣表

接下來便可按壓快捷鈕列的“執行程式”的按鈕,讓模擬系統進行整個模糊洗衣機控制系統模型的模糊推論演算及演示,而整個推論演算及演示過程如圖17說明:

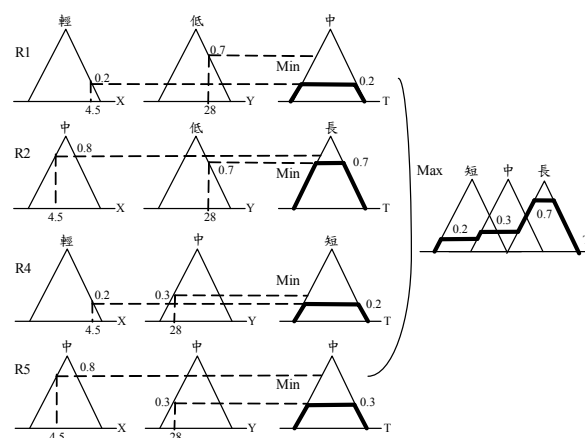


圖17 4條規則經由Mamdani的推理演算方法

將各模糊集合結論Place (T短、T中、T長)的歸屬值,傳送至反模糊T態變上。

最後是圖18,是將之前觸發傳送到的反模糊T態變上的各模糊集合結論Place (T短=0.2、T中=0.3、T長=0.7)的歸屬值解模糊,由重心解模糊法,得到:

$$y = \frac{\sum_{i=1}^m x_i \cdot \mu_A(x_i)}{\sum_{i=1}^m \mu_A(x_i)}$$

$$T = \frac{5 \times .2 + 10 \times .2 + 15 \times .2 + 20 \times .2 + 30 \times .2 + 35 \times .3 + 40 \times .3 + 45 \times .3 + 50 \times .3 + 55 \times .3 + 60 \times .4 + 65 \times .6 + 70 \times .7 + 80 \times .7 + 85 \times .6 + 90 \times .4 + 95 \times .2}{.2 + .2 + .2 + .2 + .2 + .3 + .3 + .3 + .3 + .3 + .4 + .6 + .7 + .7 + .6 + .4 + .2}$$

T=59.2分。

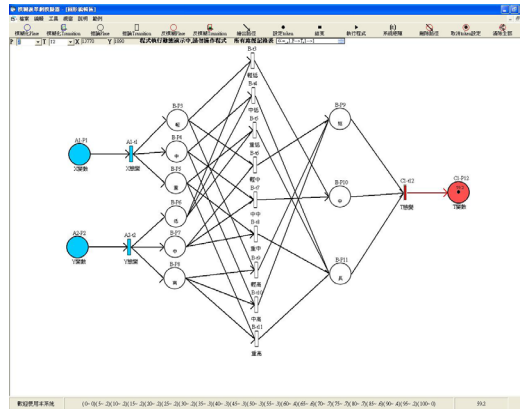


圖18 模糊控制系統推論演示

10. 結論

本研究中提出了一套具有分析模糊性控制系統推論的模糊派翠網定義模型，來解決現今複雜的系統當中常所伴隨著許多模糊、不確定、非線性及時變的事物或因素，並且本研究應用了此定義模型，開發出了一套不同型態的模擬系統軟體，也利用了此模擬系統軟體，成功的設計出一個具有模糊控制的洗衣機控制系統，故經由此驗證了本研究所提出的模糊派翠網定義模型是可行且正確的。在本研究之貢獻有以下兩部份：

1. 提供了一個新詮釋的模糊派翠網之定義模型。
2. 提供了一個新型態系統推論工具，讓設計者能於系統硬體實際製作前，可利用此系統推論工具，來先了解所設計的控制系統内部的運作過程，並且藉由此模擬系統所具備的模糊控制之建模方式，
3. 便能讓設計者能夠不必去理會所要分析的受控系統之數學模型，也可不必去寫出它的控制轉移方程式，只要利用設計者本身對受控系統的經驗法則及認知，便能順利的建立起系統内部的控制邏輯之模型，
4. 另一方面，也藉由了模擬系統另一項所具備的派翠網強大系統分析之能力，利用此派翠網圖形的動態演示及分析，來了解整個控制系統模型的模糊推演的預測過程及結果，以便精確、快速的分析出所建立的系統結構及控制邏輯的正確性，以減少在除錯上不必要浪費的時間及精神。

參考文獻

[1] S.M. Chen, "Weighted Fuzzy Reasoning Using Weighted Fuzzy Petri Nets," IEEE Transaction on Knowledge and Data Engineering, 14, 2, 386-397, 2002.

[2] Satoru Fukami, Masaharu Mizumoto, Kokichi Tanaka, "Some considerations on fuzzy conditional inference," Fuzzy Sets and Systems, 4, 3, 243-273, 1980.

[3] Kurt Jensen, "Coloured Petri Nets : A High Level Language for System Design and Analysis," Lecture Notes in Computer Science, 483, 342-416, 1990.

[4] Tadao Murata, "Petri Nets: Properties, Analysis and Applications," Proceedings of the IEEE, 77, 4, 541-583, 1989.

[5] H. Scarpelli, F. Gomide, and R. Yager, "A reasoning algorithm for high level fuzzy Petri net," IEEE Transaction on Fuzzy Systems, 4, 3, 282-294, 1996.

[6] V. R. L. Shen, and F. Lai, "Requirements Specification and Analysis of Digital Systems Using Fuzzy and Marked Petri Nets," IEEE Transaction on Systems, Man, and Cybernetics, 28, 5, 748-754, 1998.

[7] Sheng-Ke Yu, "Comments on "Knowledge Representation Using Fuzzy Petri Nets," IEEE Transaction on Knowledge and Data Engineering, 7, 1, 190-192, 1995.

[8] X. Li, W. Yu, and F. Lara-Rosano, "Dynamic Knowledge inference and learning under adaptive fuzzy Petri net framework," IEEE Transaction on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 30, 4, 442-450, 2000.

[9] L. A. Zadeh, "Fuzzy sets," Information and Control, 8, 3, 338-353, 1965.