

系統動力學線流圖資料保留處理研究 -以銷售庫存作業為例

曾光輝 Tseng Kuang-Hui

中山大學資訊管理學系

light@seed.net.tw

Abstract

System Dynamics provides utility to build dynamic complex system model. There are many application examples of System Dynamics in many domains. We can simulate the system structure with System Dynamics stock-flow diagram and Dynamo equation. The quantitative data from System Dynamics model will provide references for decision. The value of Level and Rate in stock-flow diagram will change with the time step probably. In some domain applications need approach to keep these values. This research provided an approach to keep these values used the technique of stock-flow diagram, with no extra program language. And build a sale-inventory-management model to demonstrate the application of this approach.

Keywords: System Dynamics, Stock and Flow Diagrams, Inventory Management.

摘要

系統動力學提供建立動態性複雜性統的工具，在許多領域都已經有應用的實例。透過系統動力學的線流圖以及 Dynamo 方程式來模擬系統結構，所獲得系統量化的數據，可以提供決策的參考。線流圖中的積量與率量的值有可能隨著時間而變化，在應用的系統中若需要保留積量與率量的值，則必須要有辦法把這些值紀錄下來。本研究提出來的方法可以不必透過外部的程式語言，直接以線流圖的建構技巧來達到這個目標，本研究同時也以實例來應用所提的方法，建構一個銷售與庫存管理作業的模型。

關鍵詞: 系統動力學、線流圖、庫存管理。

1. 前言

系統動力學的線流圖與 Dynamo 方程式等工具，提供建立動態性複雜的系統模型，藉由積量、率量、輔助變數等元件間的組合模擬系統結構，其應用遍及工程、自然科學、社會科學、企業管理等領域。利用系統動力學模型除了可以做決策的參考，更能透過對系統結構的了解，對所模擬的實際系統有進一步的了解。

在建立系統動力學的線流圖模型時，積量、率量及輔助變數的值受到有可能隨著時間而改變。倘若必須依據時間順序將前面時間點的值做處理，則必須有辦法把這些值紀錄下，才能做進一步的處理。本研究所提的方法不必透過外部的程式語言(VB、VC++)開發輔助程式，直接以線流圖建構的技巧，將線流圖中的資料保留下來做進一步的處理，並提出一個實例做應用的展示。

2. 文獻探討

2.1 系統動力學

系統動力學主要是研究動態性複雜 (Dynamic Complexity) 的系統，所使用的主要工具有因果關係圖、線流圖以及 Dynamo 方程式等。

因果關係圖以系統中各個變量間彼此的關係來表達，兩個變量間的關係可以是正關係，代表其中一個變量增加也會造成另一個變量的增加，反之則是負關係。而相同方向 (順時鐘或逆時鐘) 的關係則組成一個回饋

環，當回饋環中有奇數個負關係時稱為負環，反之則為正環。負環代表目標追尋的系統結構，而正環則是增長型或衰退型的系統結構。

線流圖進一步將系統中的變量歸類，包含以下幾個元件：

- (1) 積量(Level)：系統內累積流量的變量，積量為一累積值，有些文獻中稱為存量(Stock)。
- (2) 率量(Level)：單位時間內流入或流出積量的流量。
- (3) 流線(Flow)：包含實體流及訊息流，實體流用來標示率量對積量的影響為流入或流出。而訊息流則標示出積量、率量、輔助變數間的關聯性。
- (4) 輔助變數(Auxiliary)：包含常數或函數，用以協助描述率量與積量間的關係。

由線流圖可以更詳細表達系統間各個變量的關係，同時搭配相關的數學方程式，可以表示出各個變量的數值變化情形。

Dynamo 方程式則是用數學式描述系統間變量的變化，藉此獲得更詳細的系統模型結構。

2.2 庫存管理

庫存管理一般分為兩類存貨模式：定期訂購模式(P Model)、定量訂購模式(Q Model)，本研究對於這兩種存貨模式的探討，主要是參考 Krajewski 及 Ritzman(2001)的著作。

定期訂購模式中是在特定的時間對存貨盤點，每次盤點後再訂購補足存貨的安全庫存量。在定期訂購的模式下，安全存量及盤點週期是兩個重要議題，安全存量計算方式如下：

- (1) 安全存量=服務水準之常態分配值*盤點週期加上前置時間需求的標準差

- (2) 初期的盤點週期是由經濟訂購量計算得來，以求得較低的總存貨成本，計算方法如下：

$$\text{盤點週期}=\text{經濟訂購量}/\text{年需求量}$$

後續執行時，管理者可以依據實際需要再調整盤點週期。

定量訂購模式主要是維持庫存產品數量以符合銷售的需求，其中一個重要的議題是要決定存貨的再訂購點以及訂購量。再訂購點是指當產品數量低於再訂購點的數量值時，就必須發出產品訂購單。其中再訂購點的決定必須考慮經濟訂購量(EOQ, Economic Order Quantity)，避免存貨成本過高，且要能符合一定的服務水準。當產品需求不確定時，再訂購點必須考量到貨的前置時間以及安全存量(Safety Stock)，其關係如下：

$$\text{再訂購點}=\text{前置時間平均需求}+\text{安全存量}$$

而安全存量的計算如下：

$$\text{安全存量}=\text{服務水準之常態分配值}*\text{前置時間需求標準差}$$

本研究所舉的例子，採用定量訂購模式，由上述公式建立一個量化的模型。

3. 線流圖資料保留

線流圖模型是做情境分析與輔助政策制定的基礎，線流圖中各個組成元件的值，會隨著時間單位的進行而改變。本研究在後面幾節提出幾種保留線流圖資料的技術，除了保留積量、率量、輔助變數等線流圖元件的值之外，可以進一步應用來保留多組時間序列的值，並建構出不同時間單位等級的模型。

3.1 積量、率量、輔助變數資料保留

線流圖中的積量、率量、輔助變數都可以用積量來做資料保留，保留的方式包含累積式與非累積式的方法。累積式的資料保留方法，是由一個積量累積所要保留元件的值。非累積式的方法則是由一個積量保存元

件值的狀態，可能是某個前面時間點的元件內容。累積式與非累積式方法在結構上的差別是，非累積式的保存積量有一個 Out-Flow 率量將所累積的值減掉。參考圖 3.1.1 及圖 3.1.2，圖中虛線的次系統即是用來保留率量 Rate_1 的機制，Rate_1 前一個時間點的值，會被保留在 Level_2 中。圖 3.1.2 中 Rate_4 是累積式與非累積式差別的關鍵，Rate_4 會將 Level_2 前一個時間點的值消去，就成為非累積式的資料保留，圖 3.1.1 沒有 Rate_4 就成為累積式的資料保留，Rate_1 前面所有時間點的值會在 Level_2 中累積。

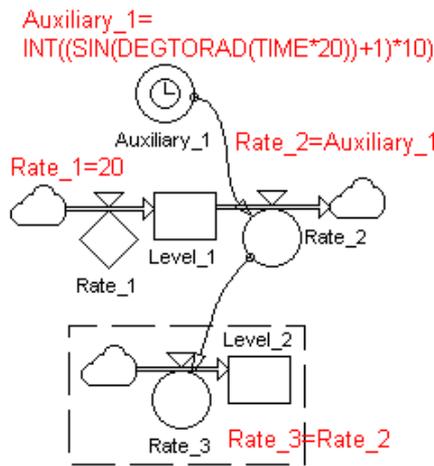


圖 3.1.1 率量資料保留(累積式)

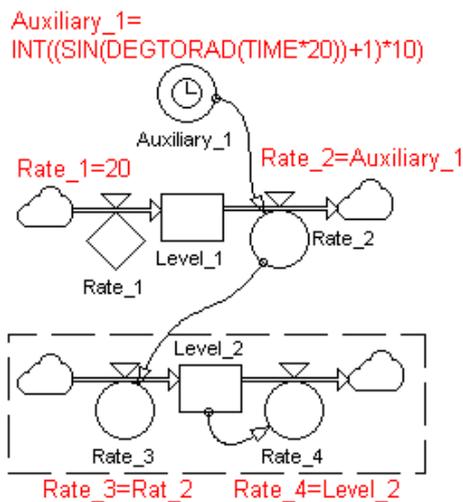


圖 3.1.2 率量資料保留(非累積式)

Time	Rate_2	Level_2
0	10.00	0.00
1	13.00	10.00
2	16.00	23.00
3	18.00	39.00
4	19.00	57.00
5	19.00	76.00
6	18.00	95.00
7	16.00	113.00
8	13.00	129.00
9	10.00	142.00
10	6.00	152.00

表 3.1.1 率量資料保留(累積式)

表 3.1.1 中 Level_2 的值即是以累積的方式保留 Rate_2 的值，而表 3.1.2 中的 Level_2 則是以非累積的方式來保留 Rate_2 的值。

Time	Rate_2	Level_2
0	10.00	0.00
1	13.00	10.00
2	16.00	13.00
3	18.00	16.00
4	19.00	18.00
5	19.00	19.00
6	18.00	19.00
7	16.00	18.00
8	13.00	16.00
9	10.00	13.00
10	6.00	10.00

表 3.1.2 率量資料保留(非累積式)

前述的資料保留還可以加上條件來選擇所要保留值的狀態，參考圖 3.1.3，圖中將保留資料送到 Level_2 的 Rate_3 更改為以條件篩選，所以只有符合條件的值才會被保留進 Level_2，圖中所篩選的是每 3 個時間單位的值才被保留。

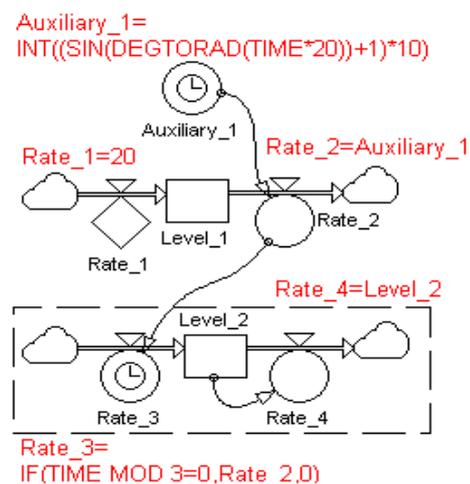


圖 3.1.3 選擇條件率量資料保留

3.2 時間序列資料保留

積量的值可能隨著時間而變化，倘若系統中有其他元件必須參考某個積量的前面時間點的資料，就必須將該積量前面時間點的值保留。比如說進貨量必須參考三天前的銷售量平均值，就必須將銷售積量三天前的值保留下來求平均值。

利用前面介紹的積量資料保存方法可以達到此要求，一個積量可以保留前一個時間單位的值，當積量串聯時，由於受前一個積量延遲的影響，多個積量便可以保留前面多個時間單位的值。

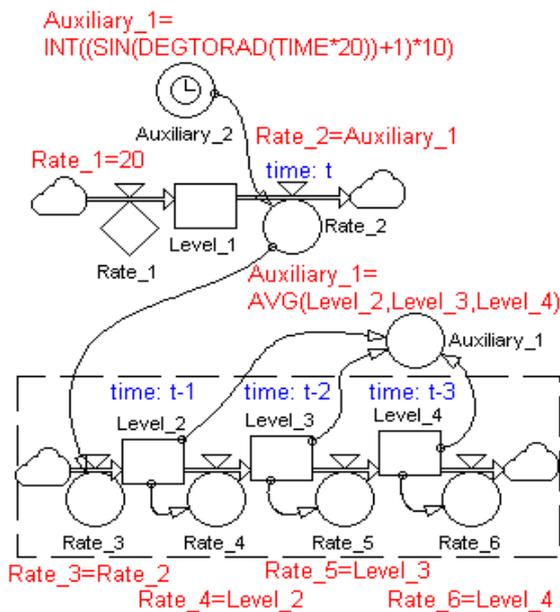


圖 3.2.1 時間序列資料保留

圖 3.2.1 利用三個積量，保留 t-1、t-2、t-3 三個時間點 Rate₂ 的值，Auxiliary₁ 是將保留下來的值計算出平均。

Time	Rate 2	Level 2	Level 3	Level 4
0	10.00	0.00	0.00	0.00
1	13.00	10.00	0.00	0.00
2	16.00	13.00	10.00	0.00
3	18.00	16.00	13.00	10.00
4	19.00	18.00	16.00	13.00
5	19.00	19.00	18.00	16.00
6	18.00	19.00	19.00	18.00
7	16.00	18.00	19.00	19.00
8	13.00	16.00	18.00	19.00
9	10.00	13.00	16.00	18.00
10	6.00	10.00	13.00	16.00

表 3.2.1 時間序列資料保留

表 3.2.1 中的 Level₂、Level₃、Level₄

的值分別保留 Rate₂ 的 t-1、t-2、t-3 三個時間點的值。

圖 3.2.2 是在 PowerSim 軟體下建立的模型，該軟體提供陣列的功能，Level₂ 及 Rate₄ 均改以一維的陣列來表示，其陣列元素個數為 3 時，即與圖 3.2.1 的效果一樣，可以保留前面三個時間單位 Rate₂ 的值 改以陣列表示的好處是，若要增加保留的時間時，只要增加陣列的元素個數即可，不必用多個積量與率量串接而佔用太多版面。

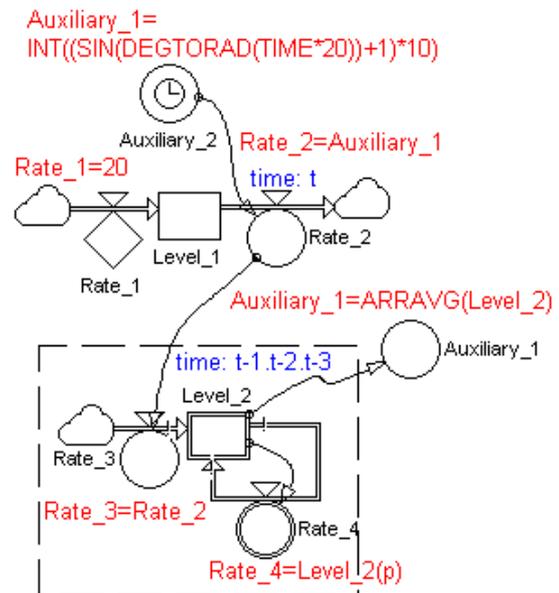


圖 3.2.2 時間序列資料保留陣列

3.3 差異時間單位等級處理

同一個系統中使用相同的時間單位，在建模一開始就必須決定好所使用的時間單位。然而在實際的應用中，可能會出現不同次系統使用不同時間等級的情況，比如說銷售數量的積量以天為單位，但是進貨所要參考的週銷售量統計資訊則是以週(7 天)為單位，因此在同一系統中便出現不同時間等級的單位需求。

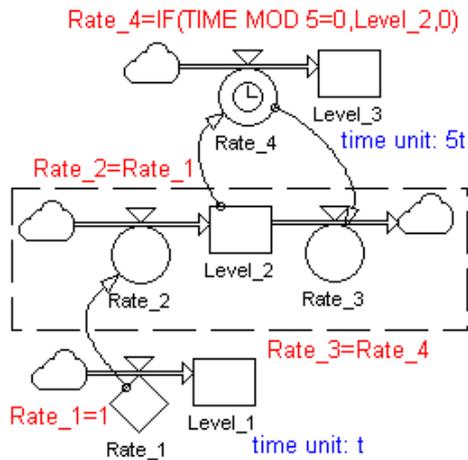


圖 3.3.1 差異時間單位等級延遲一時間單位
運用一組中介的積量與率量，可以達到時間等級差異的效果，圖 3.3.1 中虛線框的部分，即是將 time unit:1 的次系統中的率量變化值累積在 Level_2 中，等達到 time unit:5t 次系統的時間等級時，才透過 Rate_4 轉給 Level_3，因此 Level_3 與 Level_1 變化情形，便會按照時間的不同等級而有不同表現。

Time	Level 1	Level 2	Level 3
0	0.00	0.00	0.00
1	1.00	1.00	0.00
2	2.00	2.00	0.00
3	3.00	3.00	0.00
4	4.00	4.00	0.00
5	5.00	5.00	0.00
6	6.00	1.00	5.00
7	7.00	2.00	5.00
8	8.00	3.00	5.00
9	9.00	4.00	5.00
10	10.00	5.00	5.00
11	11.00	1.00	10.00

表 3.3.1 差異時間單位等級延遲一時間單位
從表 3.3.1 可以發覺，Level_2 事實上是保留了 Rate_1 的值，而由 Rate_4 控制在 5t 的時間點時，才將資料送到 Level_3，因此 Level_3 與 Level_1 便有不同的時間等級。

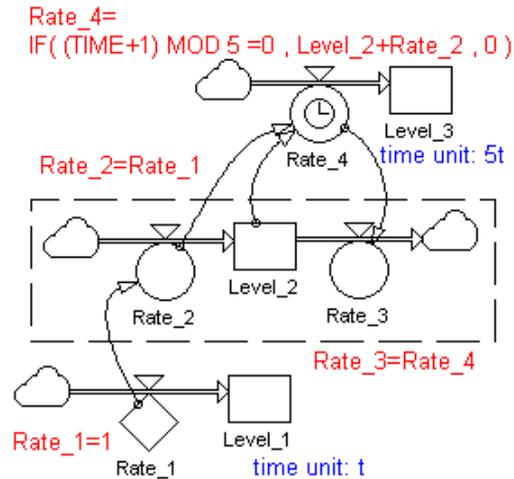


圖 3.3.2 差異時間單位等級無時間延遲
圖 3.3.1 中 Level_3 與 Level_1 除了有時間等級相差 5 倍的不同外，事實上 Level_3 還比 Level_1 延遲了一個時間單位，這是因為中介的積量與延遲的效果。倘若必須讓兩個時間等級的積量沒有延遲的現象，便必須修改成圖 3.3.2 的方式。

圖 3.3.2 中 Rate_4 取用 Level_2 的資料時，事實上是提早一個時間單位，以彌補做時間等級差異的中介積量對 time unit:5t 次系統所造成的延遲，在輸入 Level_3 時則以 Level_2 現有的值加上 Rate_2 的值，則剛好等於下一個時間點應該輸入的值，參考表 3.3.2。

Time	Level 1	Level 2	Level 3
0	0.00	0.00	0.00
1	1.00	1.00	0.00
2	2.00	2.00	0.00
3	3.00	3.00	0.00
4	4.00	4.00	0.00
5	5.00	0.00	5.00
6	6.00	1.00	5.00
7	7.00	2.00	5.00
8	8.00	3.00	5.00
9	9.00	4.00	5.00
10	10.00	0.00	10.00
11	11.00	1.00	10.00

表 3.3.2 差異時間單位等級無時間延遲

4. 案例應用說明

本節以銷售及庫存管理作業為例，實作前述的各項方法。

4.1 決定系統邊界

要建立銷售庫存管理系統的系統動力學模型，首先要定義出此系統的系統邊界，本範例模型目的要了解存貨策略對銷售的影響，因此系統兩個主要的次系統是產品庫存以及訂單。

產品庫存的次系統中包含表示產品庫存數量的積量，以及改變產品庫存的銷售函數率量。

訂單次系統則包含代表未實現訂單數量的積量，以及受運送延遲影響的訂單實現率量。

在此系統中尚包括為了計算安全存量以及再訂購點的幾組積量陣列，此部份即是前面章節所展示，本研究所提出來的建構方法。

4.2 線流圖

圖形左邊以虛線框出的是第 1 組時間序列資料保留的積量陣列，Level_1 的積量陣列會保留 7 天前的銷售情況，相關的 Dynamo 方程式如下：

$$R \quad \text{Rate}_1 = \text{ProductOutRate}$$

$$C \quad \text{ProductOutRate} = \text{MIN}(\text{SaleFunction}, \text{Inventory})$$

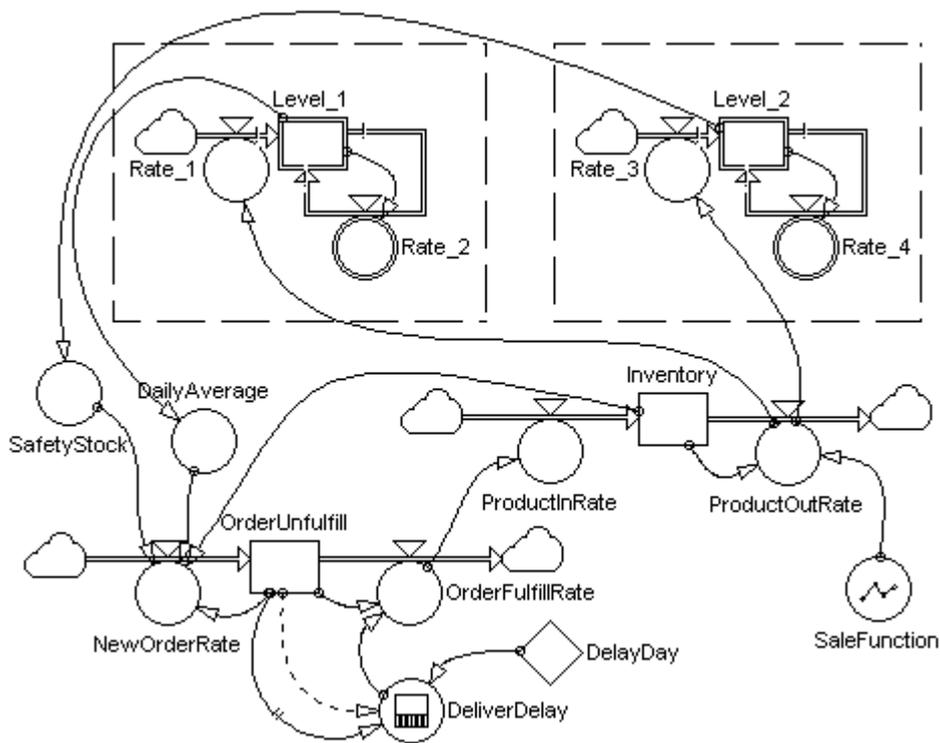


圖 4.2.1 銷售庫存作業線流圖

$$N \quad \text{Level}_1 = 0$$

$$L \quad \text{Level}_1 = +dt*(\text{Rate}_1|p=\text{FIRST}(p);0) - dt*(\text{Rate}_2(p)) + dt*(\text{Rate}_2(p-1)|p>\text{FIRST}(p);0)$$

其中 p 代表陣列註標的變數，Level_1 陣列共有 7 個陣列註標。

銷貨函數會判斷以目前庫存量與銷售情形兩者的最小值做為銷售的數量，由此避免市場需求量大於現有庫存時，造成庫存成為

負值的不合理現象。第 1 組時間序列資料保留主要是要計算 7 天前的日銷售平均數 DailyAverage。

$$N \quad \text{DailyAverage} = \text{ARRAVG}(\text{Level}_1)$$

此處運用 PowerSim 提供的陣列平均函數 ARRAVG, 求出 Level_1 積量陣列的平均數

圖形右邊虛線框出的是第 2 組時間序列資料保留的積量陣列，Level_2 的陣列會保留 3 天前的銷售情況。

N Level_2 = 0
 L Level_2 = -dt*(Rate_4(p))
 +dt*(Rate_4(p-1)|p>FIRST(p);0)
 +dt*(Rate_3|p=FIRST(p);0)
 R Rate_1 = ProductOutRate
 Level_2 陣列有 3 個陣列註標，此組時間序列資料保留主要是要計算安全存量。

C SafetyStock =
 ARRSTDDEV(Level_2)*1.28

由前面文獻探討可以知道，安全存量的計算是由服務水準的常態分配值乘上前置時間需求量的標準差，在此模型中我們所定的服務水準為 90%，查表求得其常態分配值為 1.28，而 ARRSTDDEV 是由 PowerSim 提供計算陣列標準差的函數。在此模型中貨物運送的延遲 DelayDay 是 3 天，因此在 Level_2 積量陣列中保留 3 天的銷售紀錄來計算前置時間需求量的標準差。

圖形左下角是訂單次系統的部分，相關 Dynamo 方程式如下：

N OrderUnfulfill = 0
 L OrderUnfulfill = -dt*OrderFulfillRate
 +dt*NewOrderRate
 R NewOrderRate =
 IF(Inventory+OrderUnfulfill<
 DailyAverage*3+SafetyStock,
 DailyAverage*7,0)
 R OrderFulfillRate = IF(OrderUnfulfill>0,
 MIN(OrderUnfulfill,DeliverDelay),0)
 C DeliverDelay =
 DELAYMTR(OrderUnfulfill,DelayDay,1)
 C DelayDay = 3

NewOrderRate 單位為 piece/day，此即代表訂單策略，主要是判斷現在的庫存 Inventory 加上現有未實現訂單量 OrderUnfulfill 的值，是否比再訂購點的值 DailyAverage*3+SafetyStock 還要低，若是的話則下訂單，且訂購量為 DailyAverage*7。

訂單實現 OrderFulfillRate 單位為 piece/day，受運送延遲 DeliverDelay 影響，由於延遲時間 DelayDay 較短，目前設為 3 天，所以採用 1 階延遲較符合實際情形。同時避免造成未實現訂單的量產生負值，若還有未實現訂單，則以貨運運送量及未實現訂單數的最小值做為實現的訂單數量。

圖形右下角是產品庫存次系統，相關 Dynamo 方程式如下：

N Inventory = 200
 L Inventory = +dt*ProductInRate
 -dt*ProductOutRate
 R ProductInRate = OrderFulfillRate
 R ProductOutRate =
 MIN(SaleFunction,Inventory)

產品庫存的初始值設為 200，其單位為 piece，代表初期產品的庫存數量。產品入庫 ProductInRate 即是訂單實現的數量，單位為 piece/day，產品銷貨 ProductOutRate 單位為 piece/day，其會取銷售函數 SaleFunction 與現有產品庫存量 Inventory 的最小值，避免庫存量成為負值。但當銷售函數大於產品庫存量時，即代表部分銷售需求無法滿足。現有產品庫存量的單位為 piece。銷售函數目前採用 Graph 函數來模擬銷售的變化情形。

4.3 模擬結果

從圖 4.3.1 的結果可以發現，在第 10 天左右產品庫存會降為 0，我們進一步輸出產品銷貨 ProductOurRate 與銷售函數 SaleFunction 來比較，了解銷售需求的滿足情形。

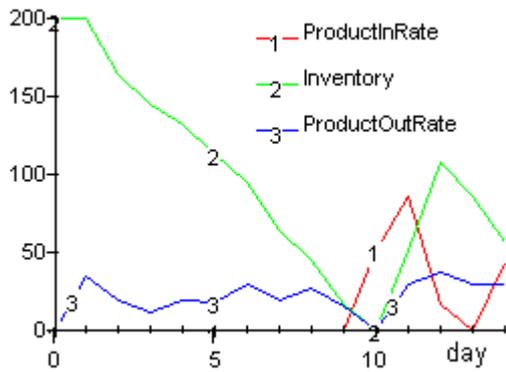


圖 4.3.1 產品入庫、產品庫存、產品銷貨

從圖 4.3.2 發現在第 10 天的確有無法滿足銷售需求的情形發生。此種情形在 90% 服務水準的範圍尚屬可以容忍，倘若要減少無法滿足銷售需求的發生次數，可以調高服務水準來改善，但這也意味著較多的存貨成本，決策者必須在兩者間衡量。

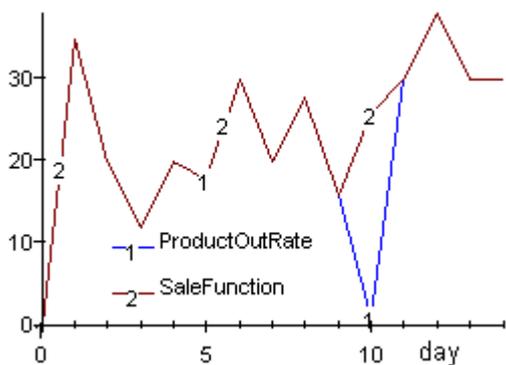


圖 4.3.2 銷貨與銷售需求

5. 結論

本研究提出數個跟資料保留有關的系統動力學線流圖建構技巧，同時以銷售及庫存管理作業為例，提出實際應用的示範。然而在其他領域的系統動力學模型建構中，也可以運用本研究所提之建構技巧。

系統動力學在不同領域應用時，由於領域間的差異性，對於模型的建構有時會有一些特殊的需求。本研究所提之建構技巧，僅解決在不同領域應用系統動力學時所遇問題的一部分，後續可以持續研究各個領域在系統動力學模型應用的其他建構技巧。

參考文獻

[1] Forrester, J. W., Industry Dynamics,

Cambridge: MIT press, 1961.

[2] Lyneis, J. M., Corporate Planning and Policy Design, Waltham, MA, Pegasus Communication, 1980.

[3] Sterman, J. D., Business Dynamics: Systems Thinking and Modeling for a Complex World, McGraw-Hill/Irwin, 2000.

[4] Krajewski, L. J. and Ritzman, L. P., Operations Management: Strategy and Analysis, Addison-Wesley, 2001.