

Mean Time Between Failure

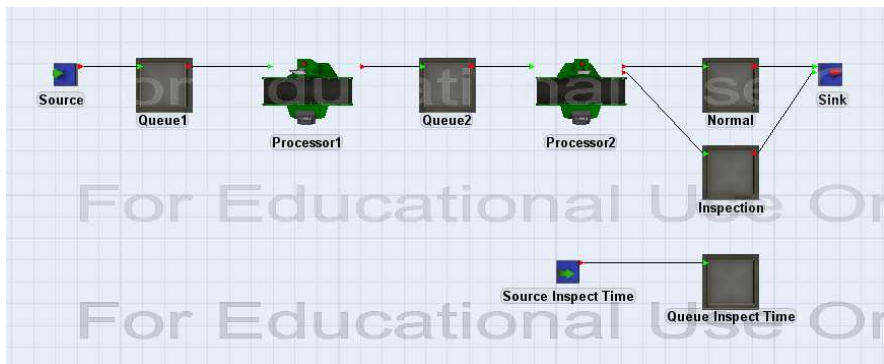
W. M. Song 桑慧敏
Tsing Hua Univ. 清華大學

2015.12.09

Model Description

- 每個 item 有兩個屬性:
 - (1) 良品/不良品 (0/1)
 - (2) 翹曲值 (Warpage),
- 兩個機台每隔一段時間會有機台故障的狀況, 故障時雖可繼續加工, 但其不良率及翹曲值增加
 - 不良率: $0.05 \rightarrow 0.2$
 - 翹曲值: $\text{normal}(0, 0.1) \rightarrow \text{normal}(0.5, 0.1)$
- 設置檢查站, 每隔 120 分鐘取 10 個檢驗, 並記錄其兩個屬性到 Global Table

Flexsim Model



Global Tables

- **ItemLabel** : 記錄每個 Item 的三個 label
- **BreakState** : 顯示機台狀態及記錄故障次數
- **InspectState** : 顯示檢查站狀態及記錄檢驗次數
- **Inspect01** : 記錄每次檢查的十個 item 的良品/不良品 (0/1)
- **InspectWarpage** : 記錄每次檢查的十個 item 的翹曲值
- **BreakTime** : 記錄兩機台故障時間及修好的時間

Create Global Table

- Tools → Global Tables → Add

ItemLabel

- Row: 5000 , Columns: 4 , Clear on Reset

Global Table - ItemLabel

Name: ItemLabel Rows: 100000 Columns: 3.00 Clear on Reset

	Uniform	Index 01	Warpage	
Row 1	0.29	0.00	-0.11	
Row 2	0.91	0.00	-0.08	
Row 3	0.65	0.00	-0.08	
Row 4	0.69	0.00	-0.11	
Row 5	0.03	1.00	0.12	
Row 6	0.44	0.00	-0.01	
Row 7	0.41	0.00	0.10	
Row 8	0.70	0.00	-0.13	
Row 9	0.68	0.00	0.12	
Row 10	0.82	0.00	-0.05	
Row 11	0.14	0.00	-0.07	
Row 12	0.07	0.00	0.06	
Row 13	0.65	0.00	0.02	
Row 14	0.37	0.00	0.04	

Create Global Table

- Tools → Global Tables → Add

BreakState

- Row: 2 , Columns: 2 , Clear on Reset

Global Table - BreakState

Name: BreakState Rows: 2.00 Columns: 2.00

	State	Round
Processor 1	0.00	0.00
Processor 2	0.00	0.00

Add Table to MTEI Close

InspectState

- Row: 1 , Columns: 2 , Clear on Reset

Global Table - InspectState

Name: InspectState Rows: 1.00 Columns: 2.00

	Inspect State	Inspect Round
	0.00	0.00

Add Table to MTEI Close

Create Global Table

- Tools → Global Tables → Add

Inspect01

- Row: 10 , Columns: 1000 , Clear on Reset

Global Table - Inspect01

Name: Inspect01 Rows: 10.00 Columns: 1000.00 Clear on Reset

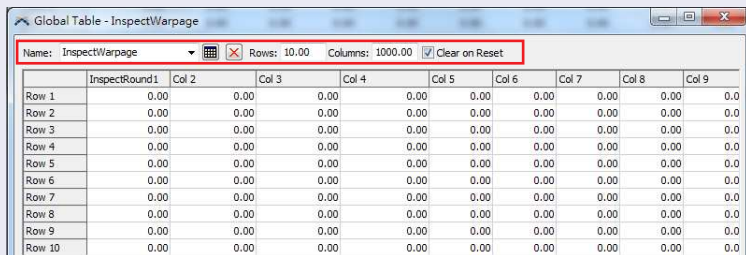
	InspectRound1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10
Row 1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Create Global Table

- Tools → Global Tables → Add

InspectWarpage

- Row: 10 , Columns: 1000 , Clear on Reset



Global Table - InspectWarpage

Name: InspectWarpage Rows: 10.00 Columns: 1000.00 Clear on Reset

	InspectRound1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9
Row 1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Create Global Table

- Tools → Global Tables → Add

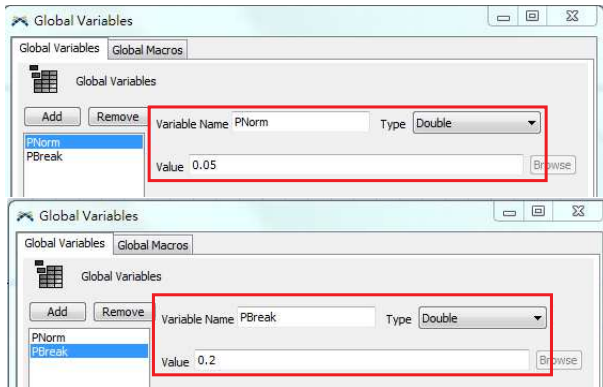
BreakTime

- Row: 100 , Columns: 4 , Clear on Reset

	Processor 1 Break	Processor 1 Repair	Processor 2 Break	Processor 2 Repair
Row 1	0.00	0.00	0.00	0.00
Row 2	0.00	0.00	0.00	0.00
Row 3	0.00	0.00	0.00	0.00
Row 4	0.00	0.00	0.00	0.00
Row 5	0.00	0.00	0.00	0.00
Row 6	0.00	0.00	0.00	0.00
Row 7	0.00	0.00	0.00	0.00
Row 8	0.00	0.00	0.00	0.00
Row 9	0.00	0.00	0.00	0.00
Row 10	0.00	0.00	0.00	0.00
Row 11	0.00	0.00	0.00	0.00
Row 12	0.00	0.00	0.00	0.00

Create Global Variables

- Tools → Global Variables → Add
 - Variable Name: PNorm , Type: Double , Value=0.05
 - Variable Name: PBreak , Type: Double , Value=0.2



Create MTBF MTTR

- Tools → MTBF MTTR → Add

Break1

The image shows a software interface for configuring MTBF and MTTR parameters. The main window is titled "MTBF/MTTR Parameters Window" and contains a form for "Break1". The form has several fields: "First Failure Time" (30), "MTBF" (weibull(0.0, 1800, 4.0, 0)), "MTTR" (120), and "Down Function" (Do nothing). Below these are "On Break Down" and "On Repair" actions, both set to "Set Color (individual)".

Two code windows are also visible, showing the logic for the break and repair actions. The first window, titled "Break1 - On Break Down", contains the following code:

```
1 treenode current = ownerobject(c);
2 treenode members = var_s(current, "members");
3 treenode involved = parnode(i);
4
5 /**Set Color (individual)**
6 /** \nColor: *
7 /**/colororange/**/ (involved);
8 /**\n\n*/
9 //change state from 0 to 1
10 settablenum("BreakState",1,1,1);
11 //record break round
12 settablenum("BreakState",1,2,gettablenum("BreakState",1,2)+1);
13 //record break time
14 settablenum("BreakTime",gettablenum("BreakState",1,2),1,time());
15
```

The second window, titled "Break1 - On Repair", contains the following code:

```
1 treenode current = ownerobject(c);
2 treenode members = var_s(current, "members");
3 treenode involved = parnode(i);
4
5 /**Set Color (individual)**
6 /** \nColor: *
7 /**/colorgreen/**/ (involved);
8 /**\n\n*/
9 //change state from 1 to 0
10 settablenum("BreakState",1,1,0);
11 //record repair time
12 settablenum("BreakTime",gettablenum("BreakState",1,2),2,time());
13
```

Create MTBF MTTR

- Tools → MTBF MTTR → Add

Break2

The screenshot displays the configuration for a 'Break2' event in a simulation software. The 'MTBF/MTTR Parameters Window' on the left shows the following settings:

- Name: Break2
- Members: Breakdowns
- First Failure Time: 30
- MTBF: weibul(0.0,1800, 4.0, 0)
- MTTR: 120
- Down Function: Do nothing
- Resume Function: Resume object
- On Break Down: Set Color (individual)
- On Repair: Set Color (individual)

Two code windows on the right show the logic for the 'Break2 - On Break Down' and 'Break2 - On Repair' events:

```
Break2 - On Break Down
1 treenode current = ownerobject(c);
2 treenode members = var_s(current,"members");
3 treenode involved = parnode(1);
4
5 /**Set Color (individual)*/
6 /** \nColor: */
7 /**/colored/**/(involved);
8 /**\n\n*/
9 //change state from 0 to 1
10 settablenum("BreakState",2,1,1);
11 //record break round
12 settablenum("BreakState",2,2,gettablenum("BreakState",2,2)+1);
13 //record break time
14 settablenum("BreakTime",gettablenum("BreakState",2,2),3,time());
15
```

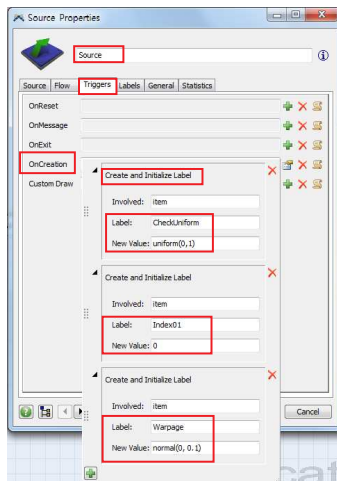
```
Break2 - On Repair
1 treenode current = ownerobject(c);
2 treenode members = var_s(current,"members");
3 treenode involved = parnode(1);
4
5 /**Set Color (individual)*/
6 /** \nColor: */
7 /**/colorgreen/**/(involved);
8 /**\n\n*/
9 settablenum("BreakState",2,1,0);
10
11 settablenum("BreakTime",gettablenum("BreakState",2,2),4,time());
12
```

Set Inter-Arrivaltime and Process Time

- Source \rightarrow Inter-Arrivaltime: exponential(0, 2, 0)
- Source Inspect Time \rightarrow Inter-Arrivaltime: 120
- Processor1 \rightarrow Process Time: 2
- Processor2 \rightarrow Process Time: 2


Create Labels

- Source → Triggers → OnCreation
→ Create and Initialize Label
 - CheckUniform → $\text{uniform}(0, 1)$
 - Index01 → 0
 - Warpage → $\text{normal}(0, 0.1)$



Set Breakdown and Label

- Processor1 → Breakdowns → This object is a member of the following MTBF MTTR's: → Add → Break1
- Processor1 → Triggers → OnEntry
- Processor1 → Triggers → OnExit



The image shows two overlapping code editor windows. The top window is titled "/Processor1 - OnEntry" and contains the following code:

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 //change label number
7 if(gettablenum("BreakState",1,1)==1)
8 {
9     setlabelnum(item, "Warpage", normal(0.5, 0.1));
10 }
11
```

The bottom window is titled "/Processor1 - OnExit" and contains the following code:

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 settablenum("ItemLabel",getoutput(current)+1,3,getlabelnum(item, "Warpage"));
```

In both windows, the title bar text is highlighted with a red box.

Set Breakdown and Label

- Processor2 → Breakdowns → This object is a member of the following MTBF MTRR's: → Add → Break2
- Processor2 → Triggers → OnEntry
- Processor2 → Triggers → OnExit

```
Processor2 - OnEntry
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 if(gettablenum("BreakState",2,1)==0)
7 {
8     if(getlabelnum(item, "CheckUniform") < PHorm )
9     {
10         setlabelnum(item, "Index01", 1);
11     }
12 }
13 else
14 {
15     if(getlabelnum(item, "CheckUniform") < PBreak)
16     {
17         setlabelnum(item, "Index01", 1);
18     }
19 }
20

Processor2 - OnExit
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 int k = getoutput(current)+1;
7 settablenum("ItemLabel", k, 1, getlabelnum(item, "CheckUniform"));
8 settablenum("ItemLabel", k, 2, getlabelnum(item, "Index01"));
9
```


Set Flow to Inspection

- Processor2 → Flow → Send To Port

The screenshot shows the 'Processor2 Properties' dialog box with the 'Flow' tab selected. The 'Send To Port' button is highlighted with a red box. Below it, the 'Custom Code' field is also highlighted with a red box. A separate window titled '/Processor2 - Send To Port' is open, showing a code editor with the following code:

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4
5
6 if(gettablenum("InspectState",1,1)==0)
7 {
8     return 1;
9 }
10 else
11 {
12     return 2;
13 }
14
```

The code block from line 6 to 13 is highlighted with a red box.

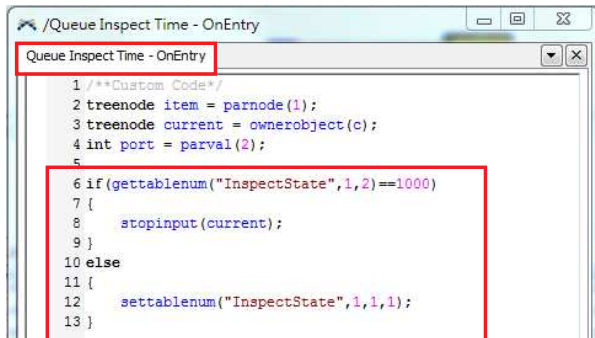
The screenshot shows the 'Processor2 Properties' dialog box with the 'General' tab selected. The 'General' tab is highlighted with a red box. In the 'Ports' section, the 'Output Ports' list is highlighted with a red box, showing:

- 1: Normal
- 2: Inspection

The 'Output Ports' list is also highlighted with a red box.

Set Inspect State

- Queue Inspect Time → Triggers → OnEntry



```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 if(gettablenum("InspectState",1,2)==1000)
7 {
8     stopinput(current);
9 }
10 else
11 {
12     settablenum("InspectState",1,1,1);
13 }
```

Inspection

- Inspection → Triggers → OnEntry



```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 int k = getinput(current);
7 double q = k - 10*(gettablenum("InspectState",1,2));
8 settablenum("Inspect01", q, gettablenum("InspectState",1,2)+1, getlabelnum(item,"Index01"));
9 settablenum("InspectWarpage", q, gettablenum("InspectState",1,2)+1, getlabelnum(item,"Warpage"));
10
11 if(q == 10)
12 {
13     settablenum("InspectState",1,1,0);
14     //record inspect round
15     settablenum("InspectState",1,2,gettablenum("InspectState",1,2)+1);
16 }
17 //change color
18 if(gettablenum("InspectState",1,1)==1)
19 {
20     setcolor(current,255,0,0);
21 }
22 else
23 {
24     setcolor(current,127,127,127);
25 }
26 if(gettablenum("InspectState",1,2)==1000)
27 {
28     stopoutput(node("/Source", model()));
29 }
30
```