A Quality Model for Responsibility Assignment in Repetitive Flow Schooling and Curriculum Processes Associated with a Repairable System of Different Failure Modes

Faten S. M. Abdel-Hameed  
Bahrain Teachers College,  
Mathematics, Science and ICT AG  
University of Bahrain, Bahrain

Salah A. A. Emara  
Studies and Research Unit  
University College of Bahrain  
Kingdom of Bahrain salahemara

Abstract

This paper proposes the use of responsibility modeling as a tool to support the designing of efficient operations of repetitive flow schooling and curriculum processes associated with repairable system of different failure modes. The objective is to develop a deeper understanding of system behaviour as a function of tasks. Based on the assignment of appropriate duration of tasks, we assign daily responsibilities, and expect the line of production to be operating seven hours a day to achieve the production of a number of completed tasks. To find the expected completion line for each activity, we provide twenty-one estimate times for twenty-one employees, and apply task assignment to obtaining the optimal time for completed non-repetitive tasks. Algebraic formulas and the supplementary variable method are used to measure and analyze the different failure modes and repair the tasks of each workstation in the system.

1. Introduction

Repetitive processes in any organization are described by a mathematical model that uses simplified assumptions of the assignment networks model with time that enables the realities of the organizational systems. These same methods can also be used if it is desired to perform repetitive processes associated with repairable system of different failure modes for educational and curricular activities. We assign twenty-one personnel (employees) to twenty-one jobs (tasks) within the distribution area of an organizational system. The models used to measure the effectiveness of a particular set of assignments of total time to perform a set of operations, and to determine the production rate of each workstation and bottleneck workstation, complementing those results obtained in ([5], [6]). Mathematical formulas are used to measure the availability and unavailability associated with repairable system of different failure modes for each workstation.

2. Activity Time

Here, we discuss the detailed design of repetitive flow processes, which focuses on balancing the assignment of work at workstations (WS), and explains the applicability of repetitive flow principles to disconnected flow processes [8], [9], [10], [11]. We apply the assignment model for twenty-one employees to be assigned to twenty-one tasks and a time \( t_{ij} \) that is associated with employee \( i \) performing task \( j \). Our goal is to minimize the total time of assigning employees to tasks so that each employee is assigned one job and each job is performed. The time required to complete each and every activity is expressed in seconds. Table 1 represents the total enumeration of all matrix possibilities of assigning twenty-one employees on twenty-one tasks. Now we apply the following steps on Table 1 to reach the time-efficient assignment:

**Step 1:** Subtract the minimum entry in each row from all the entries on that row.

**Step 2:** Subtract the minimum entry in each column of task-opportunity time matrix from all the entries of that column.

**Step 3:** Check for optimality: Draw the minimum number of horizontal and vertical lines to cover all zeros in the rows and columns. This can be done in twenty-one lines, or we apply the shuffling for the tasks among the employees, in order get the required time matrix, i.e., we obtain the minimum total time of assigning twenty-one employees to twenty-one tasks so that each employee is assigned one task as given in the diagonal of the following time-matrix as shown in Table 2.
Tables 1 and 2.

The optimal time (t) for activities as obtained from assigned with an optimal time and predecessor. Each day. At this stage of study each activity will be line of production to be operating 420 minutes to produce at least 240 tasks altogether. We expect one day with one task per person per prescribed time, we assign the 21 daily tasks to the 21 employees for rest of employees and the corresponding tasks. Now, for (Emp 2, Task 19) is 30 sec, and similarly for the optimal time for (Emp 1, Task 1) is 30 sec, task j, where i, j = 1 to 21. From Table 1, one can see that the optimal time for (Emp 2, Task 19) is 30 sec, and similarly for the rest of employees to 21 tasks.

Table 2. Time-efficient assignment of 21 employees to 21 tasks (Emp = Employee and T = task)

<table>
<thead>
<tr>
<th>Emp</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

From Table 2 the Employee i will be assigned to the task j, where i, j = 1 to 21. From Table 1, one can see that the optimal time for (Emp 1, Task 1) is 30 sec, for (Emp 2, Task 19) is 30 sec, and similarly for the rest of employees and the corresponding tasks. Now, we assign the 21 daily tasks to the 21 employees for one day with one task per person per prescribed time, to produce at least 240 tasks altogether. We expect the line of production to be operating 420 minutes each day. At this stage of study each activity will be assigned with an optimal time and predecessor. Table 3 presents the activities, the predecessors and the optimal time (t) for activities as obtained from Tables 1 and 2.

Table 3. Task optimal time for activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Predecessor</th>
<th>Optimal time (t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>30</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>30</td>
</tr>
<tr>
<td>C</td>
<td>B</td>
<td>33</td>
</tr>
</tbody>
</table>

We want to assemble a line to produce at least 240 completed tasks per day and expect the line to be operating 420 minutes each day.

(Working) day = operating time each day = 420 min = 25, 200 sec

Maximum cycle time = Cmax

= 1/ (minimum desired production rate)

= 1/ (240 completed tasks per day)

= 1 day

= 240 completed tasks

= 25, 200 seconds

= 105 sec/ completed tasks

Cmax = 105 second/ completed tasks

Theoretical minimum number of workstations = T/Cmax = 673/105 = 6.4095 ≈ 7 WS
tasks must precede others. This is shown in the Figure1.

![Diagram of tasks](image.png)

Figure 1. Procedure diagram for the activities

For each task, add up the task time for that task and all the tasks times which follow it directly and indirectly. This value is called the Positional Weight for the task. PW(A) denotes the positional weight for task A, and it is computed from the sum of the task times for the tasks: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, i.e., PW(A) = 620. Similarly we obtain the positional weights for all the remaining tasks as, PW(B) = 590, PW(C) = 530, PW(D) = 497, PW(E) = 380, PW(F) = 403, PW(G) = 397, PW(H) = 401, PW(I) = 396, PW(J) = 389, PW(K) = 246, PW(L) = 104, PW(M) = 101, PW(N) = 106, PW(O) = 126, PW(P) = 104, PW(Q) = 101, PW(R) = 120, PW(S) = 70, PW(T) = 86, PW(U) = 23.

Here, we arrange each task according to its positional weight in descending order by selecting the task with largest positional weight and assign it to be the first workstation, and next we select the task with next largest positional weight, then repeat the same for the rest of tasks. i.e., we select A first because it has the largest positional weight, and assign it to workstation WS1 (workstation 1), the second highest (largest) positional weight will be activities: B, C, D, F, H, … etc.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Optimal time (t)</th>
<th>Positional weight PW</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>30</td>
<td>620</td>
</tr>
<tr>
<td>B</td>
<td>30</td>
<td>590</td>
</tr>
<tr>
<td>C</td>
<td>33</td>
<td>530</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>497</td>
</tr>
<tr>
<td>F</td>
<td>37</td>
<td>403</td>
</tr>
<tr>
<td>H</td>
<td>35</td>
<td>401</td>
</tr>
<tr>
<td>G</td>
<td>31</td>
<td>397</td>
</tr>
<tr>
<td>I</td>
<td>30</td>
<td>396</td>
</tr>
<tr>
<td>J</td>
<td>23</td>
<td>389</td>
</tr>
<tr>
<td>E</td>
<td>20</td>
<td>380</td>
</tr>
<tr>
<td>K</td>
<td>34</td>
<td>246</td>
</tr>
<tr>
<td>O</td>
<td>40</td>
<td>126</td>
</tr>
<tr>
<td>R</td>
<td>50</td>
<td>120</td>
</tr>
<tr>
<td>N</td>
<td>20</td>
<td>106</td>
</tr>
<tr>
<td>L</td>
<td>18</td>
<td>104</td>
</tr>
<tr>
<td>P</td>
<td>18</td>
<td>104</td>
</tr>
<tr>
<td>M</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>Q</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>T</td>
<td>86</td>
<td>86</td>
</tr>
<tr>
<td>S</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>U</td>
<td>23</td>
<td>23</td>
</tr>
</tbody>
</table>

Now, we obtain the workstations as:

**WS1:** A B C
Task time = 30 + 30 + 33 = 93 < Cmax = 105

**WS2:** D F H
Task time = 15 + 37 + 35 = 87 < Cmax = 105

**WS3:** G I J E
Task time = 31 + 30 + 23 + 20 = 104 < Cmax = 105

**WS4:** K O
Task time = 34 + 40 = 74 < Cmax = 105

**WS5:** R N L
Task time = 50 + 20 + 18 = 88 < Cmax = 105

**WS6:** P M O
Task time = 18 + 15 + 15 = 48 < Cmax = 105

**WS7:** T
Task time: 86 < Cmax = 105

**WS8:** S U
Task time: 70 + 23 = 93 < Cmax = 105

The workstation with longest time (C = 104 second/ task) is called the bottleneck work-station, obtained at WS7, and the actual production rate for the production line is given by

\[
p = \frac{1}{\frac{C}{104 \text{ second}}} = \frac{1}{104 \text{ task}}
\]
\[
\text{day} = 25,200 \text{ seconds} \\
\text{second} = \frac{1}{25,200} \text{ day} \\
(\text{since the operation time is } 420 \text{ minutes }= 420 \times 60 = 25,200 \text{ sec}).
\]

From (4) and (5) we get the actual rate
\[
P = \frac{1}{C} = \frac{1}{1 \text{ day}} = \frac{1}{25,200 \text{ task}} = 104 \text{ task/ day}
\]
\[
= 242.3076 \text{ tasks/day},
\]

Therefore, the actual rate of production is \( p = 242.3076 \text{ tasks/day} \), which is greater than the target production rate 240 tasks/day, since

Actual cycle time \( C < \text{max cycle time } C_{\text{max}} \)

3. The Efficiency or Percentage Balance of the Production Line

The efficiency or percentage balance of the production line is obtained by:

\[
\text{Efficiency (percentage balance) of a line} = \left[ \frac{T}{N \times C} \right] \times 100 \%
\]

Where, \( N \) is the number of actual workstations. In our case: \( N = 8 \) WS’s, \( C = 104 \text{ second/ task} \), and \( T = 1232 \text{ seconds} \).

\[
\therefore \text{Efficiency for the production line} = \left[ \frac{673}{8 \times 104} \right] \times 100 \% = 80.89 \%.
\]
i.e., the efficiency or percentage balance of the line equals 80.89%, and the capacity wasted (idleness) is equal to 19.11%.

Here, we evaluate the efficiency and idleness for each work station: \( \text{WS}_i \) is working at 100%, while efficiency \( \text{WS}_1 = 88.57 \% \), efficiency \( \text{WS}_2 = 82.86 \% \), efficiency \( \text{WS}_3 = 70.48 \% \), efficiency \( \text{WS}_4 = 83.81 \% \), efficiency \( \text{WS}_5 = 71 \% \), and efficiency \( \text{WS}_6 = 81.90 \% \), \( \text{WS}_7 = 80.89 \% \), \( \text{WS}_8 = 88.57 \% \) with idleness of: 11.43%, 17.14%, 29.52%, 16.19%, 54.29%, 18.1%, and 11.43% for each workstation, respectively.

4. Repairable Tasks with Different Failure Modes for WS1

This section presents mathematical model and analysis of repair-replacement for WS1, and with three failure modes: normal, partial failure and total failure. Figure 2 shows the state diagram of a single repairable component with failure modes for WS1.

Similar repairable systems are applied for the rest of WS’s, the related literature may be found in ([1], [2], [3], [4], [7], [12], [13]).

\( \lambda, \beta, \text{and } \mu \) are the constant failure, repairmen-availability and replacement rates, respectively.

Figure 2. State diagram for a single repairable component with failure modes for WS1

We apply the supplementary variable method to analyze the model; the following differential equations can be obtained under the following assumptions:

\[
\begin{align*}
\frac{d P_0(t)}{dt} &= -(\lambda + \beta) P_0(t) + \beta P_1(t) + \lambda P_2(t) \\
\frac{d P_1(t)}{dt} &= \lambda P_0(t) - (\lambda + \beta) P_1(t) + \mu P_2(t) \\
\frac{d P_2(t)}{dt} &= \beta P_0(t) + \lambda P_1(t) - (\mu + \lambda) P_2(t)
\end{align*}
\]

The initial conditions are as follows:

\( P_0(0) = 1, P_1(0) = P_2(0) = 0 \).

Applying the Laplace transforms to (6) – (9) we obtain

\[
\begin{align*}
s P_0(s) - P_0(0) &= -(\lambda + \beta) P_0(s) + \beta P_1(s) + \lambda P_2(s) \\
s P_1(s) - P_1(0) &= \lambda P_0(s) - (\lambda + \beta) P_1(s) + \mu P_2(s) \\
s P_2(s) - P_2(0) &= \beta P_0(s) + \lambda P_1(s) - (\mu + \lambda) P_2(s)
\end{align*}
\]

On applying the initial conditions (9) to equations (10)- (11)

\[
\begin{align*}
(s + \lambda + \beta) P_0(s) - \beta P_1(s) - \lambda P_2(s) &= 1 \\
- \lambda P_0(s) + (s + \lambda + \beta) P_1(s) - \mu P_2(s) &= 0 \\
- \beta P_0(s) - \lambda P_1(s) + (s + \mu + \beta) P_2(s) &= 0
\end{align*}
\]

The solutions for (8)- (10) are

\[
\begin{align*}
P_0(s) &= \frac{\Delta_0}{\Delta} \\
P_1(s) &= \frac{\Delta_1}{\Delta} \\
P_2(s) &= \frac{\Delta_2}{\Delta}
\end{align*}
\]

where

\[
\Delta_0 = s^2 + s(2\lambda + \beta + \mu) + \mu\beta + \lambda^2 + \lambda\beta,
\]

\[
\Delta_1 = s^2 + s(2\lambda + \beta + \mu) + \mu\beta + \lambda^2 + \lambda\beta,
\]

\[
\Delta_2 = s^2 + s(2\lambda + \beta + \mu) + \mu\beta + \lambda^2 + \lambda\beta
\]

Copyright © 2011, Infonomics Society
\[ \Delta_1 = s \lambda + \mu \lambda + \beta^2 + \mu \beta, \]
\[ \Delta_2 = s \beta + \beta \lambda + \beta^2 + \lambda^2, \]
\[ \Delta = s^3 + s^2 (3 \lambda + 2 \beta + \mu) + s (\mu \lambda + \lambda \beta + 2 \mu \beta + 3 \lambda^2 + \beta^2) \]

The availability in the steady state of the model is
\[ A = \lim_{s \to 0} s [P_0(s) + P_1(s)] = \]
\[ \frac{2 \mu \beta + 2 \lambda^2 + \lambda \beta + \mu \lambda}{\mu \lambda + \lambda \beta + 2 \mu \beta + 3 \lambda^2 + \beta^2} \]

and the unavailability of the model is
\[ UA = \lim_{s \to 0} s P_2(s) = \]
\[ \frac{\lambda \beta + \beta^2 + \lambda^2}{\mu \lambda + \lambda \beta + 2 \mu \beta + 3 \lambda^2 + \beta^2} \]

Special case: If \( \lambda = \mu = \beta = 0.5 \), then \( A + UA = 1 \).

4. Conclusion

The main focus of the study was to describe the quality methodology and algorithm for solving a repetitive processes problem based on the assignment of appropriate duration of responsibilities for school and curriculum improvement. The interaction between successive activities was obtained in order to control those processes for the sake of achieving the desired efficiencies for the production line. We applied the repairable system with different failure modes and obtained the measure of the availability and the unavailability of the work-stations.

5. References


