Journal of Engineering Science and Technology
Vol. 5, No. 4 (2010) 447 - 456

AN ARTIFICIAL NEURAL NETWORK APPROACH – PERFORMANCE MEASURE OF A RE-ENTRANT LINE IN A REFLOW SCREENING OPERATION

SURESH KUMAR*, A. ARUNAGIRI

Department of Electrical and Electronics Engineering Technology,
Yanbu Industrial College, Yanbu Al Sinaiyah, Kingdom of Saudi Arabia
*Corresponding Author: vpsureshkumar@yahoo.com

Abstract

This paper presents an artificial neural network (ANN) method applied to a multistage re-entrant line system. Generally, queuing networks adopt analytical methods or use simulation packages to determine their performance measure. The contribution of this paper is the development of an alternate solution method using ANN approach to determine performance measure namely the total cycle time for a Reflow Screening (RS) operation in a semiconductor assembly plant. Performance measure of an operation is an important aspect in management decision making. In order to validate the proposed method, comparison results were made using the analytical method based on mean value analysis (MVA) technique for the re-entrant line and with some historical data collected from the operation. In this paper, Back Propagation Network (BPN) learning algorithm is proposed for the computation of the total cycle time with respect to the number of lots circulating in the system. Extensive training and testing of the proposed ANN method is performed which enables the BPN model to be used to determine the required total cycle time.

Keywords: Mean value analysis, Artificial neural network, Reflow screening.

1. Introduction

The Reflow Screening (RS) is a process in the semiconductor back end manufacturing industry. The RS operation is not part of the main production line, but performed separately in a reliability laboratory environment. The purpose of the reflow operation is to mount chips on printed circuit board, in the electronic assembly plants. However, in the semiconductor industry, the reflow operation

Nomenclatures

b_{il}	Buffers at stage *i* for *l* th cycle
i	Index of stage, $i = 1,\ldots, M$
j	Index of the previous stage where the lots are arriving from
k	Number of lots in the system, $k = 1,\ldots, N$
l	Number of screening cycles $l = 1, \ldots, P$
$L_{il}(k)$	Mean number of lots waiting at b_{il} with k number lots in the system
$W_{il}(k)$	Mean waiting time of a lot in b_{il} when the system has *k* number of lots
$W_T(k)$	Total cycle time of *k* number of lots going through all buffers

Greek Symbols

$\lambda(k)$	Throughput rate when the system has *k* lots
μ_{il}	Mean processing rate of a lot at buffer b_{il}

becomes a screening process to remove defective units residing in the lots. This is known as reflow screening [1,2].

The RS operation is a 5 stage operation with feedback (re-entrant) and the re-entrant happens at the second stage where lots that need subsequent round of screening are re-routed from the fifth stage. However, to facilitate this study, the operation is conditioned such that it has either single re-entrant line or without re-entrant line. For illustration purpose, if *k* number of lots enter the system, they will either go for one cycle of screening (without re-entrant) or two cycles of screening (single reentrant). The block diagram of the operation is depicted in Fig. 1.

The motivation of this research is to develop a solution method using artificial neural network (ANN) on a queueing network system to compute the total cycle time for a given number of lots which circulates in the system. The validity of this approach is verified using the analytical computation based on mean value analysis (MVA) which is widely used in the queuing network applications. The total cycle time (makespan) is defined as the time taken to complete the entire process sequence for a given number of lots.

Artificial neural networks (ANNs) apply a different way from traditional computing methods to solve problems. Basically, conventional computers use algorithmic approach which means that specific steps have to be defined for the computer in order to solve a specific problem. That means, traditional computing methods can only solve the problems that we have already understood and knew how to solve. Due to ANNs ability to adapt, learn, generalize, cluster or organise data, it has in some way become powerful tool to solve problems that we do not exactly know how to solve. There are so many structures of ANNs including, percepton, adaline, madaline, kohonen, back propagation and many others. Back propagation network (BPN) is the most commonly used, as it is relatively very simple to implement and effective. In this work, BPN approach is adopted to develop the proposed solution method.

The remainder of this paper is organized as follows. Section 2 deals with the analytical model formulation for the RS operation. Section 3 briefly introduces the back propagation network (BPN), its architecture, training algorithm and recognition phase. The implementation of BPN to determine the total cycle time is also included. Section 4 presents computation and comparative results of the system being studied by the proposed method and section 5 concludes this paper.

2. Analytical Model Formulation

The RS operation has five stages serially connected with buffers at every stage. Figure 1 illustrates the block diagram of the RS operation. When lots arrive at any stage from the previous one, the lots will queue in their respective buffer before they are processed.

Buffers are temporary waiting place before the lots are processed. In this case the lots are processed on a first come first serve basis. Every stage acts as a single server system with its own service rate. After the last stage, lots are fed back into the second stage and will undergo repeated cycles of reflow screening as required by the lots.

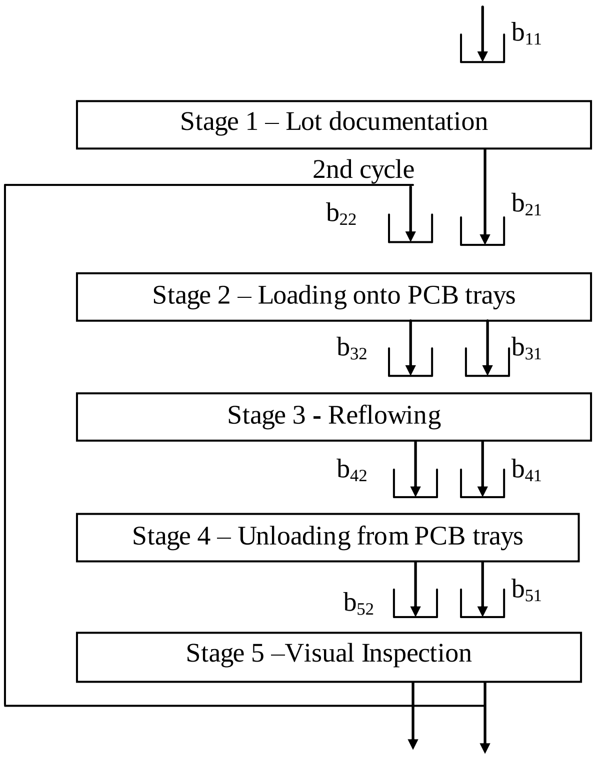

Fig. 1. Block Diagram of the Reflow Screening Operation.

2.1. System assumption

In order to facilitate the development of the proposed method for both analytical and BPN approach, the RS operation is described as a queuing system. However, to avoid mathematical complexity, several assumptions are considered as follows:

- In order to utilize the MVA equations, the RS operation is considered as a closed network system where there is no continuous external arrival. The number of lots in the system (WIP) is kept constant. Consequently, the operation can be described as a closed and product form queuing system.

Once the adaptation is done, the performance measure of the total cycle time is computed.

- All stages have their own constant mean process time. Any small changes in the lot size does not affect the process time. If the changes are large enough, then the entire lot is removed from the operation and immediately replaced by a similar lot. The lots have standard size too.

It is also assumed that the operation is free from disturbances, interruptions or additional setup time which can add to the total cycle time.

2.2. MVA equations

MVA equations are based on the arrival theorem and Little's Law [3]. Arrival theorem defined by Reiser and Lavenberg and Gross and Harris [4,5] states that the queue length observed by an arriving lot to a workstation is the same as the overall queue length seen by an outside lot when system's lot population is less by one. Little's Law gives a relationship between mean queue length with the arrival rate and mean waiting time [6,7]. Thus the mean waiting time of lots in buffer b_{il} is

$$W_{il}(k) = \frac{L_{il}(k-1)}{\mu_{il}} + \frac{1}{\mu_{il}} \tag{1}$$

The total cycle time for k number of lots in the system having P number of re-entrant lines is derived as

$$W_T(k) = \sum_{i=1}^{M} \sum_{l=1}^{P} W_{il}(k) \tag{2}$$

Equation (2) indicates that all k lots will go through all the buffers in its route. Applying little's law for the population of lots in the system; the throughput rate is obtained as

$$\lambda(k) = \frac{k}{W_T(k)} \tag{3}$$

The mean queue length at every buffer is

$$L_{il}(k) = \lambda(k)\, W_{il}(k) \tag{4}$$

The initial conditions for the iterations are

$$L_{il}(0) = 0\,,\ W_{il}(0) = 0\,,\ \lambda(0) = 0 \tag{5}$$

The total cycle time is computed by an iterative method with the given initial conditions. The analytical model shown above indicates that all k number of lots will pass through every buffer before exiting from the system. In this study the nature of the re-entrant is purely deterministic [8].

For experimenting with the re-entrant model, the input parameters are the service time of each stage, obtained from a real reflow screening operation [8,9]. The time taken by each stage to process the lots is measured over period of time and averaged. For computation purpose, the values are considered as follow.

$$\frac{1}{\mu_{11}} = 10 \quad \text{minutes}$$

$$\frac{1}{\mu_{21}} = \frac{1}{\mu_{22}} = 2 \quad \text{minutes}$$

$$\frac{1}{\mu_{31}} = \frac{1}{\mu_{32}} = 15 \quad \text{minutes}$$

$$\frac{1}{\mu_{41}} = \frac{1}{\mu_{42}} = 2 \quad \text{minutes}$$

$$\frac{1}{\mu_{51}} = \frac{1}{\mu_{52}} = 5 \quad \text{minutes}$$

3. ANN Model Formulation

3.1. BPN architecture

The most common BPN architecture is presented in Fig. 2 [10]. The architecture has three layers, namely, input, hidden and output layers. Other implementations may have several hidden layers. Back Propagation Network contains one or more layers each of which are linked to the next layer. The first layer is called input layer which meets the initial input and so do the last one output layer which holds input's identifier. The layers between input and output layers are called hidden layer(s) which only propagates previous layer's outputs to the next layer and back propagates the following layer's error to the previous layer. Actually, these are the main operations of training BPN which follows a few steps.

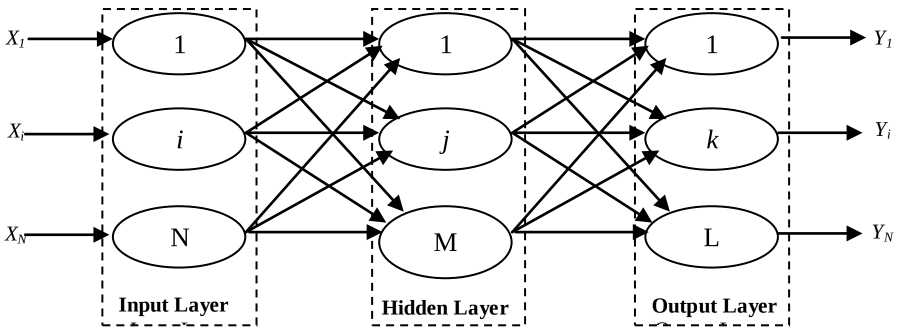

Fig. 2. Basic BPN Architecture.

Training such a network involves two phases [11]. In the first phase, the inputs are propagated forward to compute the outputs for each output node. Then, each of these outputs is subtracted from its desired output, causing an error. In the second phase, each of these output errors is passed backward and the weights are fixed. These two phases is continued until sum of square of output errors reaches to an acceptable value. During the training, several sets of the input and their corresponding output vectors are considered. The training phase is used to determine the weights between the input, hidden and output layers. The neurons used in the study utilize the sigmoid activation function defined by the following Eq. (6)

$$\begin{bmatrix} \text{Neuron} \\ \text{Output} \end{bmatrix} = \frac{1.0}{1.0 + e^{-\alpha v}} \tag{6}$$

where α is the abruptness of the sigmoid function and the range is 0-1 and the v is the total input to the neuron. Let the vector **X** represent an input to the input layer as shown in the Fig. 2. The net input at the hidden layers is computed by the matrix in Eq. (7).

$$V_H = [WH_{ji}]X \tag{7}$$

where WH_{ji} denotes the weight between i^{th} input layer node and j^{th} hidden layer node.

The output of the hidden layer nodes are given by Eq. (8)

$$Y_H = \Phi(V_H) \tag{8}$$

where Φ is the appropriate sigmoid activation function. In a similar manner, the total input at the output layer is given by Eq. (9).

$$V_O = [WO]\, V_H \tag{9}$$

WO is a random weight selection at the initial stage (starting of the iteration). V_o is compared with the actual output and if there is any difference the weight *WO* is adjusted according to the error during the successive iteration. This process (adjustment) continues until the error (difference between the calculated output V_o and the actual output) is zero or less than the specified tolerance. The output of the output layer node is given by

$$Y = \Phi(V_O) \tag{10}$$

The steps of the well-established training algorithm based upon Newton's steepest descent technique are given below:

Step 1. Read in the training set and randomly initialize the weights. Set iteration index n=1.
Step 2. Set training set index p=1.
Step 3. Propagate X^p through the network.
Step 4. Determine the error vector of the p^{th} training set $\mathbf{E}^p = \mathbf{O}^p - \mathbf{Y}^p$ where $\mathbf{O}^P$ is the vector of expected output.
Step 5. Correct the weights using Newton's steepest descent technique.
Step 6. If $p <$ number training sets P, set $p = p$+1 and go to step 3.
Step 7. If $\sum_{p=1}^{P} \left| E^p \right|^2 >$ tolerance ε, increment the iteration index n and go to step 2.

The above method works well and has been well documented. It requires input and output from a continuous domain. Furthermore, it also requires that the input and output set of vectors are non-contradictory for a successful training and operational function.

3.2. Implementation of BPN to compute the total cycle time, *WT*

The input vector for the BPN is the number of lots circulating in the system. Thus

$$\mathbf{X} = [k_1, k_2, k_3, \ldots k_N] \tag{11}$$

where k_N is the number of lots circulating for a given period of time.

Several sets of lots were created by the following scheme:

(a) Varying the number of lots of a system without re-entrant.

(b) Varying the number of lots for a system with a single re-entrant line.

The k^{th} such lots generated is referred to by the vector $\mathbf{X}^K$. Similarly the corresponding output vector for this k^{th} input is referred to by $\mathbf{O}^K$. The output vector refers to the total cycle time for a given number of lots.

Summarizing, several of these input and output vector pairs are generated by the analytical method explained in Section 2 and are stored for the BPN training. After the successful training of the BPN model, it should able to produce the total cycle time for any number of lots with minimum time and maximum accuracy.

4. Computational Experience and Results

A 5 stage re-entrant system is tested using the proposed method. Two situations are applied which are the single re-entrant and without re-entrant. In order to achieve a broad representation of the re-entrant system in the Back Propagation Network, approximately 100 input-output vector pairs are generated for the considered RS operation. The training is done in two fold. Firstly, a system without any re-entrant is attempted and secondly, the system with a single re-entrant is tested. The BPN was trained in MATLAB® environment [12] and results of this training for both cases are shown in Figs. 3 and 4 respectively. For these cases, the training seemed to require 162 and 196 iterations in order to achieve the goal which is relatively fast.

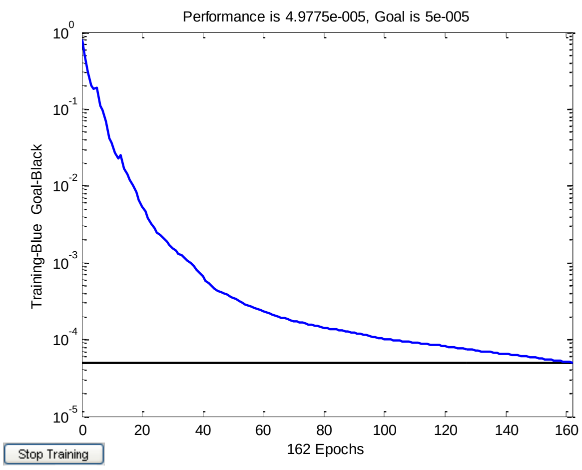

Fig. 3. BPN Training Results for System without Re-entrant.

Once the training is complete, the BPN model is ready for use. Several sets of inputs were tested and documented. However, some of the results are reported and presented. The results from the analytical MVA method and from the trained BPN for both cases of re-entrant are tabulated in Tables 1 and 2. Comparative results indicate that there is a close agreement between both methods. To support further on the finding, historical data (actual data) were gathered from the RS operation. Comparison result with the actual data shows slight variation due handling issues during the operation which tend to be stochastic in nature. However the differences are insignificant because it is not the natural characteristic of the operation. The proposed BPN method seems to work well and provides fast and reliable results and can be used as an alternate method to determine W_T.

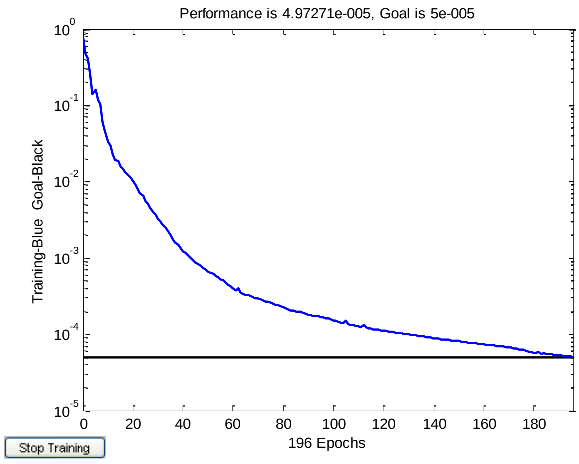

Fig. 4. BPN Training Results for System with a Single Re-entrant.

5. Conclusions

This paper presents a well defined BPN approach to determine the total cycle time, W_T for a 5 stage re-entrant line system which is utilized in a Reflow Screening operation. Several sets of lots are considered and their solutions are assessed using the analytical method. Then using these sets of input and output vector pairs, the Back Propagation Network is trained. Thereafter, the BPN is ready for use wherein, given number of lots, it gives out the total cycle time as a solution with minimum time and maximum accuracy. The contribution of this work is the development of a BPN method to determine the total cycle time for an RS operation. However, the considered system is treated in a purely deterministic way. This can be used as an alternate method apart for the conventional analytical approach. The benefit of this approach is that it provides a fast and reliable solution after a good set of training being done. For future work, systems having probabilistic re-entrant lines can be attempted and use the BPN architecture to develop the performance measure.

Table 1. Comparative Results (BPN, MVA and Actual) for a System without Re-entrant Line.

Number of Lots (k)	Total Cycle Time, W_T			
	Analytical (MVA) (Hours)	BPN (Hours)	Actual (Hours)	% Accuracy
81	20.250	20.2475	20.3	99.75
82	20.500	20.5	20.58	100
83	20.750	20.75	20.89	100
84	21.000	21	21.1	100
85	21.250	21.25	21.5	100
86	21.500	21.4975	21.55	99.75
87	21.750	21.7475	21.9	99.75
88	22.000	21.9975	22.4	99.75
89	22.250	22.25	22.58	100
90	22.500	22.5	22.89	100
91	22.750	22.7475	22.95	99.75
92	23.000	23	23.4	100
93	23.250	23.245	23.45	99.5
94	23.500	23.505	23.59	99.5
95	23.750	23.755	23.98	99.5
96	24.000	24.01	24.2	99
97	24.250	24.2725	24.6	97.7
98	24.500	24.5075	24.89	99.2
99	24.750	24.71	25.3	96
100	25.000	25	2.54	100

Table 2. Comparative Results (MVA, BPN and Actual) for a System with a Single Re-entrant Line.

Number of Lots (k)	Total Cycle Time, W_T			
	Analytical (MVA) (Hours)	BPN (Hours)	Actual (Hours)	% Accuracy
81	20.512	20.513	20.59	99.9
82	20.762	20.763	20.9	99.9
83	21.011	21.013	21.3	99.8
84	21.261	21.261	21.51	99.9
85	21.511	21.511	21.78	99.9
86	21.761	21.761	21.8	100
87	22.011	22.013	22.18	99.8
88	22.261	22.261	22.35	100
89	22.511	22.508	22.78	99.8
90	22.760	22.763	22.8	99.7
91	23.010	23.013	23.4	99.7
92	23.260	23.258	23.3	99.8
93	23.510	23.511	23.78	99.9
94	23.760	23.764	23.9	99.6
95	24.010	24.011	24.2	99.9
96	24.260	24.259	24.39	99.9
97	24.510	24.529	24.87	98.1
98	24.760	24.769	24.98	99.1
99	25.009	24.941	25.05	93.1
100	25.259	25.259	25.7	100

References

1. Kumar, S.; and Omar, M.K. (2005). Performance measure in a probabilistic reflow screening line using mean value analysis. *AIUB Journal of Science and Engineering*, 4(1), 53-58.
2. Kumar, S. (2008). Modeling of a probabilistic re-entrant line bounded by limited operation utilization time. *Journal of Industrial and System Engineering,* 2(1), 28-40.
3. Little, J.D.C. (1961). A proof for the queueing formula: $L=\lambda W$. *Operations Research*, 9(3), 383-387.
4. Reiser, M.; and Lavenberg, S.S. (1980). Mean-value analysis of closed multichain queueing networks. *Journal of the Association for Computing Machinery*, 27(2), 313-322.
5. Gross, D.; and Harris, C.M. (1998). *Networks, series and cyclic queue. Fundamentals of queueing theory*. New York: John Wiley & Sons, 189-192.
6. Park, Y.; Kim, S.; and Jun, C.H. (2002). Mean value analysis of re-entrant line with batch machines and multi-class job. *Computers & Operations Research*, 29(8), 1009-1024.
7. Kim, S.; Park, Y.; and Jun, C.H. (2006), Performance evaluation of re-entrant manufacturing system with production loss using mean value analysis. *Computers & Operations Research*, 33(5), 1308-1325.
8. Kumar S. (2007). *Performance analysis of a probabilistic re-entrant line in an environmental stress testing operation*. Doctoral Thesis, Multimedia University, Malaysia.
9. Kumar, S.; and Omar, M.K. (2006). Stochastic re-entrant line modeling for an environmental stress testing in a semiconductor assembly industry. *Applied Mathematics and Computation,* 173(1), 603-615.
10. McCollum P. (1997). An introduction to back-propagation neural networks. *The Newsletter of the Seattle Robotics Society*.
11. Shihab, K. (2006). A back propagation neural network for computer network security. *Journal of Computer Science,* 2(9), 710-715.
12. Introduction to the BPN, Neural Network Toolbox. *MathLab* 7.1*Version.*