2-18-2016

Modeling and Performance Analysis of Manufacturing Systems Using Max-Plus Algebra

Abdulrahman Saad Salah-Eldin Seleim
University of Windsor

Modeling and Performance Analysis of Manufacturing Systems Using Max-Plus Algebra

By

Abdulrahman Seleim

A Dissertation
Submitted to the Faculty of Graduate Studies
through the Department of Industrial and Manufacturing Systems Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada

2016

Modeling and Performance Analysis of Manufacturing Systems Using Max-Plus Algebra

by

Abdulrahman Seleim

APPROVED BY:

Dr. Pedro Cunha, External Examiner
Escola Superior de Tecnologia/Instituto Politécnico de Setubal, Portugal

---

Dr. Xiaobu Yuan
School of Computer Science

Dr. Zbignew Pasek
Industrial and Manufacturing Systems Engineering

Dr. Fazle Baki
Industrial and Manufacturing Systems Engineering

Dr. Hoda ElMaraghy, Advisor
Industrial and Manufacturing Systems Engineering

Feb 3rd, 2016

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research of the author and his supervisor Prof. Hoda ElMaraghy. This joint research has been published / submitted to various Journals that are listed below.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from Prof. Hoda ElMaraghy to include that material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This thesis includes 4 original papers that have been previously published / submitted for publication in peer reviewed journals and conferences as follows:

Thesis Chapter	Publication title/full citation	Publication Status
3	Seleim, A. and H. ElMaraghy (2015). "Generating max-plus equations for efficient analysis of manufacturing flow lines." Journal of Manufacturing Systems 37: 426-436.	Journal (Published)
4	Seleim, A. and H. ElMaraghy (2014). "Parametric analysis of Mixed-Model Assembly Lines using max-plus algebra." CIRP Journal of Manufacturing Science and Technology 7(4): 305-314.	Journal (Published)
3	Seleim, A., ElMaraghy, H., (2014). "Max-Plus Modeling of Manufacturing Flow Lines", Proceedings of the 47th CIRP Conference on Manufacturing Systems (CMS2014), Windsor, ON , Canada, Procedia CIRP, Vol. 17, pp. 302-307.	Conference Proceeding
5	Seleim, A., ElMaraghy, H., (2013). "Transient Analysis of a Re-entrant Manufacturing System", 5th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2013), Munich, Germany, pp. 261-266.	Conference Proceeding

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

In response to increased competition, manufacturing systems are becoming more complex in order to provide the flexibility and responsiveness required by the market. The increased complexity requires decision support tools that can provide insight into the effect of system changes on performance in an efficient and timely manner.

Max-Plus algebra is a mathematical tool that can model manufacturing systems in linear equations similar to state-space equations used to model physical systems. These equations can be used in providing insight into the performance of systems that would otherwise require numerous time consuming simulations.

This research tackles two challenges that currently hinder the applicability of the use of max-plus algebra in industry. The first problem is the difficulty of deriving the max-plus equations that model complex manufacturing systems. That challenge was overcome through developing a method for automatically generating the max-plus equations for manufacturing systems and presenting them in a form that allows analyzing and comparing any number of possible line configurations in an efficient manner; as well as giving insights into the effects of changing system parameters such as the effects of adding buffers to the system or changing buffers sizes on various system performance measures. The developed equations can also be used in the operation phase to analyze possible line improvements and line reconfigurations due to product changes. The second challenge is the absence of max-plus models for special types of manufacturing systems. For this, max-plus models were developed for the first time for modeling mixed model assembly lines (MMALs) and re-entrant manufacturing systems.

The developed methods and tools are applied to case studies of actual manufacturing systems to demonstrate the effectiveness of the developed tools in providing important insight and analysis of manufacturing systems performance. While not covering all types of manufacturing systems, the models presented in this thesis represent a wide variety of systems that are structurally different and thus prove that max-plus algebra is a practical tool that can be used by engineers and managers in modeling and decision support both in the design and operation phases of manufacturing systems.

DEDICATION

To my parents, wife, and children.

ACKNOWLEDGEMENTS

Before all, praise goes to God for giving me the strength and knowledge to complete this work. The completion of this dissertation would not have been possible without the support and help of many people, to whom I would like to express my gratitude. On top of the list comes Prof. Hoda ElMaraghy, my supervisor, who provided me with continuous and invaluable support, guidance, and mentorship. I would also like to express my appreciation to the PhD committee members: Dr. Pedro Cuhna, Dr. Xiaobu Yuan, Dr. Zbignew Pasek, andDr. Fazle Baki , for their valuable comments and suggestions which vastly improved the quality of the dissertation. Special thanks also go to my colleagues and mentors at the Intelligent Manufacturing Systems Centre (IMSC) at the University of Windsor, Prof. Waguih ElMaraghy, Dr. Ahmed Deif, Dr. Ayman Youssef, Dr. Tarek AlGeddawy, Dr. Sameh Samy, Dr. Mohammed Hanafy, Ahmed Adawy, Dr. Mohamed Kashkoush, Mohamed Abbas, Javad Navaei and Kourosh Khedri, with whom I had numerous comments, suggestions, and stimulating discussions.

I am also very grateful to my parents who always pushed me forward and supported me, and to my wife who enlightened my life and added three boys to our family during my PhD.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1. Motivation

In today's highly competitive market, it is not enough to produce products with excellent quality and low price. Under fierce competition, manufacturers are required to introduce a wide variety of products and produce them in the right quantity and at the right time. Under these circumstances, decision makers require good supporting tools that they can use to understand which parameters affect their production system as well as effect of each of these parameters on the overall performance.

Manufacturing systems can be classified under the category of Discrete Event Dynamic Systems (DEDSs) which also includes computer systems, traffic systems, and communication systems. What characterizes these systems is that their state changes not with time, but with certain events and the change from one state to another takes place instantaneously. For manufacturing systems, such events can be the arrival of a work piece, the breakdown of a machine, etc. These systems are structurally different from natural physical dynamic systems that are governed by differential equations. The behavior and natural physical systems can be accurately monitored, explained, predicted and controlled by the use of differential equations; on the other hand, for DEDSs such a robust and powerful mathematical tool does not exist yet (Cassandras and Lafortune 2007)(Cassandras and Lafortune 2007).

Available tools for modeling and performance evaluation of DEDSs include *Queuing Theory, Markov Chains, Petri Nets, Mathematical Programming, Discrete Event Simulation, and Max-plus Algebra*, figure 1.1. Both queuing theory and Markov chains are tools that deal with the *average* system performance over long time periods and thus are not very useful in short-term system analysis and control and gives little insight into the system's dynamics and behavior. Petri Nets is more of a logical tool that gives qualitative analysis of the system such as detecting deadlocks but cannot give quantitative analysis on the system behavior. Discrete event simulation is an excellent tool for the analysis of manufacturing systems' behavior and can give detailed picture of the system, however it is time consuming and can give information on the system only for the given simulated system parameters. In order to use discrete event simulation to get insight into the effect of a given system parameter on the overall behavior, numerous simulation runs would be required.

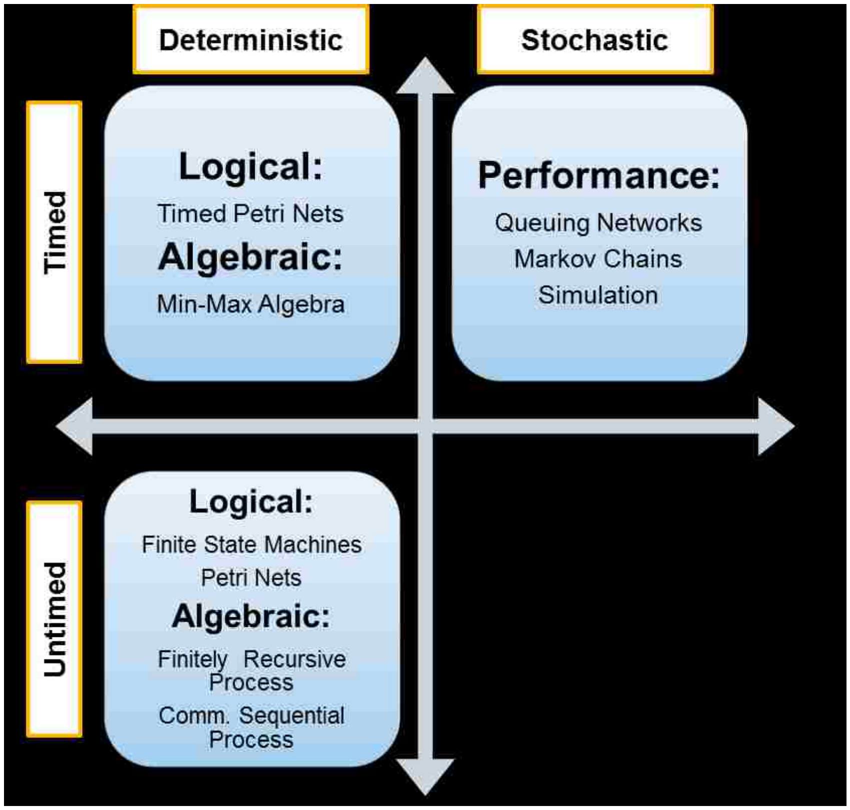


Figure 1.1 Classification of DEDS modeling tools

Max-plus algebra is an algebraic mathematical formulation that can be used to model manufacturing systems by linear state-space like equations. By modeling manufacturing systems using max-plus algebra, one can arrive at mathematical equations that can be used in the analysis and control of manufacturing systems. The use of max-plus algebra in modeling and analysis of manufacturing systems started in the nineteen eighties; however its use both commercially and academically has been limited. This is mainly because using the tool requires special mathematical background and because there are no user-friendly tools that facilitate the use of max-plus algebra in modeling and analysis of systems.

1.2. Scope

In this research different tools have been developed to make max-plus algebra more accessible to engineers and managers with little or no background in its mathematical foundation. The developed tools can enable engineers and managers to use max-plus equations in analyzing manufacturing systems and testing different what-if scenarios efficiently in both design and operation stages.

The first of these tools is a method for the automatic generation of max-plus equations for manufacturing flow lines. The method can be used to model lines with finite buffers and parallel identical stations and produces equations that can be used in parametric analysis.

The second tool is a novel approach in modeling mix-model assembly lines with max-plus algebra. The developed equations can then be used to compare given sequences of demand mix over a range of processing times of assembly tasks as well as analyze different line performance measures while considering one of the line parameters as a variable. Hence, the effect of changes in any of the system parameters on the optimality of a given sequence of demand mix and on the line performance can be assessed.

The third tool is method for modeling re-entrant manufacturing systems which are used widely in semiconductor manufacturing and paint shops. Using the developed equations, complex behavior especially in the transient phase can be detected and avoided.

It should be noted that the manufacturing systems modeled by max-plus algebra in this thesis do not cover all types of manufacturing systems. However, they represent structurally different types of systems and thus prove in principle that this tool is capable of modeling and providing useful analysis to a wide range of manufacturing systems.

1.3. Thesis Statement

The use of max-plus algebra in modeling and analysis of manufacturing systems can provide insights and information about the systems performance that cannot be otherwise efficiently obtained with available modeling tools

1.4. Max-Plus Algebra

Max-plus algebra is a mathematical tool that can model DEDSs using linear algebraic equations analogous to conventional state-space linear equations (Ho 1989). Using these equations, real time control and parametric system analysis become possible. Discrete event systems that can be modeled using max-plus equations include production systems (Di Febbraro, Minciardi et al. 1994), traffic light systems (Maia, Hardouin et al. 2013), public transportation systems (Nait-Sidi-Moh, Manier et al. 2005), and computer networking (Baccelli and Hong 2000).

Research related to max-plus algebra can be classified into two different categories. The first is research in developing the tool itself and increasing its appeal to potential users. Work in this category includes direct generation of max-plus equations for flow shop systems (Doustmohammadi and Kamen 1995; Seleim and ElMaraghy 2015), extending max-plus algebra to stochastic systems (Jean-Marie and Olsder 1996), introducing buffer and capacity constraints to the max-plus representation of manufacturing systems (Goto, Shoji et al. 2007), and introducing a block diagram based representation of manufacturing systems using max-plus algebra (Imaev and Judd 2008; Imaev and Judd 2009).

The second category of research related to max-plus algebra is concerned with applications of the tool. These include manufacturing systems modeling (Ren, Xu et al. 2007; Imaev and Judd 2008; Imaev and Judd 2009; Seleim and ElMaraghy 2014), performance evaluation (Cohen, Dubois et al. 1985; Amari, Demongodin et al. 2005; Reddy, Janardhana et al. 2009; Morrison 2010; Park and Morrison 2010; Boukra, Lahaye et al. 2013; Seleim and ElMaraghy 2014; Singh and Judd 2014), performance optimization (Di Febbraro, Minciardi et al. 1994), scheduling (Lee 2000; Goto, Hasegawa et al. 2007; Tanaka, Masuda et al. 2009; Houssin 2011) model predictive control (De Schutter and Van Den Boom 2001; van den Boom and De Schutter 2006; Goto 2013).

1.5. Dissertation Overview

Chapter 2 provides an introduction to the basics of max-plus algebra along with simple examples to make the reader familiar with how max-plus algebra works. Chapter 3 presents a method for automatic generation of max-plus equations for manufacturing flow lines. The method is first presented with simple examples then a case study is presented where a flow line for manufacturing a control valve is modeled and the generated equations are used to compare the line idle time of different configurations, the effect of buffer sizes on idle time and the effect of the processing time of a station on the total line idle time. Chapter 4 covers max-plus modeling of mixed-model assembly lines (MMALs) with either open or closed stations. Case studies are presented to show how the developed equations can be used in determining the robustness of a given solution to the assembly line sequencing problem. Chapter 5 tackles the issue of modeling re-entrant flow lines. The difficulty of modeling re-entrant flow lines using max-plus is first demonstrated, then a novel method is presented which allows for modeling these category of manufacturing systems. The developed equations are then used to analyze the effect of different system parameters on the transient and steady state behavior of the line. Finally, chapter 6 presents an overview and discussion, the research contributions, significance, and future work.

Table 1-1 Summary of Literature Review

Tool Development		Applications	
Equations generation	(Doustmohammadi and Kamen 1995; Seleim and ElMaraghy 2015)	**Modeling manufacturing systems**	(Ren, Xu et al. 2007; Imaev and Judd 2008; Imaev and Judd 2009; Seleim and ElMaraghy 2014)
Modeling Stochastic Systems	(Jean-Marie and Olsder 1996)	**Performance evaluation**	(Cohen, Dubois et al. 1985; Amari, Demongodin et al. 2005; Reddy, Janardhana et al. 2009; Morrison 2010; Park and Morrison 2010; Boukra, Lahaye et al. 2013; Singh and Judd 2014)
Buffer and capacity constraints	(Goto, Shoji et al. 2007)	**Performance optimization**	(Di Febbraro, Minciardi et al. 1994)
Block diagram representation	(Imaev and Judd 2008; Imaev and Judd 2009)	**Scheduling**	(Lee 2000; Goto, Hasegawa et al. 2007; Tanaka, Masuda et al. 2009; Houssin 2011)
		Control	(De Schutter and Van Den Boom 2001; van den Boom and De Schutter 2006; Houssin, Lahaye et al. 2007; Goto 2013)

CHAPTER 2: BASICS OF MAX-PLUS ALGEBRA

Max-plus algebra is one of many algebraic structures called *semirings or dioids* that are studied by mathematicians. The most famous of these semirings are the max-plus algebra, min-plus algebra, and the min-max algebra. These algebraic tools have been studied by mathematicians for many years and used in areas of optimization and algebraic geometry, but the first use of these tools in modeling discrete event systems was in 1985 by Cohen et al.(Cohen, Dubois et al. 1985). In their paper, Cohen et al. indicated that deterministic, discrete event systems can be represented in a linear state-space representation when modeled by these algebraic structures. Following that paper, max-plus algebra started to be used in modeling, control, and performance analysis of discrete event systems (Cohen, Gaubert et al. 1999).

In this chapter an introduction to the basic concepts and tools of the Max-Plus algebra is first presented then used to model a simple manufacturing system consisting of three machines. A more detailed presentation of max-plus algebra with in depth mathematical analysis and proofs can be found in (Baccelli, Cohen et al. 1992) and (Heidergott, Olsder et al. 2006).

2.1. Max-plus Algebra Basics

Max-Plus algebra is defined over $\mathcal{R}_{max} = \{\mathcal{R} \cup -\infty\}$ where $\mathcal{R}$ is the set of real numbers. The two main algebraic operations are maximization, denoted by the symbol $\oplus$, and addition, denoted by the symbol $\otimes$ where:

$$a \oplus b = max(a, b) \quad \forall\, a, b\ \in \mathcal{R}_{max}$$

$$a \otimes b = a + b \quad \forall\, a, b\ \in \mathcal{R}_{max}$$

Define $\varepsilon = -\infty$ and $e = 0$. In max-plus algebra, ε is the null element of the operation $\oplus$ where

$$a \oplus \varepsilon = max\ (a, -\infty) = a \quad \forall\, a\ \in \mathcal{R}_{max}$$

and e is the null element for the operation $\otimes$ where

$$a \otimes e = a + 0 = a \quad \forall\, a\ \in \mathcal{R}_{max}$$

Throughout the rest of this dissertation, $\oplus$ will be referred to as addition (or plus) and $\otimes$ will be referred to as multiplication. Similar to traditional algebra, both $\oplus$ and $\otimes$ are associative and commutative

$$a \otimes b = b \otimes a \;\; \forall\, a, b \; \in \mathcal{R}_{max}$$

$$(a \otimes b) \otimes c = a \otimes (b \otimes c) \;\; \forall\, a, b, c \; \in \mathcal{R}_{max}$$

$$a \oplus b = b \oplus a \;\; \forall\, a, b \; \in \mathcal{R}_{max}$$

$$(a \oplus b) \oplus c = a \oplus (b \oplus c) \;\; \forall\, a, b, c \; \in \mathcal{R}_{max}$$

and multiplication is left and right distributive over addition

$$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c) \;\; \forall\, a, b, c \; \in \mathcal{R}_{max}$$

$$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c) \;\; \forall\, a, b, c \; \in \mathcal{R}_{max}$$

Similar to conventional algebra, max-plus algebra can be extended over matrices. Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two matrices with equal dimension, then

$$\boldsymbol{A} \oplus \boldsymbol{B} = \boldsymbol{C}$$

where $\boldsymbol{C}_{ij} = \boldsymbol{A}_{ij} \oplus \boldsymbol{B}_{ij}$. If the number of columns of $\boldsymbol{A}$ is equal to the number of rows of $\boldsymbol{B}$ equal to n, then:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \boldsymbol{C}$$

where

$$\boldsymbol{C}_{ij} = \bigoplus_{k=1}^{n} \boldsymbol{A}_{ik} \otimes \boldsymbol{B}_{kj}$$

where $\bigoplus_{k=1}^{n} \boldsymbol{q}$ is maximization of all the elements of $\boldsymbol{q}$ for $k = 1$ to n.

If a is a scalar and $\boldsymbol{A}$ is a matrix, then $a \otimes A$ is equivalent to adding the value of a to each element in the matrix A.

To illustrate addition and multiplication over matrices, let

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 \\ e & \varepsilon \end{bmatrix},\ \boldsymbol{B} = \begin{bmatrix} e & 6 \\ 9 & 1 \end{bmatrix},\ \boldsymbol{C} = \begin{bmatrix} 7 & 9 & \varepsilon \\ 2 & e & 4 \end{bmatrix}\ and\ \boldsymbol{D} = \begin{bmatrix} 1 & 5 \\ e & \varepsilon \\ 7 & 3 \end{bmatrix},$$

then:

$$4 \otimes \boldsymbol{A} = \begin{bmatrix} 3+4 & 2+4 \\ e+4 & \varepsilon+4 \end{bmatrix} = \begin{bmatrix} 7 & 6 \\ 4 & \varepsilon \end{bmatrix}$$

$$\boldsymbol{A} \oplus \boldsymbol{B} = \begin{bmatrix} 3 \oplus e & 2 \oplus 6 \\ e \oplus 9 & \varepsilon \oplus 1 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 9 & 1 \end{bmatrix}$$

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} 3 \otimes e \oplus 2 \otimes 9 & 3 \otimes 6 \oplus 2 \otimes 1 \\ e \otimes e \oplus \varepsilon \otimes 9 & e \otimes 6 \oplus \varepsilon \otimes 1 \end{bmatrix} = \begin{bmatrix} 11 & 9 \\ e & 6 \end{bmatrix}$$

$$\boldsymbol{C} \otimes \boldsymbol{D} = \begin{bmatrix} (7 \otimes 1) \oplus (9 \otimes e) \oplus (\varepsilon \otimes 7) & (7 \otimes 5) \oplus (9 \otimes \varepsilon) \oplus (\varepsilon \otimes 3) \\ (2 \otimes 1) \oplus (e \otimes e) \oplus (4 \otimes 7) & (2 \otimes 5) \oplus (e \otimes \varepsilon) \oplus (4 \otimes 3) \end{bmatrix}$$
$$= \begin{bmatrix} 9 & 12 \\ 11 & 7 \end{bmatrix}$$

$$\boldsymbol{A} \otimes \boldsymbol{A} = \boldsymbol{A}^{\boldsymbol{2}} = \begin{bmatrix} 3 \otimes 3 \oplus 2 \otimes e & 3 \otimes 2 \oplus 2 \otimes \varepsilon \\ e \otimes 3 \oplus \varepsilon \otimes e & e \otimes 2 \oplus \varepsilon \otimes \varepsilon \end{bmatrix} = \begin{bmatrix} 6 & 5 \\ 3 & 2 \end{bmatrix}$$

Through the rest of the dissertation, the $\otimes$ operator will be omitted whenever its use is obvious, thus $a \otimes b \oplus c \otimes d$ will be written as $ab \oplus cd$.

Theorem 2.1:

An equation is the general form:

$$\boldsymbol{X} = \boldsymbol{A}\,\boldsymbol{X} \oplus \boldsymbol{B}\,\boldsymbol{U} \qquad (2.1)$$

where $\boldsymbol{X}$ is an $n \times 1$ vector of variables, $\boldsymbol{U}$ is an $m \times 1$ vector of inputs, $\boldsymbol{A}$ is an $n \times n$ square matrix and $\boldsymbol{B}$ is an $n \times m$ matrix, has a solution :

$$\boldsymbol{X} = \boldsymbol{A}^{*}\,\boldsymbol{B}\,\boldsymbol{U} \qquad (2.2)$$

where $\boldsymbol{A}^{*}$ is defined as:

$$\boldsymbol{A}^{*} = e \oplus \boldsymbol{A} \oplus \boldsymbol{A}^{2} \oplus \ldots \oplus \boldsymbol{A}^{\infty} \qquad (2.3)$$

The proof of *theorem (2.1)* is presented in Appendix A.

If matrix $\mathbf{A}$ is regarded as a directed graph and the each entry $\mathbf{A}_{i,j}$ denote the weight of path from node *i* to *j*, then $\mathbf{A}^{n}{}_{i,j}$ denotes the weights of paths with length *n* in the same graph. Therefore, for $\mathbf{A}^{*}$to have a defined value, the weights of paths larger than a given *z* should equal to zero and thus we get $\mathbf{A}^{n} = -\infty$ for $n > z$ and equation *(2.3)* becomes:

$$\boldsymbol{A}^{*} = e \oplus \boldsymbol{A} \oplus \boldsymbol{A}^{2} \oplus \ldots \oplus \boldsymbol{A}^{z} \tag{2.4}$$

For an $n \times n$ matrix $\boldsymbol{A}$, an $n \times 1$ vector $\boldsymbol{v}$, and a scalar μ, if

$$\boldsymbol{A} \otimes \boldsymbol{v} = \mu \otimes \boldsymbol{v} \tag{2.5}$$

then μ is called the eigenvalue of $\boldsymbol{A}$ and $\boldsymbol{v}$ is its associated eigenvector. Assume, $\boldsymbol{X}$ is a vector of variables defining a system such that:

$$\boldsymbol{X_{k+1}} = \boldsymbol{A}\,\boldsymbol{X_{k}}$$

then at steady state, the eigenvalue of $\boldsymbol{A}$ is the average growth rate of X. The eigenvalues can be calculated using different numerical algorithms presented in (Heidergott, Olsder et al. 2006).

2.2. Example of Modeling a Manufacturing System

Consider a simple manufacturing system consisting of three stations *A*, *B*, and *C*. Stations *A* and *B* are independent and station *C* is an assembly operation that requires a workpiece from station *A* and another from station *B* as shown in Figure 2.1. Let the processing time for stations *A*, *B* and *C* be t1, t2 and t3 respectively, the starting time of processing the k^{th} workpiece on stations *A*, *B* and C be$x1_k$,$x2_k$, and $x3_k$ respectively, the time at which required inputs are made available to stations *A* and *B* for the k^{th} time be U1 and U2 respectively and the time the k^{th} workpiece has finished processing on station *C*, i.e. arrival time of the k^{th} finished product, is Y_k.

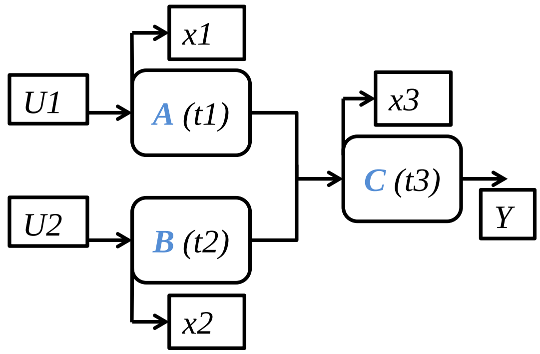


Figure 2.1 A simple 3 machine manufacturing system.

Considering station *A*, the time at which the station starts processing job *k* is the later of the two the two events: 1) required inputs for job *k* are available, which is equal to $U1_k$, and 2) station *A* has finished processing the job *k-1*, which is equal to the time at which station *A* started processing the job *k-1* plus the processing time on station *A*. In conventional algebra this can be written as:

$$x1_k = max\ (U1_k,\ x1_{k-1} + t1) \qquad (2.6)$$

Similarly for station *B*, the time at which the station starts processing job *k* can be written as:

$$x2_k = max\ (U2_k,\ x2_{k-1} + t2) \qquad (2.7)$$

For station *C*, processing the k^{th} jobs can start at the latest of three events:

1) station *A* has finished processing the k^{th} job, which is equal to $x1_k + t1$,

2) station *B* has finished processing the k^{th} job, which is equal to $x2_k + t2$,

3) station *C* has finished processing the job *k-1*.

In conventional algebra this can be written as:

$$x3_k = max\ (x1_k + t1,\ x2_k + t2, x3_{k-1} + t3) \qquad (2.8)$$

Equations *(2.6-2.8)* can be written in max-plus algebra as:

$$x1_k = U1_k \oplus t1\ x1_{k-1} \quad (2.9)$$

$$x2_k = U2_k \oplus t2\ x2_{k-1} \quad (2.10)$$

$$x3_k = t1 x1_k \oplus t2\ x2_k \oplus t3\ x3_{k-1} \quad (2.11)$$

The arrival time of the k^{th} finished product is equal to the time it started processing on station *C* plus the processing time on station *C*, this can be written as:

$$\boldsymbol{Y}_k = t3\ x3_k \quad (2.12)$$

Equations *(2.9-2.12)* fully describe the simple manufacturing system in figure 2.1 and can be put in state-space vector form as:

$$\boldsymbol{X}_k = \boldsymbol{A}\,\boldsymbol{X}_k \oplus \boldsymbol{B}\,\boldsymbol{X}_{k-1} \oplus \boldsymbol{D}\,\boldsymbol{U}_k \quad (2.13)$$

$$\boldsymbol{Y}_k = \boldsymbol{C}\,\boldsymbol{X}_k \quad (2.14)$$

where:

$$\boldsymbol{X}_k = \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix}_k, \boldsymbol{U}_k = \begin{bmatrix} U1 \\ U2 \end{bmatrix}_k, \boldsymbol{A} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ t1 & t2 & \varepsilon \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} t1 & \varepsilon & \varepsilon \\ \varepsilon & t2 & \varepsilon \\ \varepsilon & \varepsilon & t3 \end{bmatrix}, \boldsymbol{D} = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e \\ \varepsilon & \varepsilon \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} \varepsilon & \varepsilon & t3 \end{bmatrix}.$$

Notice that equation *(2.13)* is implicit in $\mathbf{X}_k$. According to *theorem* (2.1), the implicit equation *(2.13)* can be transformed into:

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A}}\,\boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{B}}\,\boldsymbol{U}_k \quad (2.15)$$

where $\widehat{\boldsymbol{A}} = \boldsymbol{A}^*\boldsymbol{B}$, $\widehat{\boldsymbol{B}} = \boldsymbol{A}^*\boldsymbol{D}$ and using equation *(2.4)* $\boldsymbol{A}^*, \widehat{\mathbf{A}}$ and $\widehat{\boldsymbol{B}}$ can be calculated as:

$$\boldsymbol{A}^* = e \oplus \boldsymbol{A} \oplus \boldsymbol{A}^2 = \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix} \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ t1 & t2 & \varepsilon \end{bmatrix} \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} = \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ t1 & t2 & e \end{bmatrix},$$

$$\widehat{\boldsymbol{A}} = \boldsymbol{A}^*\boldsymbol{B} = \begin{bmatrix} t1 & \varepsilon & \varepsilon \\ \varepsilon & t2 & \varepsilon \\ t1 & t2 & t3 \end{bmatrix}, \text{ and } \widehat{\boldsymbol{B}} = \boldsymbol{A}^*\boldsymbol{D} = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e \\ t1 & t2 \end{bmatrix}.$$

Using equations *(2.14)* and *(2.15)* and given the arrival time of inputs to stations *A* and *B,* the time at which each station starts processing each job as well as the completion time of each job can be determined. These equations can then be used in dynamic analysis of the system as well as in dynamic control as mentioned in section 1.4.

It should be noted, however, that the example presented above assumes infinite buffer capacity between stations *A* and *B* and station *C*. Accounting for finite buffers between stations will be considered in chapter 3 when considering the method to automatically generate the equations for manufacturing flow lines.

In the case when different products are processed on the same manufacturing system and different products have different processing times on each machine, equations *(2.14)* and *(2.15)* can still be used while changing the parameters *t1*, *t2*, and *t3* into $t1_k$, $t2_k$, and $t3_k$ and accordingly the matrices ***A*, *B*,** and ***C*** will be changed to $\boldsymbol{A_k}$, $\boldsymbol{B_k}$, and $\boldsymbol{C_k}$.

2.3. Coding Max-plus Algebra in Wolfram Mathematica

The symbolic computational software *Mathematica 6.0* (Grzymkowski, Kapusta et al. 2008) was used for solving max-plus calculations. A toolbox-like code was developed that included the basic operations for max-plus and some advanced operations (like calculating the Eigenvalues). The complete code with proper comments is included in Appendix B.

CHAPTER 3: MAX-PLUS MODELING OF MANFUFACTURING FLOW LINES

3.1. Introduction

Modeling simple manufacturing systems using max-plus equations is easy and intuitive; however, as the systems grow in size and/or have complicated structure, deriving the model equations becomes tedious, less intuitive and time consuming. In addition, deriving max-plus equations for manufacturing systems with finite buffers or parallel identical stations is not straight-forward or easy even for simple systems. The difficulty of deriving these equations limits the benefits of using max-plus algebra in modeling and controlling manufacturing systems especially when frequent changes in products or system configurations take place and the need for quickly assessing their effects and making decisions intensifies.

In this chapter, a method for automatic generation of the max-plus system equations for flow lines is presented. The method can generate the equations for lines with complicated structures regardless of their size and can model finite buffers and parallel identical stations. Flow lines studied in this chapter are assumed to have deterministic processing times and reliable stations. The first assumption is realistic for automated systems as well as semi-automated systems with palletized material handling where the process time variation is much less than the processing time and thus can be neglected. The second assumption is also realistic when studying the short-term system operation with the objective of understanding and optimizing the system behavior rather than studying the long-term operation with the objective of planning system capacity where machines breakdown would have an effect.

A review of related research is presented in section 3.2. Section 3.3 presents the method for generating the max-plus equations followed by a case study with an example of analysis in section 3.4, and finally section 3.5 presents the discussion and conclusions.

3.2. Related Research

Several papers have been published focusing on facilitating the modeling of manufacturing systems using max-plus algebra. Doustmohammadi and Kamen (1995) presented a procedure for direct generation of event-time max-plus equations for generalized flow shop manufacturing systems. The procedure is limited to flow shops with infinite buffers and cannot model identical parallel machines. The procedure generates the equations directly only for serial flow lines with one station in each stage, otherwise the equations are generated for each machine separately, interconnection matrices which describe the flow of jobs through the line are derived and then the final equations are generated using matrix manipulations and recursions. Goto et al. (Goto, Shoji et al. 2007) proposed a manufacturing systems representation that can account for finite buffers by adding relations between future starting times of jobs on a station and past starting times for the same and subsequent stations. Imaev and Judd (2009) used block diagrams which can be interconnected to form a manufacturing system model. This approach also assumes infinite buffer sizes and cannot model parallel redundant machines. Park and Morrison (2010) presented a method for modeling flow lines with parallel redundant stations again by adding relations between future and past starting times on a station and the subsequent ones. However, their equations provide the processing starting time for jobs not stations, which is unusual in modeling manufacturing systems and causes the model variables and number of equations to grow with the number of jobs.

In summary, the literature is lacking a method for generating max-plus equations for complex flow lines which contain finite buffers and parallel identical stations.

3.3. Flow Lines Modeling

The presented method for modeling flow lines capitalizes on the observation that certain features of the line affect the final equations each in a specific way. For illustration, each specific feature will be presented separately to show its effect on the final equations and then the steps of arriving at the final equations for a general line will be presented followed by an example.

Modeling will start with a flow line with *n* serial stations, followed by *n* different lines merging (assembling) in one line, and then the effect of introducing parallel identical stations will be shown. Initially, infinite buffers are assumed before each station and then in section *3.3.4* the effect of introducing finite buffers will be presented. Finally in section 3.*3.5* the whole model will be assembled and demonstrated by an example of a manufacturing flow line that contains serial and merging stations, parallel identical stations and finite buffers.

3.3.1. Modeling 'n' serial stations

The most common structure of a flow line is a serial structure with *n* processing stations, one input of incoming parts $\boldsymbol{U}$, and one output of finished products $\boldsymbol{Y}$ as shown in figure 3.1. Let U_k,

Y_k, and $X_{i,k}$ be the time at which the incoming parts are made available to the line, the time at which the finished product leaves the line and the starting time of processing on the i^{th} station for the k^{th} job respectively.

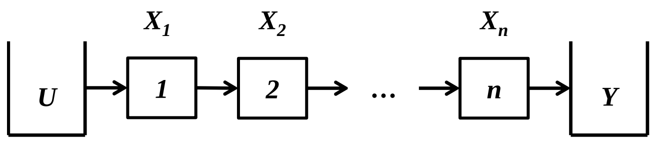


Figure 3.1 Flow line with n serial stations

For station 1 to start processing the k^{th} job, the following conditions must be fulfilled: 1) Arrival of incoming parts for the k^{th} job, and 2) Completion of processing the $k\text{-}1^{th}$ job. If t_1 is the processing time for station 1, then these conditions are translated into the following equation:

$$X_{1,k} = max(\, t_1 + X_{1,k-1}, U_k) \tag{3.1}$$

which is presented in the max-plus algebra as:

$$X_{1,k} = t_1 X_{1,k-1} \oplus U_k \tag{3.2}$$

Similarly, for any station i the conditions are: 1) End of processing the k^{th} job on the $i\text{-}1^{th}$ station, and 2) End of processing the $k\text{-}1^{th}$ job on i^{th} station. These are expressed in max-plus algebra as:

$$X_{i,k} = t_i X_{i,k-1} \oplus t_{i-1} X_{i-1,k} \tag{3.3}$$

Combining equations *(3.2)* and *(3.3)* in matrix form yields:

$$\boldsymbol{X}_k = \boldsymbol{A}\,\boldsymbol{X}_k \oplus \boldsymbol{B}\,\boldsymbol{X}_{k-1} \oplus \boldsymbol{D}\,\boldsymbol{U}_k \tag{3.4}$$

where,

$$\boldsymbol{X}_k = \begin{bmatrix} X_{1,k} \\ X_{2,k} \\ \vdots \\ X_{n,k} \end{bmatrix}, \ \boldsymbol{A} = \begin{bmatrix} \varepsilon & \varepsilon & \cdots & \varepsilon \\ t_1 & \varepsilon & \cdots & \varepsilon \\ & \ddots & & \vdots \\ \varepsilon & \varepsilon & t_{n-1} & \varepsilon \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} t_1 & \varepsilon & & \varepsilon \\ \varepsilon & t_2 & & \varepsilon \\ \vdots & & \ddots & \vdots \\ \varepsilon & \varepsilon & \cdots & t_n \end{bmatrix}, \text{and } \boldsymbol{D} = \begin{bmatrix} e \\ \varepsilon \\ \vdots \\ \varepsilon \end{bmatrix}.$$

Following theorem (2.1), equation *(3.4)* can be written as:

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A}}\,\boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{B}}\,\boldsymbol{U}_k \tag{3.5}$$

where:

$$\widehat{\boldsymbol{A}} = \boldsymbol{A}^* \otimes \boldsymbol{B} = \begin{bmatrix} t_1 & \varepsilon & \dots & \varepsilon \\ t_1^2 & t_2 & & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ t_1^2 t_2 \ldots t_{n-1} & t_2^2 t_3 \ldots t_{n-1} & \dots \; t_{n-1}^2 & t_n \end{bmatrix},$$

$$and \;\; \widehat{\boldsymbol{B}} = \boldsymbol{A}^* \otimes \boldsymbol{D} = \begin{bmatrix} e \\ t_1 \\ \vdots \\ t_1 t_2 \ldots t_{n-1} \end{bmatrix}.$$

From equation *(3.5)* it can be deduced that for any station *i*, the starting time for the k^{th} job is equal to:

$$X_{i,k} = t_i X_{i,k-1} \oplus t_{i-1}^2 X_{i-1,k-1} \oplus t_{i-2}^2 t_{i-1} X_{i-2,k-1} \oplus \dots \oplus t_1^2 t_2 \dots t_{i-1} X_{1,k-1} \oplus t_1 t_2 \dots t_{i-1} U_k \quad (3.6)$$

Since equations *(3.5)* and *(3.6)* were generated for a general serial flow line, they can be used to directly generate the max-plus equations for serial lines with any number of stages given the number of stations in the line.

3.3.2. Modeling ‘n’ merging lines

Merging lines are common in assembly flow lines. A merging station requires input from more than one station or line and delivers one output to the next station. Figure 3.2 shows *n* stations, each with its own input of incoming parts, merging into one station.

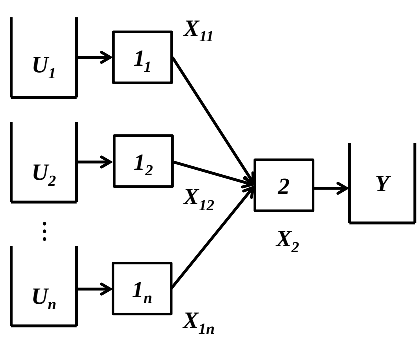


Figure 3.2 Flow line with *n* merging lines

If t_i is the processing time for station *i*, and $U_{i,k}$ is the time at which incoming parts are made available for the 1_i^{th} station, then equation *(3.2)* holds for any station 1_i and the conditions for station *2* to start processing are: 1) End of processing the k^{th} job on stations 1_i ($i = 1 \rightarrow n$), and 2) End of processing the $k\text{-}1^{th}$ job on station 2. Accordingly, the max-plus equations for the system in figure 3.2 can be presented as:

$$\boldsymbol{X}_k = \boldsymbol{A}\,\boldsymbol{X}_k \oplus \boldsymbol{B}\,\boldsymbol{X}_{k-1} \oplus \boldsymbol{D}\,\boldsymbol{U}_k \qquad (3.7)$$

where,

$$\boldsymbol{X}_k = \begin{bmatrix} X_{11,k} \\ X_{12,k} \\ \vdots \\ X_{1n,k} \\ X_{2,k} \end{bmatrix}, \boldsymbol{U}_k = \begin{bmatrix} U_{1,k} \\ U_{2,k} \\ \vdots \\ \\ U_{n,k} \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} \varepsilon & \varepsilon & & \dots & \varepsilon \\ \vdots & \vdots & & \dots & \vdots \\ & & \ddots & & \\ \varepsilon & \varepsilon & & \varepsilon & \varepsilon \\ t_1 & t_2 & \dots & t_{1n} & \varepsilon \end{bmatrix}, \boldsymbol{B} =$$

$$\begin{bmatrix} t_{11} & \varepsilon & \dots & & \varepsilon \\ \varepsilon & t_{12} & \varepsilon & \dots & \varepsilon \\ \vdots & \varepsilon & \ddots & & \vdots \\ & \vdots & & t_{1n} & \varepsilon \\ \varepsilon & \varepsilon & \dots & \varepsilon & t_2 \end{bmatrix}, \text{and } \boldsymbol{D} = \begin{bmatrix} e & \varepsilon & & \dots & \varepsilon \\ \varepsilon & e & \varepsilon & \dots & \varepsilon \\ \vdots & \varepsilon & \ddots & & \vdots \\ \varepsilon & \vdots & & e & \varepsilon \\ \varepsilon & \varepsilon & \dots & \varepsilon & \varepsilon \end{bmatrix}.$$

Again following theorem (2.1), equation *(3.7)* becomes:

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A}}\,\boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{B}}\,\boldsymbol{U}_k \qquad (3.8)$$

where:

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} t_{11} & \varepsilon & & \dots & \varepsilon \\ \varepsilon & t_{12} & & & \varepsilon \\ \vdots & & \ddots & & \vdots \\ \varepsilon & \varepsilon & & t_{1n} & \varepsilon \\ t_{11}^2 & t_{12}^2 & \dots & t_{1n}^2 & t_2 \end{bmatrix}, \text{ and } \widehat{\boldsymbol{B}} = \begin{bmatrix} e & \varepsilon & & \dots & \varepsilon \\ \varepsilon & e & & & \vdots \\ \vdots & & \ddots & & \\ \varepsilon & \varepsilon & & e & \varepsilon \\ t_{11} & t_{12} & \dots & & t_{1n} \end{bmatrix}.$$

From equation *(3.8)* it can be deduced that for any station 1_i, the starting time for the k^{th} job is equal to:

$$X_{1i,k} = t_{1i} X_{i,k-1} \oplus U_i \qquad (3.9)$$

and for station *2*, the starting time for the k^{th} time is equal to:

$$X_{2,k} = t_{11}^2 X_{11,k-1} \oplus t_{12}^2 X_{12,k-1} \oplus \dots \oplus t_2 X_{2,k-1} \oplus t_{11} U_{1,k} \oplus t_{12} U_{2,k} \oplus \dots \oplus t_{1n} U_{n,k} \qquad (3.10)$$

Equations *(3.8)*, *(3.9)*, and *(3.10)* can similarly be used to directly generate the max-plus equations for any number of merging lines.

From equations *(3.9)* and *(3.10)* it can be observed that the equation for merging lines is just a concatenation of the equations of several serial lines. Therefore, using equations *(3.6)* and *(3.10)* the $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ matrices can be constructed for any structure of flow lines with infinite buffers and no parallel identical stations at any stage.

3.3.3. Modeling parallel identical stations

Adding parallel identical stations is a common method for increasing capacity and throughput in flow lines. Modeling parallel identical stations in max-plus algebra is not straight forward as it represents a logical OR in the system where jobs arriving at the stage with parallel identical stations can go to one of the stations OR another. In max-plus algebra, modeling logical OR requires modeling all possible cases which increases the size of the model exponentially with the number of jobs. One possible approximation to make, in order to model *n* parallel identical stations, is to transform them into *n* serial ones each with a processing time of *t/n*, where *t* is the processing time of the parallel identical stations. This approximation will result in equal average throughput but not accurate starting and finishing times for stations.

Figure 3.3 shows a three stage flow line with *n* parallel identical stations in the second stage. For the stations in the first and third stages to start working on the k^{th} job, the same conditions mentioned in section *3.3.1* are required. However, for a station in the second stage, the condition that the station should have finished processing the $k\text{-}1^{th}$ job is not required as there are parallel stations that can process the job. Alternatively, all the parallel identical stations in the second stage can be regarded as one station with processing time t_2 and capacity of *n* jobs. Thus the condition that the station should have finished processing the $k\text{-}1^{th}$ job would be replaced by a condition that processing the $k\text{-}n^{th}$ job has ended. Thus, the model equations for the system in figure 3.3 would be:

$$\boldsymbol{X}_k = \boldsymbol{A}\,\boldsymbol{X}_k \oplus \boldsymbol{B_1}\,\boldsymbol{X}_{k-1} \oplus \boldsymbol{B_2}\,\boldsymbol{X}_{k-n} \oplus \boldsymbol{D}\,\boldsymbol{U_k} \qquad (3.11)$$

where:

$$\boldsymbol{B_1} = \begin{bmatrix} t_1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_3 \end{bmatrix} \textit{and}\ \boldsymbol{B_2} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_2 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}.$$

Using theorem (2.1), equation *(3.11)* can then be written as*:*

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A}}\, \boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{AP}}_{\boldsymbol{2}}\, \boldsymbol{X}_{k-n} \oplus \widehat{\boldsymbol{B}}\, U_k \qquad (3.12)$$

where:

$$\widehat{\boldsymbol{A}} = \boldsymbol{A}^* \otimes \boldsymbol{B}_{\boldsymbol{1}} = \begin{bmatrix} t_1 & \varepsilon & \varepsilon \\ t_1^2 & \varepsilon & \varepsilon \\ t_1^2 t_2 & \varepsilon & t_3 \end{bmatrix}, \widehat{\boldsymbol{B}} = \boldsymbol{A}^* \otimes \boldsymbol{D} = \begin{bmatrix} e \\ t_1 \\ \vdots \\ t_1 t_2 .. t_{n-1} \end{bmatrix},$$

$$and\ \ \widehat{\boldsymbol{AP}}_{\boldsymbol{2}} = \boldsymbol{A}^* \otimes \boldsymbol{B}_{\boldsymbol{n}} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_2 & \varepsilon \\ \varepsilon & t_2^2 & \varepsilon \end{bmatrix}.$$

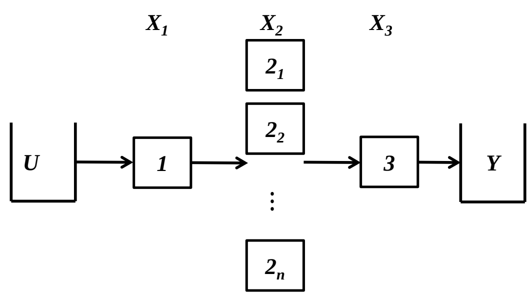


Figure 3.3 A three stage flow line with *n* parallel identical stations in the second stage.

By examining equations *(3.5)* and *(3.12)*, the following can be observed: first, matrix $\widehat{\boldsymbol{B}}$ is unchanged; second, matrix $\widehat{\boldsymbol{A}}$ is unchanged except for taking out the column corresponding to the stage where parallel stations are added and replacing it by a column of 'ε's; third, the column removed from matrix $\widehat{\boldsymbol{A}}$ is placed in a the same position in another matrix of 'ε's and multiplied by $\boldsymbol{X}_{k-n}$.

Thus in order to model parallel identical stations in one stage, it is assumed that only one station exists and the equations are generated as per section *3.3.1* or *3.3.2* then the column corresponding to the stage with parallel stations in matrix $\widehat{\boldsymbol{A}}$ is replaced by a column of 'ε's, then is inserted in another matrix $\widehat{\boldsymbol{AP}}$ and multiplied by the vector $\boldsymbol{X}_{k-n}$ where *n* is the number of parallel identical stations in that stage.

To demonstrate, assume a system as in figure 3.4 where all the parallel stations are identical and jobs arriving at each stage can be served by any station. The system is first assumed to be a serial line with four stages and one station in each stage. Accordingly, following equation *(3.5)*, the $\widehat{\boldsymbol{A}}$ matrix will be:

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} t_1 & \varepsilon & \varepsilon & \varepsilon \\ t_1^2 & t_2 & \varepsilon & \varepsilon \\ t_1^2 t_2 & t_2^2 & t_3 & \varepsilon \\ t_1^2 t_2 t_3 & t_1^2 t_3 & t_3^2 & t_4 \end{bmatrix}.$$

Using the generated matrix $\widehat{\boldsymbol{A}}$**,** the equations describing that system can be directly generated as:

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A}}\,\boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{AP}}_{\mathbf{1}}\,\boldsymbol{X}_{k-2} \oplus \widehat{\boldsymbol{AP}}_{\mathbf{3}}\,\boldsymbol{X}_{k-4} \oplus \widehat{\boldsymbol{AP}}_{\mathbf{4}}\,\boldsymbol{X}_{k-2} \oplus \widehat{\boldsymbol{B}}\,U_k \qquad (3.13)$$

where:

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_2 & \varepsilon & \varepsilon \\ \varepsilon & t_2^2 & \varepsilon & \varepsilon \\ \varepsilon & t_1^2 t_3 & \varepsilon & \varepsilon \end{bmatrix}, \widehat{\boldsymbol{AP}}_{\mathbf{1}} = \begin{bmatrix} t_1 & \varepsilon & \varepsilon & \varepsilon \\ t_1^2 & \varepsilon & \varepsilon & \varepsilon \\ t_1^2 t_2 & \varepsilon & \varepsilon & \varepsilon \\ t_1^2 t_2 t_3 & \varepsilon & \varepsilon & \varepsilon \end{bmatrix},$$

$$\widehat{\boldsymbol{AP}}_{\mathbf{3}} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_3 & \varepsilon \\ \varepsilon & \varepsilon & t_3^2 & \varepsilon \end{bmatrix}, \widehat{\boldsymbol{AP}}_{\mathbf{4}} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & t_4 \end{bmatrix}, \textit{and } \widehat{\boldsymbol{B}} = \begin{bmatrix} e \\ t_1 \\ t_1 t_2 \\ t_1 t_2 t_3 \end{bmatrix}.$$

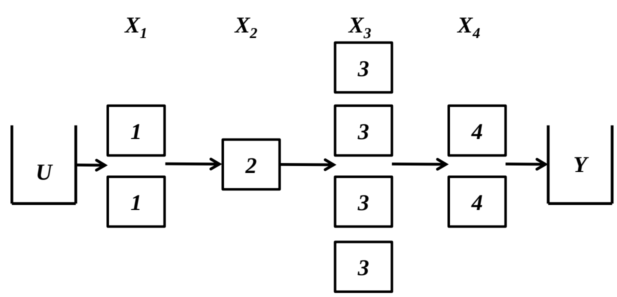


Figure 3.4 Flow line with parallel identical stations in several stages

It should be noted that equation *(3.13)* can be simplified by combining the matrices that are multiplied by the same delayed state vector, hence, equation *(3.13)* becomes:

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A}}\,\boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{AP}}_{\mathbf{1},\mathbf{4}}\,\boldsymbol{X}_{k-2} \oplus \widehat{\boldsymbol{AP}}_{\mathbf{3}}\,\boldsymbol{X}_{k-4} \oplus \widehat{\boldsymbol{B}}\,U_k \qquad (3.14)$$

where:

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_2 & \varepsilon & \varepsilon \\ \varepsilon & t_2^2 & \varepsilon & \varepsilon \\ \varepsilon & t_1^2 t_3 & \varepsilon & \varepsilon \end{bmatrix}, \widehat{\boldsymbol{AP}}_{\mathbf{1,4}} = \begin{bmatrix} t_1 & \varepsilon & \varepsilon & \varepsilon \\ t_1^2 & \varepsilon & \varepsilon & \varepsilon \\ t_1^2 t_2 & \varepsilon & \varepsilon & \varepsilon \\ t_1^2 t_2 t_3 & \varepsilon & \varepsilon & t_4 \end{bmatrix}, \widehat{\boldsymbol{AP}}_{\mathbf{3}} =$$

$$\begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_3 & \varepsilon \\ \varepsilon & \varepsilon & t_3^2 & \varepsilon \end{bmatrix}, \text{ and } \widehat{\boldsymbol{B}} = \begin{bmatrix} e \\ t_1 \\ t_1 t_2 \\ t_1 t_2 t_3 \end{bmatrix}.$$

However, it is better to keep the system equations in the form presented in equation *(3.13)* as it becomes clearer and easier to adjust the equations if the number of stations in any of these stages is changed.

3.3.4. Modeling finite buffers

To model finite buffers; assume a general station *i* followed by a buffer with a finite size *B*. For the k^{th} job to start on station *i* an additional condition is required to account for the buffer, which is for station *i+1* to have started processing the job number *k-B-1*. Assuming that station *i* mentioned above is part of a general flow line, then the line equations will be the same as equation *(3.5)* with the addition of one term as follows:

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A}}\, \boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{B}}\, U_k \oplus \widehat{\boldsymbol{A}}_{\boldsymbol{i}}\, \boldsymbol{X}_{k-B-1} \qquad (3.15)$$

where:

$$\widehat{\boldsymbol{A}}_{\boldsymbol{i}} = \boldsymbol{A}^* \otimes \boldsymbol{A}_{\boldsymbol{i}}, \text{ and } \boldsymbol{A}_{\boldsymbol{i}} = \begin{bmatrix} \varepsilon & \dots & \varepsilon & \varepsilon & \dots & \varepsilon \\ \vdots & & \vdots & \vdots & & \vdots \\ & & & \varepsilon & & \\ \varepsilon & \dots & \varepsilon & e & \varepsilon & \\ \vdots & & \vdots & \varepsilon & \vdots & \vdots \\ & & & \vdots & & \\ \varepsilon & \dots & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix},$$

where $\boldsymbol{A_i}$ is a null matrix with only one *e* located at the i^{th} row and the $i+1^{\text{th}}$ column.

To demonstrate, assume a flow line with four serial machines and three buffers as in figure 3.5. The size of buffers b_2, b_3 and b_4 is B_2, B_3 and B_4 respectively. The equations to model the line can be directly generated as:

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A}}\,\boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{B}}\,\boldsymbol{U}_k \oplus \widehat{\boldsymbol{AB}}_{\boldsymbol{2}}\,\boldsymbol{X}_{k-B_2-1} \oplus \widehat{\boldsymbol{AB}}_{\boldsymbol{3}}\,\boldsymbol{X}_{k-B_3-1} \oplus \widehat{\boldsymbol{AB}}_{\boldsymbol{4}}\,\boldsymbol{X}_{k-B_4-1} \tag{3.16}$$

where $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ are the same as in equation *(3.5)*,

$$\widehat{\boldsymbol{AB}}_{\boldsymbol{2}} = \begin{bmatrix} \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & t_1 & \varepsilon & \varepsilon \\ \varepsilon & t_1 t_2 & \varepsilon & \varepsilon \\ \varepsilon & t_1 t_2 t_3 & \varepsilon & \varepsilon \end{bmatrix}, \widehat{\boldsymbol{AB}}_{\boldsymbol{3}} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & t_2 & \varepsilon \\ \varepsilon & \varepsilon & t_2 t_3 & \varepsilon \end{bmatrix}, \text{and}$$

$$\widehat{\boldsymbol{AB}}_{\boldsymbol{4}} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & t_3 \end{bmatrix}.$$

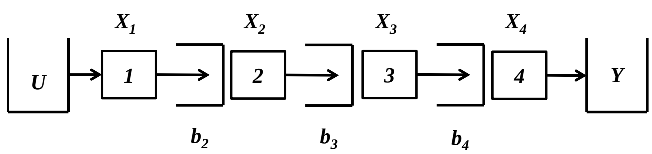


Figure 3.5 Flow line with 4 serial stations and 3 finite buffers.

By examining equation *(3.16)*, it is clear that the model of a finite buffer between two stations *i* and *j* uses the same equations for lines without the buffer with the addition of another matrix multiplied by vector $\boldsymbol{X}_{k-B-1}$ where *B* is the buffer size and this matrix is a Null matrix except for the j^{th} column which is equal to the column corresponding to station *i* in the $\widehat{\boldsymbol{A}}$ matrix divided by the processing time of station *i*.

3.3.5. Modeling general flow lines

An algorithm is presented for the automatic generation of the max-plus equations for a general flow line as follows:

Step 1: Simplify the flow line to be modelled by assuming infinite buffers and no parallel identical stations.

Step 2: Encode the simplified flow line into an adjacency matrix while assuming the line to be an undirected graph.

Step 3: Re-arrange the rows and columns of the matrix and identify merging stations according to the rank order clustering technique.

Step 4: Arrange X_i in the vector $\boldsymbol{X}$ according to the new order of stations in the adjacency matrix, where *i* is the total number of stations in the line excluding parallel identical ones.

Step 5: Generate the $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ matrices for the simplified flow line according to equations *(3.6)* and *(3.10)*.

Step 6: Take into account parallel identical stations by altering the $\widehat{\boldsymbol{A}}$ matrix and adding new matrices for each stage with parallel identical stations as described in section 3.3.3.

Step 7: Finalize the equations by accounting for finite buffers as described in section 3.3.4.

To demonstrate how the algorithm works consider the flow line shown in figure 3.6 (a) which includes parallel identical stations in stages *C* and *B* and finite buffers b_1, b_2 and $b_3$with sizes 2, 2 and 4 respectively.

In *step 1* the flow line in figure 3.6 (a) is transformed into that in figure 3.6 (b) with all buffers removed and parallel identical stations replaced by only one station.

Figure 3.7 shows the adjacency matrix of the simplified flow line following *step 2*. It should be noted that the function of the adjacency matrix is to encode the structure of the line into a digital form that can be used by software. Figure 3.8 shows the same matrix after applying the rank order clustering technique following *step 3*. Applying the rank order clustering rearranges the stations so that the generated $\widehat{\boldsymbol{A}}$ has a lower triangular form. It should be noted that the row and column of the output *Y* are excluded from ranking when applying the rank order clustering technique because *Y* does not represent a station in the line.

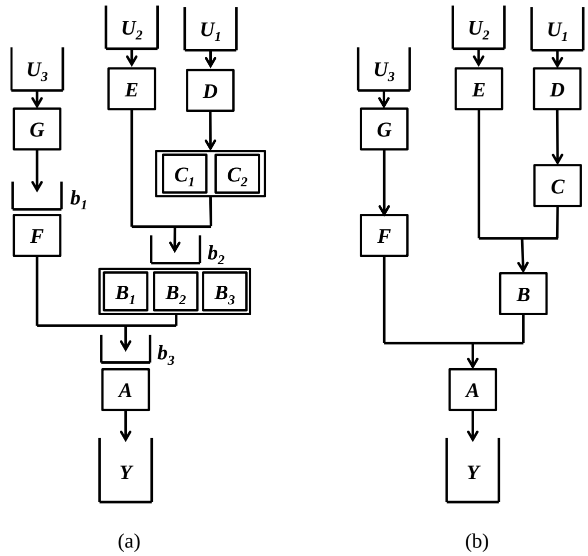
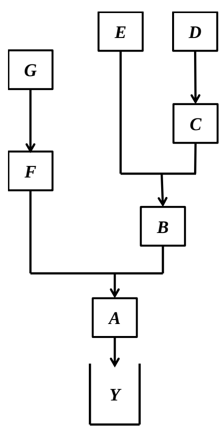


Figure 3.6 General flow line. (a) Line with parallel identical machines and buffers. (b) Line after simplification.

From the adjacency matrix in figure 3.8 and following *step 4*, the starting times vector of the different stations is given by:

$$\boldsymbol{X}_k = [X_{D,k}\ X_{G,k}\ X_{E,k}\ X_{C,k}\ X_{F,k}\ X_{B,k}\ X_{A,k}]^T$$



	Y	A	B	C	D	E	F	G
Y	0	1	0	0	0	0	0	0
A	1	0	1	0	0	0	1	0
B	0	1	0	1	0	1	0	0
C	0	0	1	0	1	0	0	0
D	0	0	0	1	0	0	0	0
E	0	0	1	0	0	0	0	0
F	0	1	0	0	0	0	0	1
G	0	0	0	0	0	0	1	0

Figure 3.7 A general flow line and its corresponding adjacency matrix.

	Y	A	B	F	C	E	G	D
Y	0	1	0	0	0	0	0	0
A	1	0	1	1	0	0	0	0
B	0	1	0	0	1	1	0	0
F	0	1	0	0	0	0	1	0
C	0	0	1	0	0	0	0	1
E	0	0	1	0	0	0	0	0
G	0	0	0	1	0	0	0	0
D	0	0	0	0	1	0	0	0

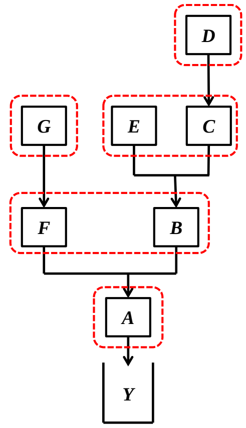


Figure 3.8 Adjacency matrix and its corresponding flow line diagram after re-arranging the rows and columns of the matrix

Following step 5, the ordered adjacency matrix is used along with equations (3.6), (3.9) and (3.10) to generate the $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ matrices for the simplified flow line. This step is automated and performed using the symbolic mathematical solver Wolfram Mathematica 6.0 (Grzymkowski, Kapusta et al. 2008) and the generated matrices are:

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} t_D & \varepsilon & & \cdots & & & \varepsilon \\ \varepsilon & t_G & \varepsilon & & & & \\ \varepsilon & \varepsilon & t_E & \varepsilon & & & \\ t_D^2 & \varepsilon & \varepsilon & t_C & \varepsilon & & \vdots \\ \varepsilon & t_G^2 & \varepsilon & \varepsilon & t_F & \varepsilon & \\ t_D^2 t_C & \varepsilon & t_E^2 & t_C^2 & \varepsilon & t_B & \varepsilon \\ t_D^2 t_C t_B & t_G^2 t_F & t_E^2 t_B & t_C^2 t_B & t_F^2 & t_B^2 & t_A \end{bmatrix}, \textit{ and } \widehat{\boldsymbol{B}} = \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e \\ \varepsilon & e & \varepsilon \\ t_D & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_G \\ t_D t_C & t_E & \varepsilon \\ t_D t_C t_B & t_E t_B & t_G t_F \end{bmatrix}.$$

Next, the parallel identical stations at stations *B* and *C* are modelled. Following section *3.3.3*, the equation for the line while taking into account the parallel identical stations becomes:

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A1}}\, \boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{AP}}_{\boldsymbol{C}}\, \boldsymbol{X}_{k-2} \oplus \widehat{\boldsymbol{AP}}_{\boldsymbol{B}}\, \boldsymbol{X}_{k-3} \oplus \widehat{\boldsymbol{B}}\, U_k \qquad (3.17)$$

where:

$$\widehat{\boldsymbol{A1}} = \begin{bmatrix} t_D & \varepsilon & & \cdots & & & \varepsilon \\ \varepsilon & t_G & \varepsilon & & & & \\ \varepsilon & \varepsilon & t_E & \varepsilon & & & \\ t_D^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & & \vdots \\ \varepsilon & t_G^2 & \varepsilon & \varepsilon & t_F & \varepsilon & \\ t_D^2 t_C & \varepsilon & t_E^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ t_D^2 t_C t_B & t_G^2 t_F & t_E^2 t_B & \varepsilon & t_F^2 & \varepsilon & t_A \end{bmatrix}, \widehat{\boldsymbol{AP}}_{\boldsymbol{C}} = \begin{bmatrix} \varepsilon & \varepsilon & & \cdots & & & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & & & & \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & & & \\ \varepsilon & \varepsilon & \varepsilon & t_C & \varepsilon & & \vdots \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \\ \varepsilon & \varepsilon & \varepsilon & t_C^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & t_C^2 t_B & \varepsilon & \varepsilon & \varepsilon \end{bmatrix},$$

$$\widehat{\boldsymbol{AP}}_{\boldsymbol{B}} = \begin{bmatrix} \varepsilon & \varepsilon & & \cdots & & & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & & & & \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & & & \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & & \vdots \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_B & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_B^2 & \varepsilon \end{bmatrix}, \textit{ and } \widehat{\boldsymbol{B}} = \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e \\ \varepsilon & e & \varepsilon \\ t_D & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_G \\ t_D t_C & t_E & \varepsilon \\ t_D t_C t_B & t_E t_B & t_G t_F \end{bmatrix}.$$

The final step is then to include the finite buffers by augmenting equation *(3.17)* with the matrices $\widehat{\boldsymbol{AB}}_{\boldsymbol{F}}$, $\widehat{\boldsymbol{AB}}_{\boldsymbol{B}}$ and $\widehat{\boldsymbol{AB}}_{\boldsymbol{A}}$ multiplied by $\boldsymbol{X}_{k-2-1}$, $\boldsymbol{X}_{k-2-1}$ and $\boldsymbol{X}_{k-4-1}$ respectively according to section 3.3.4. The final equations then become:

$$\begin{aligned} \boldsymbol{X}_k = \widehat{\boldsymbol{A1}}\, \boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{AP}}_{\boldsymbol{C}}\, \boldsymbol{X}_{k-2} \oplus \widehat{\boldsymbol{AP}}_{\boldsymbol{B}}\, \boldsymbol{X}_{k-3} \oplus \widehat{\boldsymbol{AB}}_{\boldsymbol{F}}\, \boldsymbol{X}_{k-3} \oplus \widehat{\boldsymbol{AB}}_{\boldsymbol{B}}\, \boldsymbol{X}_{k-3} \\ \oplus \widehat{\boldsymbol{AB}}_{\boldsymbol{A}}\, \boldsymbol{X}_{k-5} \oplus \widehat{\boldsymbol{B}}\, U_k \end{aligned} \qquad (3.18)$$

where $\widehat{\boldsymbol{A1}}$, $\widehat{\boldsymbol{AP}}_{\boldsymbol{C}}$, $\widehat{\boldsymbol{AP}}_{\boldsymbol{B}}$, and $\widehat{\boldsymbol{B}}$ are the same as in equation *(3.17)* and:

$$\widehat{\boldsymbol{AB}}_{\boldsymbol{F}} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_G & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_G t_F & \varepsilon & \varepsilon \end{bmatrix}, \widehat{\boldsymbol{AB}}_{\boldsymbol{B}} =$$

$$\begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_E \oplus t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & (t_E \oplus t_C)t_B & \varepsilon \end{bmatrix} ,and$$

$$\widehat{\boldsymbol{AB}}_{\boldsymbol{A}} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_B \oplus t_F \end{bmatrix}.$$

It should be noted that changing the number of parallel identical stations or buffer size for the finite buffers in equation *(3.18)* requires only changing the number subtracted from state vector multiplied by the corresponding matrix. For example changing the size of buffer b_3 from 4 to 6 will only change the term $\widehat{\boldsymbol{AB}}_{\boldsymbol{A}}\, \boldsymbol{X}_{k-5}$ in equation *(3.18)* to $\widehat{\boldsymbol{AB}}_{\boldsymbol{A}}\, \boldsymbol{X}_{k-7}$.

3.4. Case Study and Analysis

A case study is presented where three possible assembly system configurations for a back flushing control valve are modeled, analyzed and compared using max-plus equations generated by the developed method. Assembly lines for valves can be automated lines with moving pallets similar to the system presented in figure 3.9.

Figure 3.9 Automated Valve assembly line (Delta-Tech).

Figure 3.10 presents the 8 components of the back flushing control valve (Dorot (2001)). The assembly sequence tree of the valve (Kashkoush and ElMaraghy 2014) is presented in figure 3.11 (a) along with three possible assembly line configurations as shown in figure 3.11 (b, c and d). In the assembly sequence tree, each node represents an independent subassembly, therefor; assembling components 1 and 2 and components 6 and 7 and components 3 and 4 can all start simultaneously as no precedence relationship exists between them. Translating assembly sequences into possible line configurations depends on many factors such as available space, available number of workers, required tools for each operation etc. This is done using techniques for planning plant layout including optimization analysis.

The main component of the valve is the body which is component 3. Components 1 and 2, the bonnet and diaphragm are assembled to one side of the body while the rest of the components are assembled from the opposite side. The assembly line starts with the valve body moving on a pallet, the first assembly operation is to add component 4 which is the seat to the body then component 5; the guide cone, is added to the previous subassembly. In the next operation, the subassembly of components 6 and 7, which is already sub-assembled in a different station, is added to the body subassembly. Then component 8, the adapter, is added to the body subassembly and the valve is inverted to assemble the rest of the components on the opposite side. The final assembly operation is then to add the subassembly of components 1 and 2 to the body. All assembly operations are manual except for inverting the valve which is done by a robot.

The assembly line configurations in figure 3.11 (b) follow the same assembly sequence mentioned above but differ in assigning different operations to different stations. The assembly operations at each station and the corresponding required time for each configuration are given in table 3.1.

For the three configurations in figure 3.11, the stations with the same name perform the same exact assembly operations and require the same assembly time. The differences between the configurations are: 1) The assembly operations in stations *C* and *D* in configuration 1 are combined together in configuration 2 and performed in station C^*) The assembly operations in station *E* in configuration 1 are distributed over stations E^* and G in configuration 3. Combining the processes of two stations into one decreases the required number of workers but increases the total line cycle time. Since the three configurations are similar, detailed analysis is required to compare and choose the best among them.

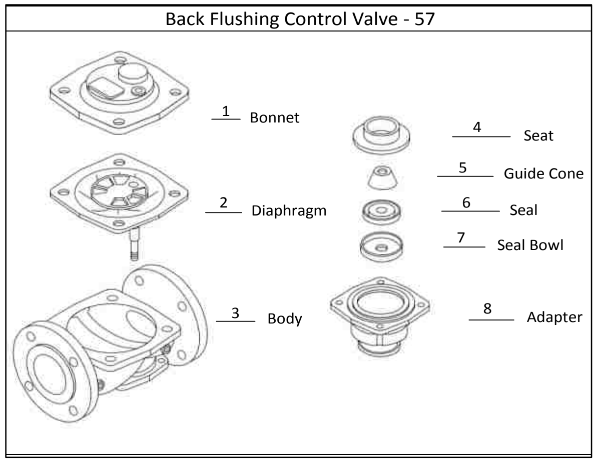


Figure 3.10 Back flushing control valve components (Dorot (2001)).

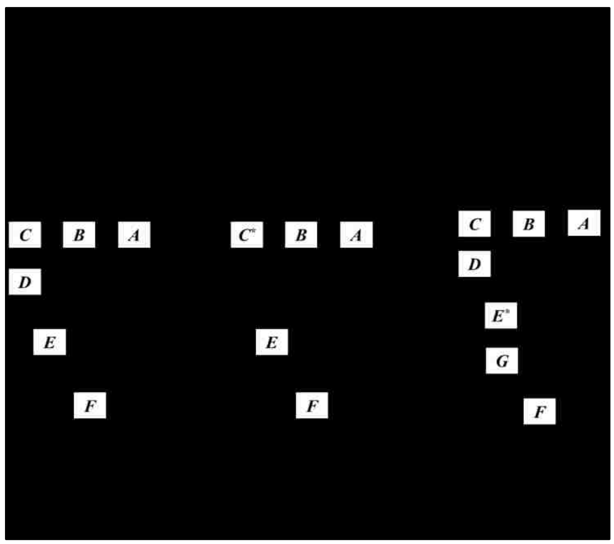


Figure 3.11 Assembly sequence tree (Kashkoush and ElMaraghy 2014) (a) and three possible corresponding assembly line configurations (b).

Following the procedure in section 3.3, the max-plus equations for three configurations, assuming buffers with equal sizes between all stations, are:

$$\boldsymbol{X}_k = \quad \widehat{\boldsymbol{A}}\, \boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{AB}}\, \boldsymbol{X}_{k-b} \oplus \widehat{\boldsymbol{B}}\, \boldsymbol{U}_k \tag{3.19}$$

where for configuration 1 $\boldsymbol{X} = [X_C\ X_D\ X_B\ X_E\ X_A\ X_F]^T$, for configuration 2 $\boldsymbol{X} = [X_{C^*}\ X_B\ X_{E^*}\ X_A\ X_F]^T$ and for configuration 3 $\boldsymbol{X} = [X_C\ X_D\ X_B\ X_E\ X_G\ X_A\ X_F]^T$. The values of $\widehat{\boldsymbol{A}}$, $\widehat{\boldsymbol{AB}}$ and $\widehat{\boldsymbol{B}}$ for each of the configurations are given in appendix C.

Table 3-1 Assembly processes and required processing times for stations in Figure 3.11 (b).

Station	Assembly processes	Required time in seconds
A	Assemble Bonnet and Diaphragm (Parts1 & 2)	$t_A = 43$
B	Assemble Seal and Seal Bowl (Parts 6 &7)	$t_B = 15$
C	Assemble Body and Seat (Parts 3 &4)	$t_C = 20$
D	Add Guide cone to Body and Seat (Part 5)	$t_D = 6$
E	Assemble Body subassembly with Seal subassembly then add the Adapter (Assemble (3,4,5) & (6,7) then add 8)	$t_E = 25$
F	Add Bonnet and Diaphragm to the assembly (Add (1,2) to (3,4,5,6,7,8))	$t_F = 21$
C^*	Assembly Body and Seat then add Guide cone. (Assemble 3 & 4 then add 5)	$t_{C*} = 28$
E^*	Assemble Body subassembly with Seal subassembly (Assemble (3,4,5) & (6,7))	$t_{E*} = 18$
G	Add Adapter to Body and Seal subassembly (Add 8 to (3,4,5,6,7))	$t_G = 5$

Using equation *(3.19)* and assuming $\boldsymbol{U_k}$ is given, the exact starting times for every station for every job can be obtained, where the k^{th} starting time on station *m* is given by $X_{m,k}$. For example, assuming stations *A*, *B* and *C* are never starved (i.e. $\mathbf{U_1} = [\mathbf{0\ 0\ 0}]^{\mathbf{T}}$ and $\mathrm{U_k} \leq [\mathrm{X_C\ X_B\ X_A}]^{\mathrm{T}}_{\mathrm{k-1}} + [\mathrm{t_C\ t_B\ t_A}]^{\mathrm{T}}$) and starting from an empty line ($i.e.\ \boldsymbol{X_0} = [-\infty\ -\infty\ -\infty]^{\boldsymbol{T}}$), then for the given values of stations processing times, the starting times for all stations for configuration 1 with buffers size of 2 will be given by table 3.2.

Let I_m and I_t be the idle time for station *m* and the total idle time in the whole line respectively, then for a cycle of *k* jobs and a total number of stations *M* in any line configuration:

$$I_m = \sum_{i=1}^{k} X_{m,i+1} - X_{m,i} - t_m \quad (3.20)$$

$$I_t = \sum_{j=1}^{M} I_m \quad (3.21)$$

Using equation *(3.21)*, I_t can be easily calculated for the three configurations for different sizes of buffers. Figure 3.12 shows a plot of I_t for the considered three assembly line configurations in figure 3.10 for different sizes of buffers. Figure 3.12 shows that line idle time decreases with increasing the size of buffers up to a certain critical size after which further increase has no effect. The critical buffer size for the three configurations can be obtained from figure 3.12 along with other less intuitive results such that configuration 1 is the least affected by changes in size of buffers.

Table 3-2 Starting times for 10 jobs for configuration 1.

	k=1	**k=2**	**k=3**	**k=4**	**k=5**	**k=6**	**k=7**	**k=8**	**k=9**	**k=10**
X_C	0	20	40	60	80	100	120	140	160	180
X_D	20	40	60	80	100	120	140	160	180	200
X_B	0	15	30	45	60	76	101	126	151	176
X_E	26	51	76	101	126	151	176	201	226	251
X_A	0	43	86	129	172	215	258	301	344	387
X_F	51	86	129	172	215	258	301	344	387	430

Another useful application of equation *(3.19)* is in evaluating the effect of changing the stations processing times on a given performance measure. This is very useful in the design stages when the exact processing time for a given part on a given station is unknown and the system designers want to know the effect of variation in processing time on the performance of the line. It can also be used during the system operation phase to assess the merits and trade-off of buying new equipment or conducting workers training which would decrease the station's processing time. In order to accomplish such objective, equation *(3.19)* is evaluated as a function of each one of the stations processing times. The result would be a table similar to table 3.2 but function of a given processing time. For example, by evaluating equation *(3.19)*, for configuration 3 with buffers size of 2, as a function of t_{E*} then $X_{F,5}$ is given by:

$$X_{F,5} = \begin{cases} 215, & 5t_{E^*} \leq 184 \\ 31 + 5t_{E^*}, & 5t_{E^*} > 184 \end{cases}$$

Equations *(3.20)* and *(3.21)* can then be used to find the total line idle time as a function of t_{E*} and plot it as a continuous function. Figure 3.13 shows a plot of the total line idle time for configuration 3 for three different sizes of buffers as a function of t_{E*}.

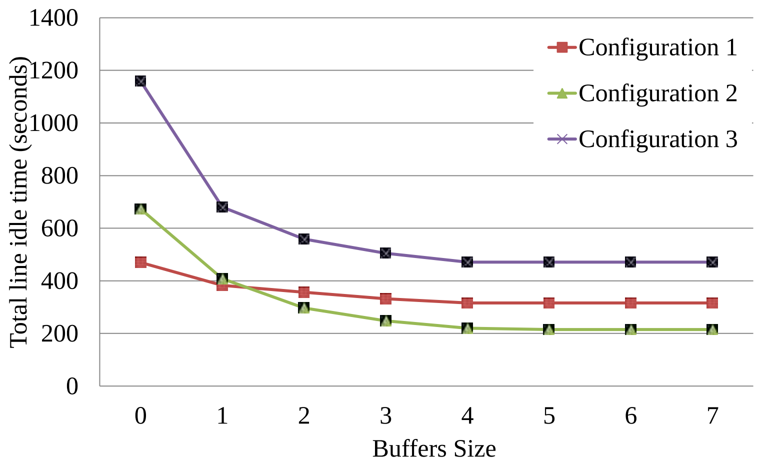


Figure 3.12 The effect of buffers size on total line idle time for the three line configurations given in figure 3.10.

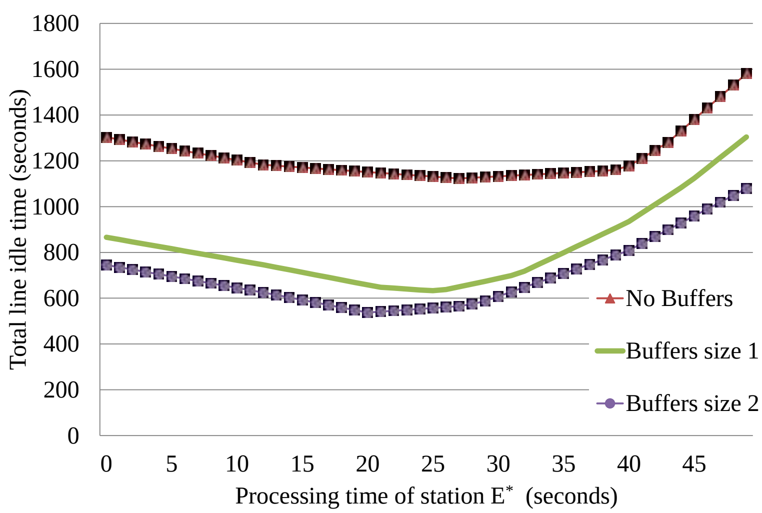


Figure 3.13 Total line idle time for configuration 3 as a function of the processing time of station E^*.

Using figure 3.13 designers can easily determine the optimal station's processing time that minimizes the total assembly line idle time. Table 3.3 summarizes the results from figure 3.13 listing the optimal processing time for station *E** for each case and the corresponding line idle time.

Table 3-3 Summary of results from Figure 3.13

	NO BUFFERS	BUFFERS SIZE 1	BUFFERS SIZE 2
OPTIMAL T_{E*} (SECONDS)	27	24.8	20
LINE IDLE TIME (SECONDS)	1123	632.5	537

It should be noted that line idle time is not the only performance measure that can be evaluated after solving equation *(3.19)*. Other possible performance measures include but are not limited to line lead time, throughput and stations utilization.

The data presented in figures 3.12 and 3.13 can be obtained using discrete event simulation, however; it would require constructing three different simulation models, performing tens of complete simulation runs for each model and then extracting the data from each simulation run to be plotted together, a process that is very time consuming and tedious.

3.5. Discussion and Conclusions

A new method was developed for quick and efficient generation of the max-plus equations for flow lines of any size and structure, while taking into consideration finite buffers and parallel identical stations. The method is based on the observation that a flow line can be decomposed into different additive 'features' each of which uniquely affects the final equations. These features can be integrated sequentially to form the final system equations. The correctness of all generated equations was verified by comparing the results with discrete event simulation models equivalent to each of the examples presented in this chapter. The results from the max-plus model and simulation were identical since the processing times of all stations were deterministic. The discrete event simulation software used was FlexSim (Beaverstock, Greenwood et al. 2011).

With the help of the method presented in this chapter for automatic generation of max-plus equations, the use of max-plus algebra in modeling, performance evaluation and control of manufacturing systems can be extended to large and complicated manufacturing systems.

The ability to generate max-plus equations quickly is useful in both design and operation phases of manufacturing systems. In the system design phase, easy generation of equations enables analyzing and comparing any number of possible line configurations in an efficient manner. It can also give insights into the effects of adding buffers to the system or changing buffers sizes on various system performance measures. In the operation phase, it can be used to analyze possible line improvements and line reconfigurations due to product changes. These uses were demonstrated by analyzing three possible configurations of an assembly line of a back flushing control valve. The max-plus equations for each configuration were generated then used to analyze the effect of buffer size and changes in assembly times on the total line idle time. Generating the equations took only few minutes and then using these equations, the data used in analyses were obtained in few seconds. Generating discrete event simulation models and conducting simulation runs to obtain equivalent data would have required days or even weeks.

The developed method requires only a user interface to be made available as a simulation and analysis tool that can be used without requiring any knowledge of max-plus algebra. The resulting tool would only require the user to input the structure of the line, which can be done using drag and drop components such as machines, transporters, and/or buffers as well as the time required at each machine and the capacity of buffers, etc. Users could then generate various analyses showing how different performance measures change with any of the system parameters.

CHAPTER 4: MAX-PLUS MODELING OF MIXED-MODEL ASSEMBLY LINES

4.1. Introduction

Mass production thrived in an era when low prices were enough to satisfy customers, but currently customer satisfaction requires product variety and quality in addition to a competitive price. Mixed Model Assembly Lines (MMAL) are lines that can handle more than one model of the same general product simultaneously and continuously (not in batches) on the same line (Thomopoulos 1967; Groover 2007). Due to their ability to offer product variety at a competitive cost, MMALs have become the most common assembly systems in the fields of automotive industry, consumer electronics, furniture and clothing (Hu, Ko et al. 2011).

The two main types of work transport in MMALs are continuous transport and intermittent or synchronous transport. In continuous transport lines, the work units are usually fixed on the conveyor which moves continuously at a constant speed and workers walk downstream with the work unit while doing the assembly operations then walk upstream to work on the next work unit. In intermittent or synchronous transport; the whole line moves to position work units at the next stations then stays stationary for a period of dwell time to allow workers to perform the assembly operations before the cycle is repeated. The dwell time is the same for all stations and is called the takt time. It is equal to the largest assembly time required in any station and thus the line often suffers significant idle time. In continuous transport lines, extra space can be provided to those stations which require more time without affecting other stations.

A typical example of MMALs is the assembly of automotive car seats. Car seats are usually assembled in a pull system where the OEM gives the order with exact seat colors and specs then the seats are assembled according to the order. Some models require significantly more assembly time like for example power adjustable and heated seats. In these cases, line balancing problems arise even when using continuous transport lines and the effective utilization of the line requires sequencing the models with higher work load apart and assembling other models that require less time in between. The problem of finding the optimum order of models on the line to satisfy a given demand mix is called the "Model Sequencing" problem (Thomopoulos 1967; Gökcen and Erel 1998). The first on MMAL sequencing appeared in 1963 by Kilbridge and Wester (Kilbridge and Wester 1963) and since then the field has been thriving with publications addressing the problem from many different angles. Important literature include the work by

Thomopoulos (Thomopoulos 1967), Bard et al. (Bard, Dar-El et al. 1992), Hyun (Hyun, Kim et al. 1998) and Miltenburg (Miltenburg and Sinnamon 1989). A recent survey and classification of sequencing models are presented by Boysen et al (Boysen, Fliedner et al. 2009). Recent publications on MMALs sequencing address the problems of integrating line balancing and sequencing (Uddin, Soto et al. 2010), minimizing the number of work overload stations (Boysen, Kiel et al. 2011), sequencing MMALs under a just in time approach (Tavakoli and Fattahi 2012), the combination of planning methods for sequenced lines (Matyas and Auer 2012) and sequencing low volume high mix production (Bohnen, Buhl et al. 2013).

One common feature in the majority of available sequencing techniques is that they all assume deterministic times for the given assembly tasks (Boysen, Fliedner et al. 2008). If assembly times vary then different analysis methods should be used. Assembly times can be non-deterministic in manual assembly - which is the case in many MMALs - where the exact assembly time cannot be accurately determined a priori or in operations that require skill and different workers may require different time periods. Stochastic times have been considered in very few cases (Chutima, Nimmano et al. 2003; Boysen, Fliedner et al. 2009), however, it still does not consider inaccuracies arising when actual assembly times t are significantly different from the estimated ones.

A closely related but different problem is assessing possible line improvements where the assembly times of certain tasks can be reduced through investment in workers training or better equipment. Reducing assembly time of some tasks might be profitable up to a certain point after which more reduction leads to workers' idle time without increasing throughput. The same applies to adjusting different line parameters such as the conveyor speed and the launching rate of products on the line.

Therefore, it is required to assess some performance measures of MMALs as a function of the line parameters such as assembly times of different tasks, setup times and launching rate of jobs on the line. If MMALs are modeled by max-plus equations, it would be possible to evaluate several performance measures as a function of line parameters of interest and thus, assessing the effect of changing these parameters on the performance of the line would be possible leading to more realistic improvements and better performance. Max-plus algebra is a complementary tool that is used alongside with whatever analysis and optimization models that an engineer would use for sequencing.

Continuous transport MMALs are usually divided into stations with specific assembly activities assigned to each station. Stations can be closed, open from both sides or open from only one side. A closed station is a station that has fixed boundaries and the worker(s) has to finish the assembly operation within these boundaries. This is usually the case when the assembly operation requires power tools that have limited reach. An open station on the contrary has no boundaries, and the worker in that station can start working on the work unit when s/he is ready. Stations in a line can be all of one type or can be mixed with some stations closed and others open. A detailed description of closed and open stations will be presented in section 4.2.

In this chapter, max-plus algebra is first used to model MMALs with closed and open stations. Using these models, a complete characterization of the line can be obtained in a parametric form, namely the position at which each worker starts and finishes working on each job, which determines the station length, can be obtained as a function of the processing time and launching rate of these jobs. Using these models, several performance measures, such as length of each station, total line length and workers idle time, are evaluated as a function of the assembly time of different tasks, changeover time and launching rate of jobs on the line. A numerical case study is presented to demonstrate how the developed equations can be used to solve the problem of assessing the optimality of a given sequence over a range of assembly times as well as the problem of analyzing effect of changes in the line parameters on the performance of the line. For the first problem the best sequences of the required product mix obtained by optimization for a given case study are compared for varying assembly times and thus ranges of better performance for different sequences can be determined. For the second problem, assessing the line performance as a function of jobs launching rate for different models is demonstrated.

The rest of the chapter is organized as follows: section 4.2 presents step by step modeling of MMALs using max-plus equations for lines with either closed or open stations, section 4.3 includes a numerical case study for both mentioned problems, section 4.4 presents an industrial case study to showcase the usefulness of the developed models, and finally the discussion and conclusions are provided in section 4.4.

4.2. Modeling MMALS

The most common structure of a MMAL is a conveyor moving with a constant speed with jobs fixed to it. The rate of launching jobs to the line can be fixed or variable. In Fixed products launching rate, which is the most common policy, the time between launching jobs on the line is equal to the weighted average of all assembly times over the required demand of products. In variable rate launching the time between launching is usually equal to the assembly time of the previous job on the first station. A study has been published on variable rate launching where the launching rate is included as a variable for the optimization problem, the developed model produces better results when compared to fixed rate launching, but it was not compared to variable rate launching using the assembly time of the first station (Fattahi and Salehi 2009).

In both fixed or variable rate launching, workers in each station walk downstream with the conveyor while executing their assembly tasks then walk back upstream to the next job (Thomopoulos 1967). Stations can either be closed or open, in closed stations workers are assigned a space that they cannot exceed and the assembly activities have to be finished within these boundaries. Figure 4.1 shows a diagram that demonstrates the workers movement in a MMAL with two closed stations. A worker can only start working on a job when it enters the boundaries of his station, then s/he moves downstream with the job while performing the assembly activities. After completing work on a given work unit, the worker walks upstream until s/he meets the next work unit and starts working on it. The walk back distance is always constant and equal to the distance between launching work units on the line. If the worker reaches the station boundary when moving upstream, s/he has to stay idle until the work unit enters the station. The length of a station is a function of the assembly operations required in the station, the launching rate of work units on the line, and the sequence of models on the line.

In open stations, workers do not have boundaries to their stations, they keep moving with the work unit until their assembly tasks are finished, then they walk upstream until they reach the next work unit and start working on it as long as the previous worker is done with the work unit (Bard, Dar-El et al. 1992). Figure 4.2 shows a diagram that demonstrates the workers movement in a MMAL with two open stations. The worker in the first station can only be idle if s/he reaches the beginning of the line before the launch of the next work unit, while the worker in later stations can be idle if s/he walks upstream and reaches the work unit before the previous worker has finished working on it as can be seen in the second and the last work units for the worker in station 2.

In sequencing problems, a *minimum part set* (MPS) strategy is usually employed where the minimum part set is the smallest possible set having the same proportion as the required demand mix (Bard, Dar-El et al. 1992). For example if the mix contains three variants a, b, and c and the demand is 600 of part a, 400 of part b and 300 of part c, then the minimum part set is 6 units of part a, 4 units of part b, and 3 units of part c. Thus instead of sequencing 1300 parts, a sequence is obtained for the MPS which is only 13 parts then repeated 100 times.

The following notations will be used in this chapter:

N: Total number of stations in line.

n: Station number in the line, $n = 1 \rightarrow N$.

M: Number of different models in the line.

m: Model type number, $m = 1 \rightarrow M$.

K: Number of jobs in the sequence (Length of MPS).

v: Speed of the conveyor.

v_o: Walking speed of workers.

l_t: Launching time of line (time between launching products on the line).

w: Upstream walking distance $w = v\, l_t(v_o/(v + v_o))$.

$t_{n,k}$: time to assemble model k in station n.

$l_{n,k}$: Distance on line required to assemble model k in station n, $l_{n,k} = v \times t_{n,k}$.

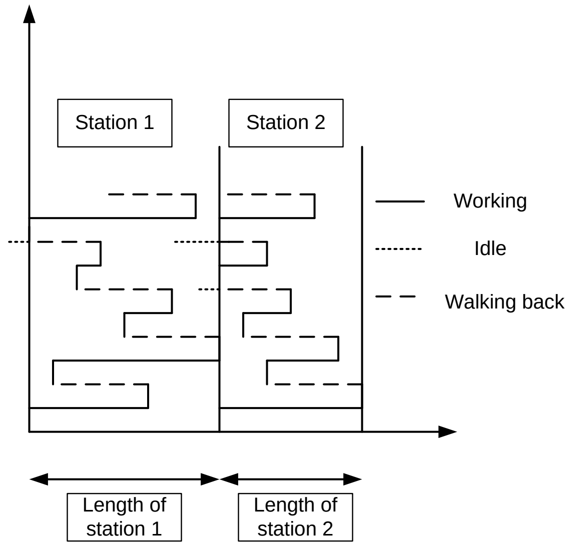


Figure 4.1 Worker movement in a continuous transport MMAL with closed stations.

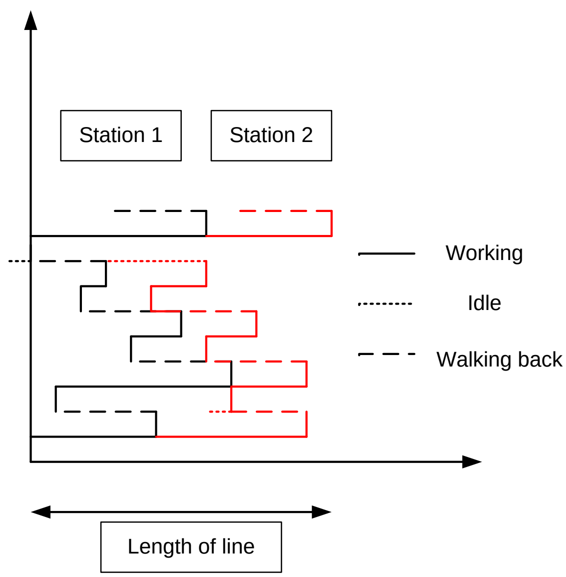


Figure 4.2 Worker movement in a continuous transport MMAL with open stations.

When walking speed of workers is much faster than speed of the conveyor (v_o>>v), the upstream walking distance becomes: $w = v\, l_t$. If the conveyor speed is normalized to 1 then $t_{n,k} = l_{n,k}$ and $w = l_t$.

In the following subsections max-plus models will be developed for lines with closed stations (section 4.2.1) then for lines with open stations (section 4.2.2).

4.2.1. Closed Stations

In closed stations with fixed rate launching each worker has a specific working area that cannot be exceeded. After completing the work on job *k-1*, the worker walks back towards the starting edge of his station, s/he either walks a distance $w = l_t$ and starts working on the next job k as in figure 4.3(a), or reaches the station's edge and remains idle until the next job reaches his station as in figure 4.3(b).

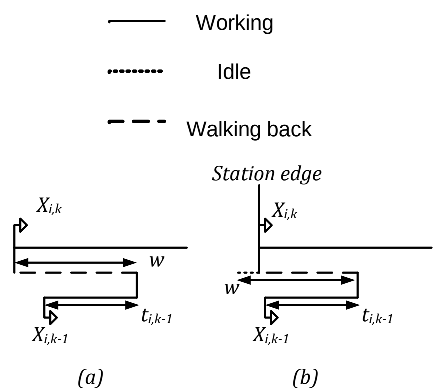


Figure 4.3 Job starting scenarios for workers in a closed station MMAL.

Let $X_{n,k}$ be the position, relative to starting edge of station n, where the worker starts working on job k, then:

$$X_{n,k} = max(X_{n,k-1} + t_{n,k-1} - w, 0) \tag{4.1}$$

In max-plus algebra equation *(4.1)* is written as:

$$X_{n,k} = \frac{t_{n,k-1}}{w} X_{n,k-1} \oplus e \tag{4.2}$$

and for the whole line:

$$\boldsymbol{X}_k = \boldsymbol{A}_k\, \boldsymbol{X}_{k-1} \oplus \boldsymbol{B} \tag{4.3}$$

where,

$$\boldsymbol{X}_k = \begin{bmatrix} X_{1,k} \\ X_{2,k} \\ \vdots \\ X_{N,k} \end{bmatrix}, \ \boldsymbol{A}_k = \begin{bmatrix} \frac{t_{1,k-1}}{w} & \varepsilon & \dots & \dots & \varepsilon \\ \varepsilon & \frac{t_{2,k-1}}{w} & \varepsilon & \dots & \varepsilon \\ \vdots & \varepsilon & \ddots & & \vdots \\ \vdots & \ddots & & \ddots & \varepsilon \\ \varepsilon & \varepsilon & \dots & \varepsilon & \frac{t_{N,k-1}}{w} \end{bmatrix}, \textit{and } \boldsymbol{B} = \begin{bmatrix} e \\ e \\ \vdots \\ e \end{bmatrix}.$$

Using equation *(4.3)* and assigning $\boldsymbol{X}_1 = [0 \ 0 \ \cdots \ 0\,]^T$, the starting point of each job in each station can be obtained.

Knowing the starting position of each job in each station enables us to compute any required performance measure.

Let L_n, L_T, I_k and I_T be station *n*'s length, total line length, idle time associated with job *k* and total line idle time respectively, then:

$$L_n = max\ \left(X_{n,1} + t_{n,1},\, X_{n,2} + t_{n,2},\ \dots\ , X_{n,K} + t_{n,K}\right) = \bigoplus_{k=1}^{K} (t_{n,k} X_{n,k}) \qquad (4.4)$$

$$L_T = \sum (\,L_1,\ \ L_2, \dots,\ \ L_N) = \bigotimes_{n=1}^{N} L_n \qquad (4.5)$$

$$I_k = \sum_{n=1}^{N}\left(max\left(-(X_{n,k-1} + t_{n,k-1} - w),0\right)\right) = \bigotimes_{n=1}^{N} (-\boldsymbol{A}_k\, \boldsymbol{X}_{k-1} \oplus \boldsymbol{B}) \qquad (4.6)$$

$$I_T = \sum (\,I_1,\ \ I_2, \dots,\ \ I_K) = \bigotimes_{k=1}^{K} I_k \qquad (4.7)$$

It should be noted that in equation *(4.6)*, the idle time is calculated as the distance the worker would have walked past the station boundaries to reach the next job which is represented by dotted line in figure 4.3(b). The idle time is equal to that distance since the conveyor speed is normalized to one.

4.2.2. Open stations

In open stations with fixed rate launching, the worker in the first station acts exactly the same as in closed stations, whenever s/he finishes working on a job s/he walks back and either start working on the next job or waits at the beginning of the line for the next job to launch. On the other hand, a worker in station *n*, where *n* >1, can start working on a job *k* if the previous worker has completed working on it and s/he has finished working on job *k-1*.

These conditions can be expressed as:

$$X_{n,k} = \begin{cases} \frac{t_{1,k-1}}{w} X_{1,k-1} \oplus e\,, & n = 1 \\ t_{n-1,k} X_{n-1,k} \oplus \frac{t_{n,k-1}}{w} X_{n,k-1}, & n > 1 \end{cases}$$

which in matrix form can be written as:

$$\boldsymbol{X}_k = \boldsymbol{A}_k\, \boldsymbol{X}_k \oplus \boldsymbol{B}_k\, \boldsymbol{X}_{k-1} \oplus \boldsymbol{C} \qquad (4.8)$$

where,

$$\boldsymbol{X}_k = \begin{bmatrix} X_{1,k} \\ X_{2,k} \\ \vdots \\ X_{N,k} \end{bmatrix}, \boldsymbol{A}_k = \begin{bmatrix} \varepsilon & \varepsilon & \dots & \varepsilon & \varepsilon \\ t_{1,k} & \varepsilon & \dots & \varepsilon & \varepsilon \\ \varepsilon & t_{2,k} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \varepsilon & \varepsilon \\ \varepsilon & \dots & \varepsilon & t_{N-1,k} & \varepsilon \end{bmatrix},$$

$$\boldsymbol{B}_k = \begin{bmatrix} \frac{t_{1,k-1}}{w} & \varepsilon & \dots & \dots & \varepsilon \\ \varepsilon & \frac{t_{2,k-1}}{w} & \varepsilon & \dots & \varepsilon \\ \vdots & \varepsilon & \ddots & & \vdots \\ \vdots & \ddots & & \ddots & \varepsilon \\ \varepsilon & \varepsilon & \dots & \varepsilon & \frac{t_{N,k-1}}{w} \end{bmatrix}, \; and\; \boldsymbol{C} = \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix}.$$

Equation *(4.8)* is an implicit equation in $\boldsymbol{X}_k$ and according to theorem (2.1) it can be transformed into:

$$\boldsymbol{X}_k = \widehat{\boldsymbol{A}}_k\, \boldsymbol{X}_{k-1} \oplus \widehat{\boldsymbol{B}}_k \qquad (4.9)$$

where:

$$\widehat{\boldsymbol{A}}_k = \boldsymbol{A}_k^* \boldsymbol{B}_k\,,\; and\;\; \widehat{\boldsymbol{B}}_k = \boldsymbol{A}_k^* \boldsymbol{C}.$$

According to equation *(2.3)*, $\boldsymbol{A}_k^*$ can be calculated as:

$$\boldsymbol{A}_k^* = e \oplus \boldsymbol{A}_k \oplus {\boldsymbol{A}_k}^2 \oplus \dots \oplus {\boldsymbol{A}_k}^{\infty} \qquad (4.10)$$

which according to equation *(2.4)* can be reduced to:

$$\boldsymbol{A}_k^* = e \oplus \boldsymbol{A}_k \oplus \boldsymbol{A}_k{}^2 \oplus \ldots \oplus \boldsymbol{A}_k{}^N = \begin{bmatrix} 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ t_{1,k} & 0 & \varepsilon & \varepsilon & \varepsilon \\ t_{1,k}t_{2,k} & t_{2,k} & \ddots & \varepsilon & \varepsilon \\ \vdots & \ddots & \ddots & 0 & \varepsilon \\ t_{1,k}t_{2,k}\ldots t_{N-1,k} & t_{2,k}\ldots t_{N-1,k} & \ldots & t_{N-1,k} & 0 \end{bmatrix} \quad (4.11)$$

Using equation *(4.11)*, $\widehat{\boldsymbol{A}}_k$ and $\widehat{\boldsymbol{B}}_k$can be calculated as:

$$\widehat{\boldsymbol{A}}_k = \begin{bmatrix} \frac{t_{1,k-1}}{w} & \varepsilon & \varepsilon & \varepsilon \\ \frac{t_{1,k}t_{1,k-1}}{w} & \ddots & \varepsilon & \varepsilon \\ \vdots & \ddots & \frac{t_{N-1,k-1}}{w} & \varepsilon \\ \frac{t_{1,k}\ldots t_{N-1,k}t_{1,k-1}}{w} & \ldots & \frac{t_{N-1,k}t_{N-1,k-1}}{w} & \frac{t_{N,k-1}}{w} \end{bmatrix} \text{and}$$

$$\widehat{\boldsymbol{B}}_k = \begin{bmatrix} 0 \\ t_{1,k} \\ t_{1,k}t_{2,k} \\ \vdots \\ t_{1,k}t_{2,k}\ldots t_{N-1,k} \end{bmatrix}.$$

Again, using equation *(4.9)* and assigning $\boldsymbol{X}_0 = [\varepsilon \;\; \varepsilon \;\; \cdots \;\; \varepsilon\,]^T$, the starting position, relative to the line beginning, of each job can be obtained and similar to the closed stations case this enables computing any required performance measure.

Let L_T, I_k and I_T be total line length, idle time associated with job k and total line idle time respectively, then:

$$L_T = max(X_{N,1} + t_{N,1},\, X_{N,2} + t_{N,2}, \ldots, X_{N,K} + t_{N,K}\,) \quad (4.12)$$

$$I_k = \sum_{n=1}^{N} \big(X_{n,k} - (X_{n,k-1} + \; t_{n,k-1} - w)\big) \quad (4.13)$$

$$I_T = \sum_{k=2}^{K} I_k \quad (4.14)$$

Using the same procedure, variable rate launching as well as lines with mixed open and closed stations can be modeled using similar max-plus equations.

4.3. Numerical Examples

In this section a numerical example will be used to show how the derived equations in section 4.2 can be used in: 1) comparing sequences of the demand mix and determining ranges of assembly times over which certain sequences are better, and 2) analysis of various performance measures as a function of MMAL parameters. The analysis will be conducted using a line presented by Bard et al. in (Bard, Dar-El et al. 1992). The line containing four stations assembling three different product models with a minimum part set (*MPS)* of five units of model 1, three units of model 2 and two units of model 4, i.e. the MPS is (5,3,2,). Table 4.1 gives the assembly times for each model in each station normalized for conveyor velocity *v=1* and launching time of products L_t= 6. Moreover, since *v=1*, then $w=L_t=6$.

Table 4-1 Assembly times for each model in each station.

Model	Station			
	1	**2**	**3**	**4**
1	4	6	8	4
2	8	9	6	7
3	7	4	6	5

4.3.1. Comparing Sequences

Table 4.2 presents the optimal sequence of products given the assembly times in table 4.1 and the *MPS* of (5,3,2) for a line with open stations and a line with closed stations as obtained in (Bard, Dar-El et al. 1992). Each of the sequences in table 4.2 is optimal for the given objective and the assembly times in table 4.1, however, the robustness of these sequences and their sensitivity to changes in the assembly times is not given.

To check the robustness of the sequences in table 4.2 and their sensitivity to changes in assembly times, the max-plus equations are derived for a line with closed stations and another time for a line with open stations. Then the line lengths are evaluated for the optimal sequences as well as other sequences and plotted as a function of some assembly task times.

Table 4-2 Optimal sequence for system parameters in Table 4.1 as obtained from (Bard, Dar-El et al. 1992).

	Job #	**Closed stations**	**Open stations**
Optimal sequence	1	2	1
	2	1	1
	3	1	1
	4	3	2

	5	1	1
	6	2	3
	7	1	2
	8	3	3
	9	1	2
	10	2	1

The tested sequences for each case are presented in table 4.3 where S2 for each case is the optimal sequence for the values given in table 4.1.

Table 4-3 Sequences to be compared.

	Closed stations			Open stations		
Job #	S1	S2	S3	S1	S2	S3
1	1	2	3	1	1	1
2	2	1	1	3	1	1
3	3	1	2	1	1	1
4	1	3	1	2	2	2
5	2	1	2	1	1	2
6	1	2	1	3	3	2
7	3	1	3	1	2	1
8	1	3	1	2	3	1
9	2	1	2	1	2	3
10	1	2	1	2	1	3

Equations (4.5) and (4.12) which give the total line length for lines with closed and open stations respectively can be evaluated using the data in table 4.1 while keeping the value of one of the assembly times as a variable. The resulting equation can then be evaluated and plotted for different values of that assembly time. Figures 4.4 and 4.5 show a plot of the total line length as a function of four assembly times for the case of closed and open stations respectively.

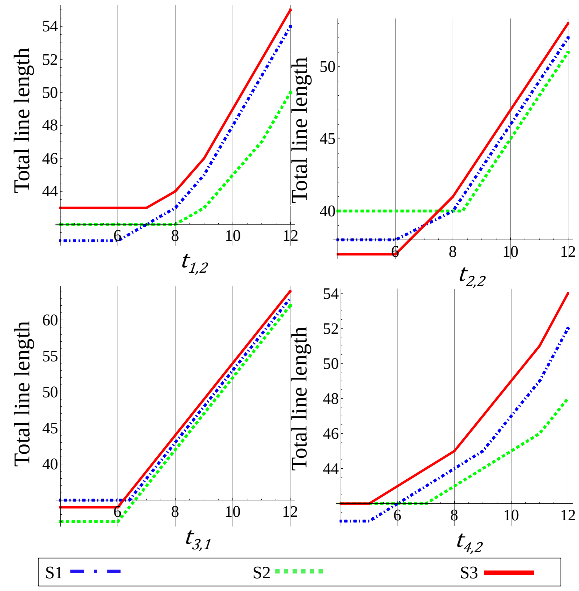


Figure 4.4 Total line length as a function of assembly times $\boldsymbol{t_{1,2}}$, $\boldsymbol{t_{2,2}}$, $\boldsymbol{t_{3,1}}$ and $\boldsymbol{t_{4,2}}$ for closed stations MMAL.

It is clear from figures 4.4 and 4.5 that the sequences obtained by optimization for the values given in table 4.1 are not robust and are sensitive to changes in the assembly time of different tasks. For example, in the case of closed stations the sequence in table 4.2 is optimal only for values of $t_{2,2}$ that are greater than or equal to 8. When the value of $t_{2,2}$ is between 7 and 8, the best sequence becomes S2 in table 4.3. For values of $t_{2,2}$ below or equal to 7, S3 becomes the best one. Figures 4.4 and 4.5 also point out which assembly times have small or no effect on the best sequence as is the case with $t_{3,1}$ in both the open and closed stations cases. This information can be very useful in many situations, like for example when certain stations have workers with significantly different skill levels working in different shifts, in this case different sequences of products should be used in different shifts to assure optimality and reduce waste. It can also be very useful when introducing new changes to a certain station like better tools to training for the workers, the above information can show if the change in the station's assembly time would require changes in the sequence or not.

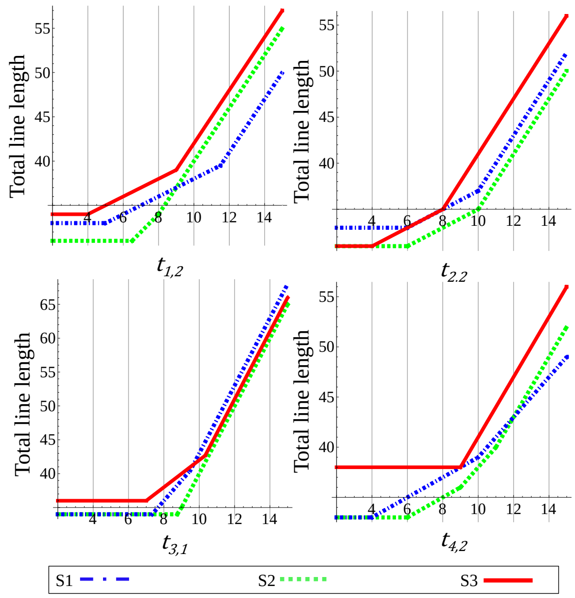


Figure 4.5 Total line length as a function of assembly times $\boldsymbol{t_{1,2}}$, $\boldsymbol{t_{2,2}}$, $\boldsymbol{t_{3,1}}$ and $\boldsymbol{t_{4,2}}$ for open stations MMAL.

4.3.2. Parametric analysis of MMALs

In this section the same example used in section 4.3.1 will be used to show how max-plus models can be used in assessing the effect of changing some system parameters on the overall performance of the line.

The system parameter that will be studied is the section is the launching timel_t of the line. Again, because the conveyor speed is normalized, the launching time l_t is equal to the distance between jobs on the line and consequently equal to the walk back distance w which is one of the parameters in equations *(4.5)* and *(4.12)*, so using these equations, the relationship between the launching rate and several line performance measures can be obtained. It should be noted that if the conveyor speed is not normalized, the same equations can still be used to relate launching time to performance measures by replacing w with $v \times l_t$ where v is the conveyor speed.

When l_t is increased jobs become widely spaced on the line, a direct consequence is increasing the walk back distance w and thus workers have more opportunity to walk back towards the beginning of their stations and start working on jobs at earlier positions in their stations. On the other hand, when L_t is decreased, jobs become closely spaced on the line, the walk back distance diminishes and workers start working on jobs at later positions in their stations, this of course leads to increasing the length of stations and consequently the whole length of the line.

Figure 4.6 shows where workers work on different jobs for the flow line with closed stations with the sequence S1 and a launching time l =4 for figure 4.6 (a) and l_t =8 for figure 4.6 (b). The huge difference in line length between the two cases is clear from the figure. The same applies also for lines with open stations as can be seen in figure 4.7, although the difference is not as significant as in the closed stations case since the concept of open stations in itself reduces the length of lines by allowing workers to start working on a job immediately after the pervious worker is done with it.

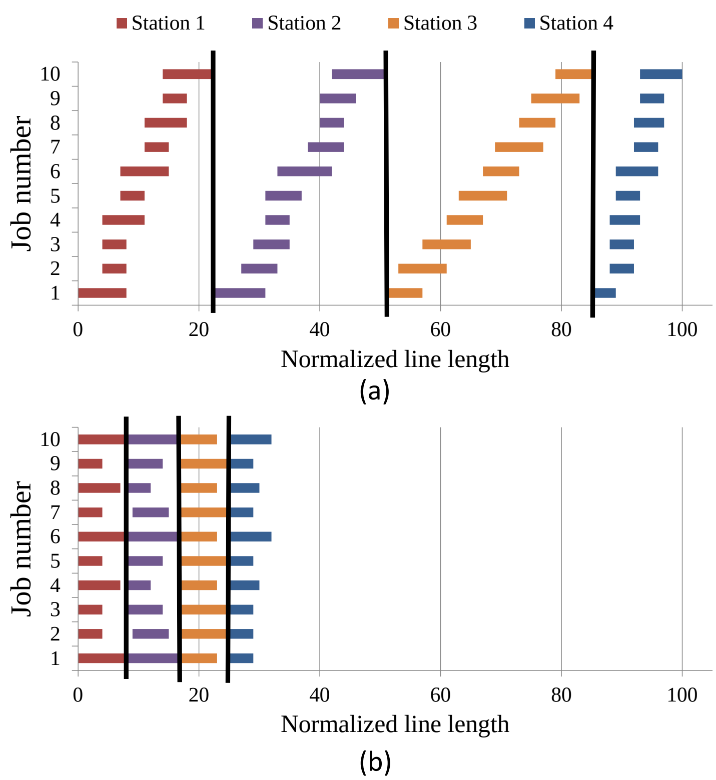


Figure 4.6 Starting position of each job on the line for closed stations with $\boldsymbol{l_t}$= 4 (a) and l=8 (b).

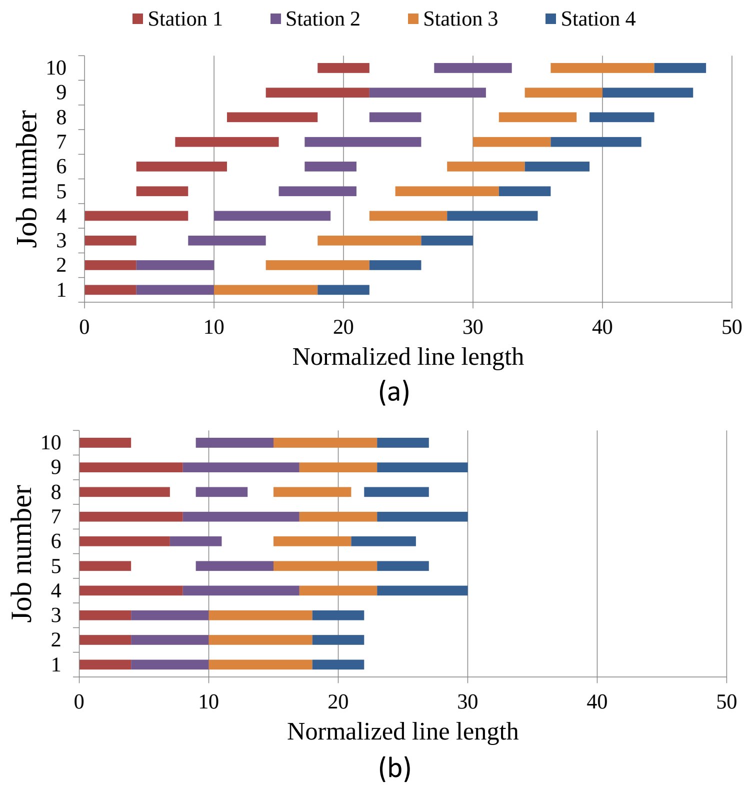


Figure 4.7 Starting position of each job on the line for open stations with l_t= 4 (a) and l_t=8 (b).

From figures 4.6 and 4.7 it is clear that changing l_t affects the length of the line; however, they do not give the full relationship between l_t and the line length. Furthermore, the total line length is not the only system performance measure affected by changing l_t . Other more important but less visible measures are the total line idle time and the line throughput.

The effect of changing L_t on the line idle time can be intuitive, the more l_t is decreased the less idle time there is for workers on the line. However, this is true only for certain l_t after which further decrease has no effect on line idle time and leads only to increase in the line length. Using equations *(4.5)*, *(4.7)*, *(4.12)*, and *(4.14)* and recalling that $l_t = w$, the effect of changing CT on both idle time and line length can be computed. Figures 4.8 and 4.9 show a plot of normalized idle time and total line length as a function of normalized CT for closed and open stations respectively. From figures 4.8 and 4.9 the complete picture of how changing the launching rate affects the line length and total idle time can be seen and decision makers can decide on the trade-off between the increasing cost of longer lines and savings from less workers idle time.

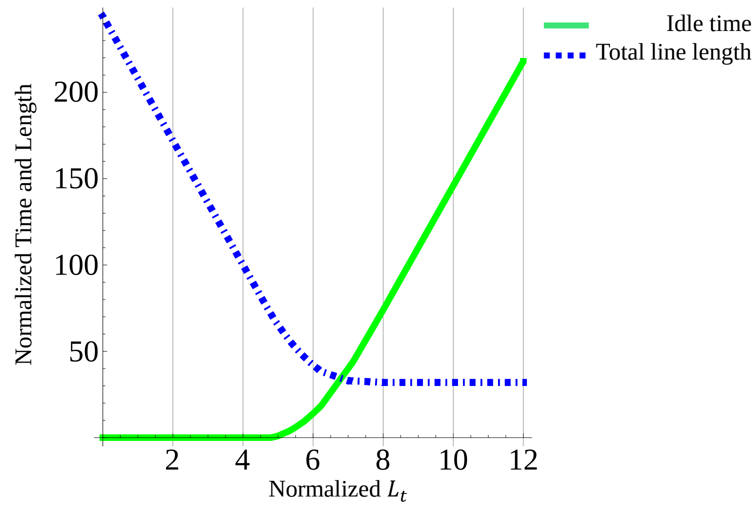


Figure 4.8 Total idle and line length as a function of l_t for closed stations MMAL.

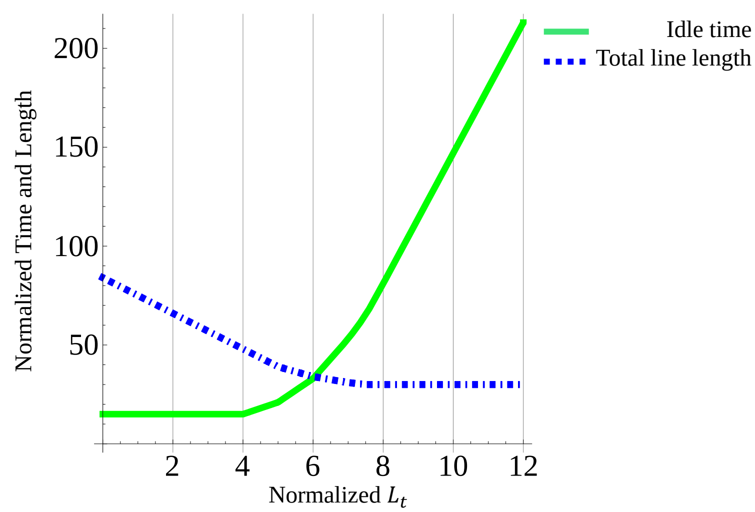


Figure 4.9 Total idle and line length as a function of l_t for open stations MMAL.

The other performance measure that is directly affected by changing l_t is the total throughput time which is defined as the total time required to produce the demanded MPS. The total throughput time can be calculated as the sum of two times, the first is the total time between launching the first job and launching the last job on the line, and the second is the total time spent by the last job in the line. For a line with closed stations total throughput time ($T_{th\text{-}C}$) can be calculated as:

$$T_{th-C} = (K-1) \times L_t + \sum \frac{(L_1,\ L_2, \ldots,\ L_{N-1}, X_{N,K})}{v} + t_{N,K} \qquad (4.15)$$

And for a line with open stations total throughput time ($T_{th\text{-}O}$) can be calculated as:

$$T_{th-O} = (K-1) \times l_t + \frac{X_{N,K}}{v} + t_{N,K} \qquad (4.16)$$

Using equations *(4.15)* and *(4.16)* and recalling that $X_{N,\mathrm{K}}$ can be expressed in terms of l_t for both open and closed stations, the exact relation between CT and the total throughput time can be evaluated. Figures 4.10 and 4.11 show the effect of changing l_t on the total throughput of a line with closed and open stations respectively. The figures show that for a line with closed stations there is an optimal l_t for which throughput time is minimal, while in the case of the line with open stations decreasing l_t leads to decreasing the throughput time up to a certain value after which the throughput time remains constant and does not change with further decrease in l_t.

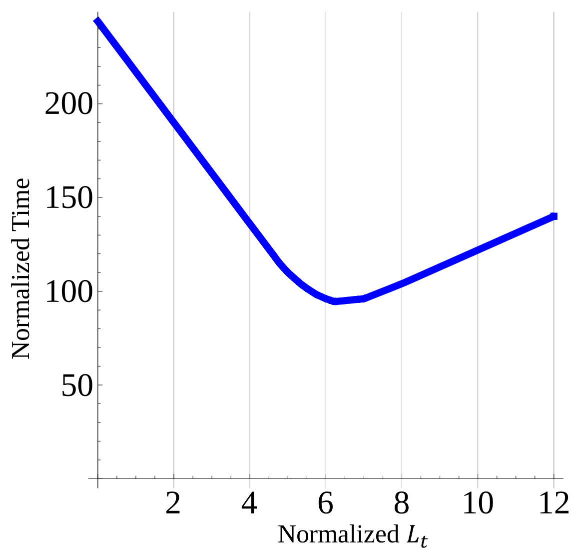


Figure 4.10 Total throughput time as a function of $\boldsymbol{l_t}$ for closed stations MMAL.

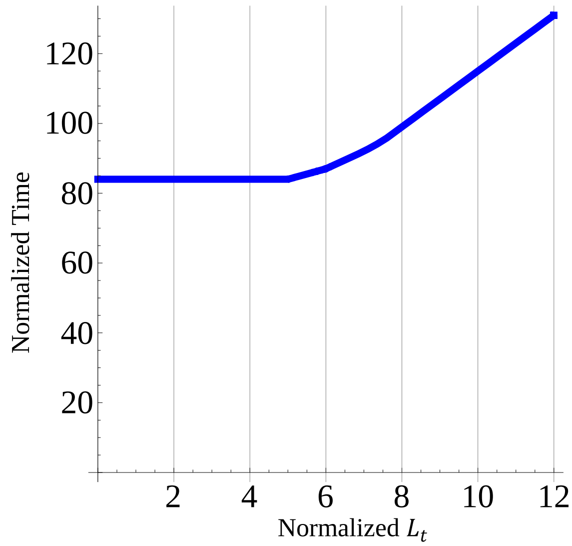


Figure 4.11 Total throughput time as a function of l_t for open stations MMAL.

4.4. Industrial Case Study: MMAL of Auto Car Seats

In auto car seats, variety can be found in the color and material of the seat cover, the availability of power seat adjustment, and the availability of heating and cooling options, etc.. Auto car seats are usually assembled in plants that are close to the final auto assembly plants and delivered just-in-time and in order to the final automobile assembly line. Orders come to the seat assembly plants in a certain order; however, delivery to the final assembly plant is done in batches. This gives the car assembly plants the ability to sequence the orders within any given batch. For example assuming a batch size of 30 car seats and a given order requires five seats with power adjustment option. Given that installing the power adjustment option requires significantly more time, it would make sense to sequence those five seats apart and then put them in order before shipping the seats to the final assembly plant.

In the case under study, a new line is being designed for the assembly of front seats for a Ford passenger vehicle. The seat can be configured to one of four seat configurations: 1) Manual Adjustment, 2) Power Adjustment, 3) Power Adjustment with Heating, and 4) Power Adjustment with Heating and Cooling. Configuration 1 is the basic seat configuration and comes with manual seat adjustment and no heating or cooling. Configuration 2 comes only with power seat adjustment and configuration 3 has both power adjustment and seat heating option. Configuration

4 is the top of the line with power seat adjustment, heating and cooling options. The assembly line consists of five stations as illustrated in figure 4.12.

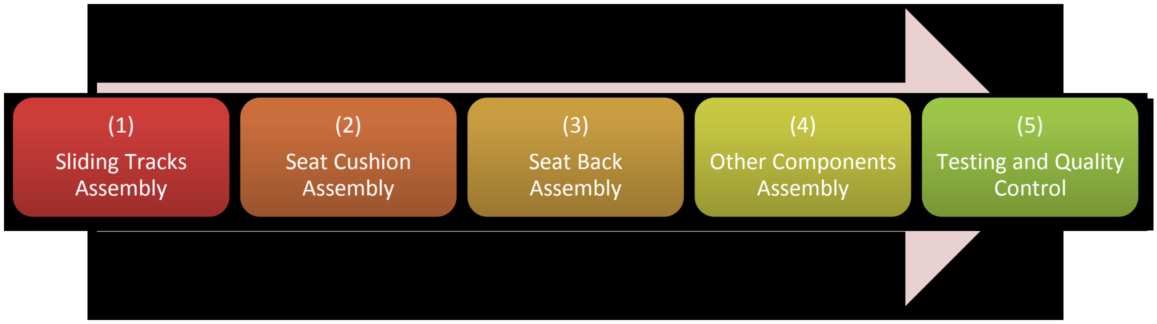


Figure 4.12 Assembly line stations for front seat.

The main components in the seat assembly are presented in figure 4.13. Note that not all components are present in all seat configurations. For example, the heating pads can be found only in configurations 3 and 4. In station (1), the sliding track of the seat is mounted on a moving pallet on the line. For configurations 2, 3 and 4, extra wirings and components are added. In station (2) the seat cushion and cover are added to the sliding track followed by adding the seat back assembly and cover in station (3). In station (4), other components such as the head rest and finish panels are added, and finally in station (5) the seat is tested for manual adjustment, power adjustment, and heating and cooling where applicable.

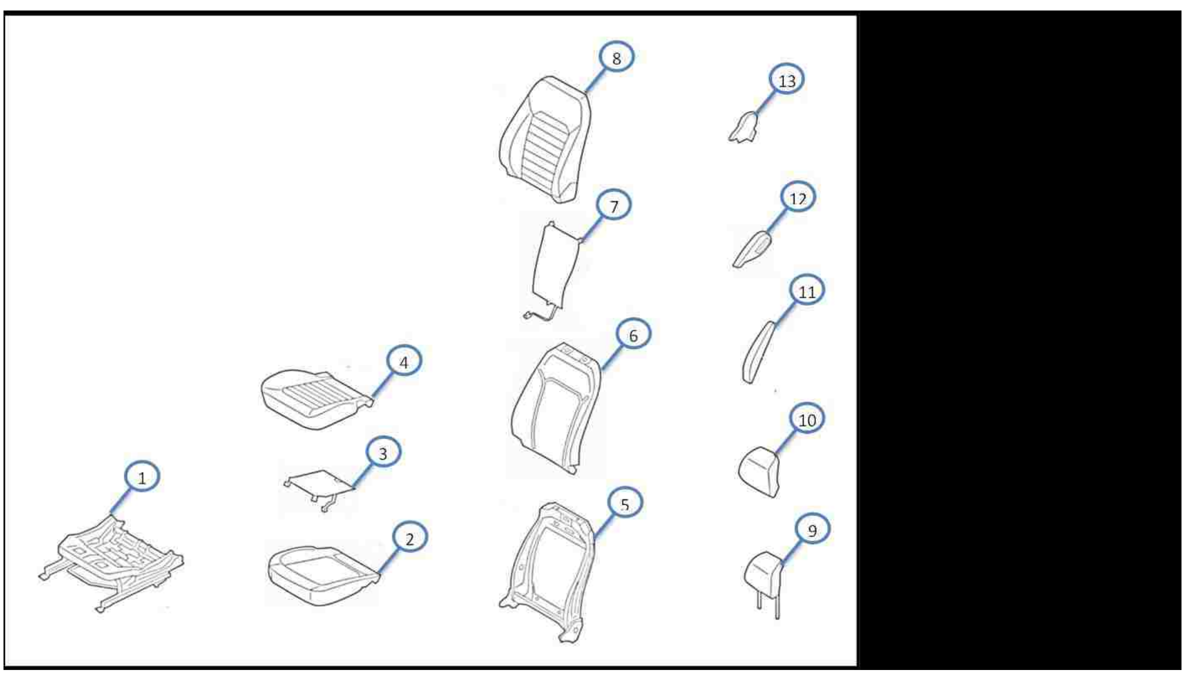


Figure 4.13 Main seat components.

The assembly time in seconds for each seat configuration at each station is provided in table 4.4. The time required to execute the task for a configuration *m* on a station *j* will be denoted as $t_{m,j}$.It should be noted that the required time for the manual configuration is the least in each station except for testing and quality control where the seat adjustment has to be tested manually and thus takes more time than the other configurations that are tested electronically. In addition to that, variation in task execution in all stations for all models is insignificant, except for the manual configuration in station (1) where a significant variation can exist depending on the skill of the worker.

Table 4-4 Required time in seconds for each seat configuration at each station.

	Sliding Tracks Assembly (1)	**Seat Cushion Assembly (2)**	**Seat Back Assembly (3)**	**Other Components Assembly (4)**	**Testing and Quality Control (5)**
1- Manual	30-40	38	38	41	47
2- Power	46	38	40	44	40
3- Power-Heating	50	42	43	47	40
4- Power-Heating-Cooling	53	44	45	47	40

For the new line to be designed, the seats are planned to be shipped to the final assembly plant in batches of 24 seats in each batch. The typical order for which the line is designed consists of 10 manual seats, 8 power seats, 4 power-heated seats and 2 power-heated-cooled seats i.e. the minimum part (MPS) is (10, 8, 4, 2). Given that workers require special tools with limited reach, the line is designed with closed stations.

Since the line is designed based on a given optimal sequence, obtaining that sequence should be the first step. However, obtaining an optimal sequence using an optimization model can only be done using specific task execution times for each model at each station, and therefore, deciding on the task execution time for configuration 1 on station (1) would be a problem. A good solution for this problem is to find the optimal sequence once using the lower limit of $t_{1,1}$, and then another time using the upper limit, and a third time using a middle value. Then using the model presented in section 4.2, the total line length for the sequences can be plotted as a function of $t_{1,1}$, and the sequence that has a lower length for most of the span of the possible values of $t_{1,1}$ should then be used.

In order to obtain the three optimal sequences, the model presented in (Bard, Dar-El et al. 1992) is used. The parameters for the model are *v=1* and launching time of products L_t= 42. Moreover, since *v=1*, then *w*=L_t*=42*. The three optimal sequences are presented in table 4.5 and the complete model used for obtaining these sequences is presented in Appendix D.

Table 4-5 Optimal sequences for different $\boldsymbol{t_{1,1}}$

	Optimal Sequence
$\boldsymbol{t_{1,1} = 30}$	1,2,2,1,4,1,3,2,1,3,2,1,2,2,2,1,1,4,1,3,2,1,3,1
$\boldsymbol{t_{1,1} = 35}$	3,1,2,2,1,2,2,2,1,3,1,2,1,4,1,1,3,2,1,2,1,3,1,4
$\boldsymbol{t_{1,1} = 40}$	2,1,1,3,1,2,2,2,3,3,1,4,2,1,1,1,1,2,2,2,4,1,1,3

The next step is to derive the max-plus equations for the line under study. Following equation (4.3), the line can be presented by:

$$\boldsymbol{X_k} = \boldsymbol{A}_k\, \boldsymbol{X}_{k-1} \oplus \boldsymbol{B} \qquad (4.17)$$

Where $\mathbf{X_k} = [X_1 \;\; X_2 \;\; X_3 \;\; X_4 \;\; X_5]_{\boldsymbol{k}}^{\boldsymbol{T}}$, where $X_{n,k}$ is the position, relative to starting edge of station *n*, where the worker starts working on job *k*, and

$$\boldsymbol{A}_k = \begin{bmatrix} \frac{t_{1,k-1}}{w} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \frac{t_{2,k-1}}{w} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \frac{t_{3,k-1}}{w} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \frac{t_{4,k-1}}{w} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \frac{t_{5,k-1}}{w} \end{bmatrix}, \text{ and } \boldsymbol{B} = \begin{bmatrix} e \\ e \\ e \\ e \\ e \end{bmatrix}.$$

Using equations (4.5) and (4.7), the total length and the total idle time of the line for each sequence can be found as a function of $t_{1,1}$ as:

$$L_T = \otimes_{n=1}^{N} L_n = \otimes_{n=1}^{N} \oplus_{k=1}^{K} (t_{n,k} X_{n,k})$$

$$I_T = \otimes_{k=1}^{K} I_k = \otimes_{k=1}^{K} \otimes_{n=1}^{N} (-\boldsymbol{A}_k \boldsymbol{X}_{k-1} \oplus \boldsymbol{B})$$

Figure 4.14 shows the plot of the total line length as a function of $t_{1,1}$ over values ranging from $t_{1,1} = 25$ to $t_{1,1} = 45$. And Figure 4.15 shows a plot of the total line idle time as also as a function of $t_{1,1}$ over the same range.

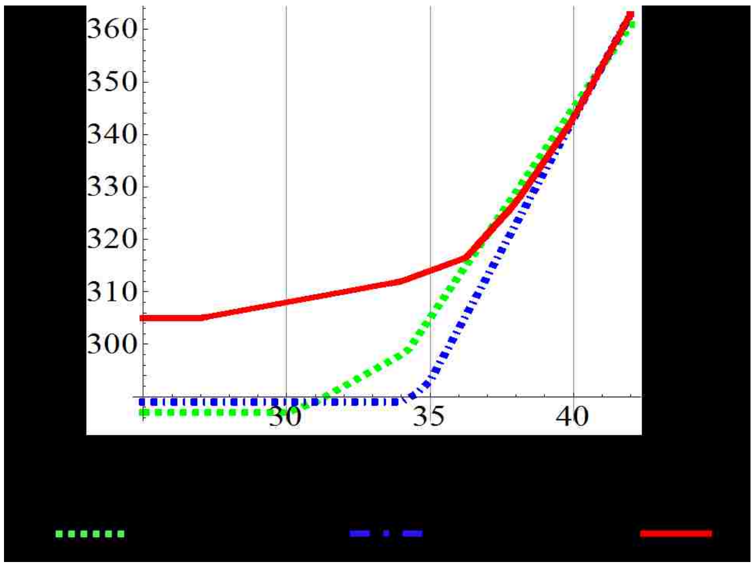


Figure 4.14 Total line length as a function of assembly time $\boldsymbol{t_{1,1}}$

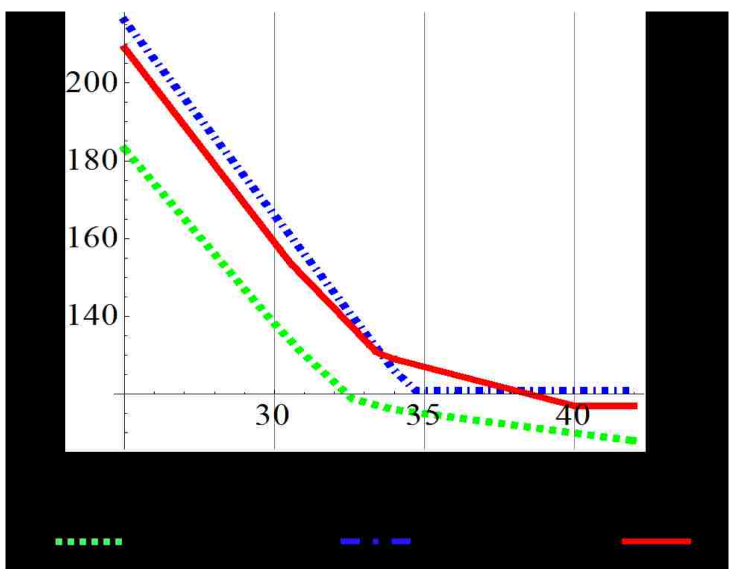


Figure 4.15 Total line idle time as a function of assembly time $\boldsymbol{t_{1,1}}$

From figure 4.14 it is obvious that sequence S2 is the best choice as it yields a shorter line length for a bigger portion of the $t_{1,1}$ range of values and is slightly worse than S1 and S3 when it is not the optimal sequence. The figure also informs management that when using sequence S2, workers at station 1 should not attempt to assemble variant 1 in less than 35 seconds as this will not improve the line length and will increase idle time. When examining figure 4.15 in addition to 4.14, the management decision could lean towards choosing S1as the difference in line length between S1 and S2 is not big and S1 always has significantly less total idle time.

It should be noted that using equations (4.15) and (4.17), the total line length and total idle time of the assembly line can be obtained as a function of more than one assembly time. As an example, if there are significant variations in both $t_{1,1}$ and $t_{1,2}$, total line length can be obtained as a function of both variables and plotted as a 3 D plot as in figure 4.16.

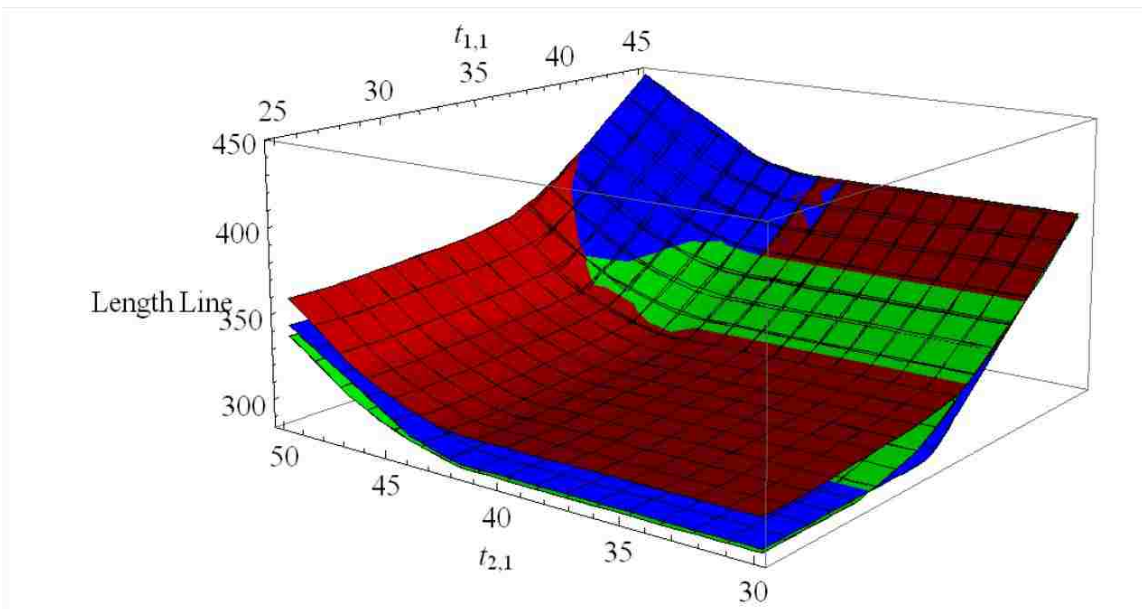


Figure 4.16 Total line length as a function of $t_{1,1}$ and $t_{1,2.}$

4.5. Discussion and Conclusions

Mixed Model Assembly lines with both closed and open stations have been modeled for the first time using max-plus algebra. The developed models were used to compare given sequences of demand mix over a range of processing times of assembly tasks as well as analyze different line performance measures while considering one of the line parameters as a variable. Hence, the effect of changes in any of the system parameters on the optimality of a given sequence and on the line performance can be assessed.

In this dissertation, only closed and open stations with fixed launching rate have been considered; however, using the same procedure variable rate launching, lines with mixed stations – some open and some closed – and lines with stations that are only left open or only right open can be modeled and analyzed.

A major advantage of using this mathematical modeling approach is that the developed model provides many insights to decision makers on the effect of line parameters on its performance in a very short time. In order to arrive at the same insights using discrete events simulation, a large number of complete simulation runs would be required as every simulation run would provide only one data point on any of the graphs presented above.

The analyses that are made possible using the developed model can be useful to decision makers during the early design phase of a new line as well as when considering line improvements since they provide a complete picture of the effect of changing any system variable on its total performance. In early design stages there is usually an expected demand mix for which the line is designed as well as the expected assembly time for each variant in each station. The presented analyses would allow designers to see which stations are most sensitive to changes in assembly time and whether it will affect the line length or the idle time. Designers can also use the presented analyses in redesigning existing lines, in which case the line length is a fixed constraint but the line capacity can be adjusted by changing the launching rate and length of each station. Also when considering line improvements, the presented analyses can be useful to decision makers in assessing if the improvements would affect the optimality of the current sequence and whether the line capacity can be increased by changing the lunching rate. An industrial case study was presented to show instances where the developed models can be useful in providing insight into the effect of changing system parameters on the performance.

CHAPTER 5: MAX-PLUS MODELING OF RE-ENTRANT MANUFACTURING SYSTEMS

5.1. Introduction

Re-entrant flow lines are a special class of manufacturing flow lines where parts flowing through the system are processed on some machines more than once (Kumar 1993; Diaz-Rivera, Armbruster et al. 2000). This type of flow lines is used widely in the semiconductor wafer fabrication where the final product consists of several layers each of which requires similar production operations and duplication of resources would not be warranted. Thus, instead of wasting capital on several identical machines, the products flow through the manufacturing line, or parts of it, several times (Kumar and Kumar 2001). Re-entrant flow lines can also be found in the automotive industry, as in fuel injector production lines (Wang and Li 2010), in manufacturing systems with automatic storage retrieval systems (ASRSs) (Suk and Cassandras 1989), in textile industry, and in mirror manufacturing plants (Choi and Kim 2006). Examples of simple re-entrant systems are shown in figure 5.1.

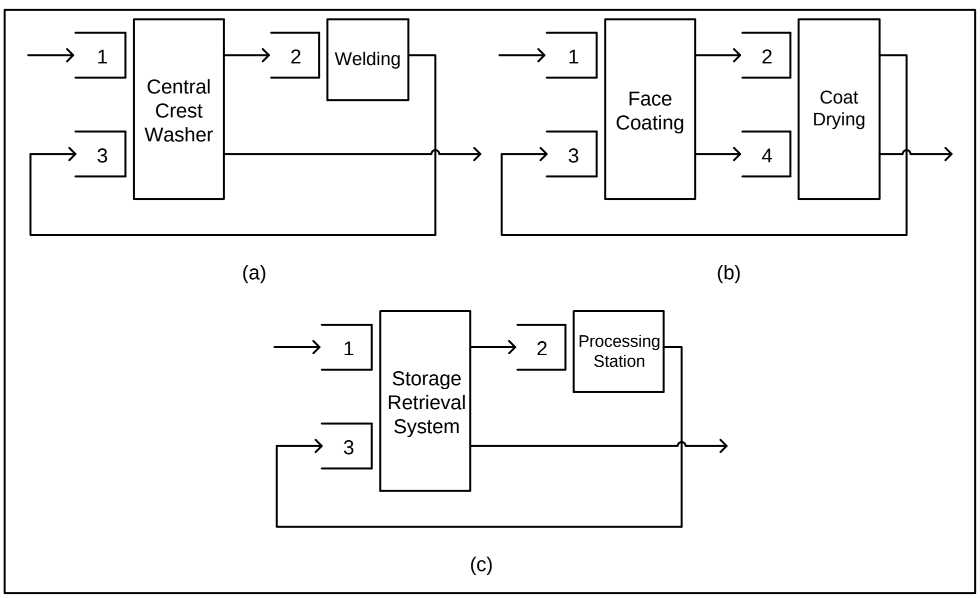


Figure 5.1 Examples of simple re-entrant manufacturing systems. (a) Part of layout of fuel injector assembly system, adapted from (Wang and Li 2010). (b) Plating process in mirror manufacturing (Choi and Kim 2006). (c) Processing station with ASRS

The feedback flows in re-entrant systems lead to complex behavior that is difficult to predict and control even in the simplest re-entrant manufacturing systems (Diaz-Rivera, Armbruster et al. 2000). In many cases, such behavior has been reported as chaotic, in the sense of dynamic chaos theory (Beaumariage and Kempf 1994; Ott 2002). In a recent review on throughput analysis of production systems, Li et al. (2009) asserted that analysis of re-entrant lines is almost non-existent and that analytical tools for accurate performance analysis are required especially for large-volume manufacturing industries.

In this chapter, max-plus algebra is used to generate state-space equations that model a simple re-entrant system similar to that in figure 5.1(a) with two machines, where two operations are performed on the first machine and one operation on the second. The derived equations are used to analyze the system and determine the effect of changing the processing times on the steady state inter-arrival time of finished jobs. Section 5.2 reviews the literature on the complex behavior of re-entrant manufacturing systems as well as the use of max-plus algebra in modeling manufacturing systems. Section 5.3 presents a brief overview of the max-plus algebra. The studied system is presented in section 5.4 then the model is developed in section 5.5, the model is analysis and results are presented in section 5.6, and finally discussion and conclusions are included in section 5.7.

5.2. Literature Review

5.2.1. Re-entrant Systems

Complex behavior in re-entrant manufacturing systems has been reported in many publications such as (Beaumariage and Kempf 1994; Dini, Failli et al. 1999; Wiendahl and Scheffczyk 1999; Diaz-Rivera, Armbruster et al. 2000; Schmitz, Van Beek et al. 2002; Chryssolouris, Giannelos et al. 2004; Alfaro and Sepulveda 2006; ElMaraghy and Manns 2009; Manns and ElMaraghy 2009; Dong and He 2012). Describing this complex behavior as chaotic in the sense of dynamic chaos theory is debatable and it was proven in some cases to be in fact periodic or eventually periodic (Diaz-Rivera, Armbruster et al. 2000; Schmitz, Van Beek et al. 2002). Accordingly the complexity of re-entrant systems might only be imaginary complexity (ElMaraghy, ElMaraghy et al. 2012) resulting from the lack of understanding of the system behavior.

Due to the general complexity of re-entrant systems, most of the research in this field depended mainly on simulation (Lu and Kumar 1991; Beaumariage and Kempf 1994; Bispo and Tayur 2001; Schmitz, Van Beek et al. 2002; Alfaro and Sepulveda 2006; He, Dong et al. 2011), where the system's structure and parameters were changed systematically, simulated and the resulting trends recorded in order to arrive at conclusions regarding the effect of these parameters on the system behavior. However, because exhaustive simulation of all possible combinations of system parameter values is impossible, definite conclusions regarding causes of complex behavior and the effect of certain parameters on the system behavior are not possible.

Narahari and Khan (Narahari and Khan 1996) presented an approximate technique for analytical performance prediction of re-entrant systems. Discrete event simulation was used to validate their model, however, the performance indicators considered in their model were only the mean steady state cycle time and the mean steady state throughput rate for a given fixed WIP in the system and thus no information could be obtained about the detailed behavior of the system and its periodicity. Another approximate technique was presented by Wang and Li in (Wang and Li 2010), where a re-entrant line consisting of M machines with one re-entrance is converted to 2 M machines serial line which is then analyzed using queuing theory. They also used discrete event simulation to verify their model by comparing simulated and analytically derived production rates.

A system similar to that in figure 5.1(a) was studied in (Diaz-Rivera, Armbruster et al. 2000) using dynamical systems theory to determine whether its behavior is indeed chaotic. The system was modelled as a continuous fluid model of a queuing network and observed at fixed events to arrive at a piecewise linear map. All possible system states as well as allowable transitions between them were then derived and periodic sequences of state transitions were determined. However, the periodic sequences were obtained by inspection or using simulation, which made the analysis valid only for very simple cases, where periodic orbits can be observed by inspection or only for the simulated case when simulation was used.

ElMaraghy and Manns (2009) presented a synchronization methodology that limits the number of different inter-arrival times of a re-entrant manufacturing system and controls the length of inter-arrival time periods. By doing so the predictability of the system's states increases and the unanticipated states that can lead to system failures are eliminated. In a later publication Manns and ElMaraghy (2009) presented an analytical approach to model the inter-arrival time behavior of a re-entrant system. They employed a queuing-situational decomposition which helps the manufacturing system designer avoid the resulting decrease in capacity and reliability due to the undesirable dynamic behavior which increased system complexity and unpredictability.

5.2.2. Max-Plus and Re-entrant Systems

One of the main underlying assumptions in max-plus algebra is that the system to be modeled must be representable by a Timed Event Graph (TEG) (Cohen, Dubois et al. 1985). The main characteristic of TEGs is that they are decision free, which, in the jargon of petri nets is translated to having only one upstream and one downstream transition for each place. This characteristic limited the use of max-plus in modeling manufacturing systems that are not decision free such as flow lines with shared resources (e.g. a robot that serves two stations), job shops, and re-entrant lines.

There have been several attempts to overcome this modeling issue and extend max-plus to systems with decision. Linear time-varying max-plus equations have been proposed in (Lahaye, Boimond et al. 2004; Addad, Amari et al. 2010) , however they are limited to cyclic systems where the cycle is defined a priori. A switching max-plus linear system was proposed by (van den Boom and De Schutter 2006), where the system contains different operating modes represented by different equations and switching between them occurs according to a given rule, thus the behavior of the system as a whole cannot be analyzed. Correia et al. (2009) proposed a model for systems with resource sharing using linear equations. A matrix model (Bogdan, Kovacic et al. 2004) is combined with a max-plus algebra model to combine control and system analysis for re-entrant systems, however, the sequence of allocation of resources has to be known a priori. These attempts succeeded in arriving at a max-plus state-space representation for the system to be used in control, but all of them resolved the decision points apriori and the representation is thus given only for a certain sequence and different sequences will have different representations.

In summary, re-entrant flow lines can exhibit complex behavior even in very simple manufacturing systems configurations. While simulation can be used to analyze a given instance of the system, it cannot be used to understand the effect of changing the systems' parameters on the overall behavior over time unless results of countless simulation runs are integrated and analyzed. Analysis of re-entrant manufacturing systems found in literature was based mostly on simulation (Lu and Kumar 1991; Beaumariage and Kempf 1994; Bispo and Tayur 2001; Schmitz, Van Beek et al. 2002; Alfaro and Sepulveda 2006; He, Dong et al. 2011) and thus were not capable of providing information about the behavior of the system in the case of any small change in the system parameters. Some approximation models were used to convert re-entrant systems into serial ones (Narahari and Khan 1996; Wang and Li 2010); however, the available analysis outcomes were not exact. Using max-plus algebra state-space linear equations to describe re-entrant systems is possible but is only limited to systems with known schedule and every change in the schedule would require re-modeling. The advantage of using max-plus algebra to model re-entrant systems is that the resulting equations representing the system can be easily and quickly used to gain insight into the systems' behavior and the effects of different systems parameters on the overall system dynamic behavior.

5.3. Re-Entrant Manufacturing System Description

The system under investigation, figure 5.2, consists of two stations performing three processes, *A*, *B*, and *C*. Processes *A* and *C* are performed on machine 1, with a dedicated queue for each process, and process *B* is performed on machine 2. This system is similar to that in figure 5.1 representing the automotive fuel injector assembly system (Wang and Li 2010), the semiconductor manufacturing cell used in (Diaz-Rivera, Armbruster et al. 2000), and automated storage-retrieval systems (Suk and Cassandras 1989).

The following assumptions are employed:

- Processing times are deterministic and constant for each process.
- Machine break-downs are not modelled.
- The system is palletized with a constant number of pallets circulating within the line.
- Transfer time between machines is negligible.

The first assumption is realistic for automated systems as well as semi-automated systems with palletized material handling where the process time variation is much less than the processing time and thus can be neglected. The second assumption is also realistic when studying the normal short-term operation with the objective of understanding and optimizing the system behavior as opposed to studying long-term operation with the objective of planning capacity where machines breakdown would have an effect. Transfer time between the machines is assumed negligible following the system in (Diaz-Rivera, Armbruster et al. 2000), and can be easily taken into consideration in future work.

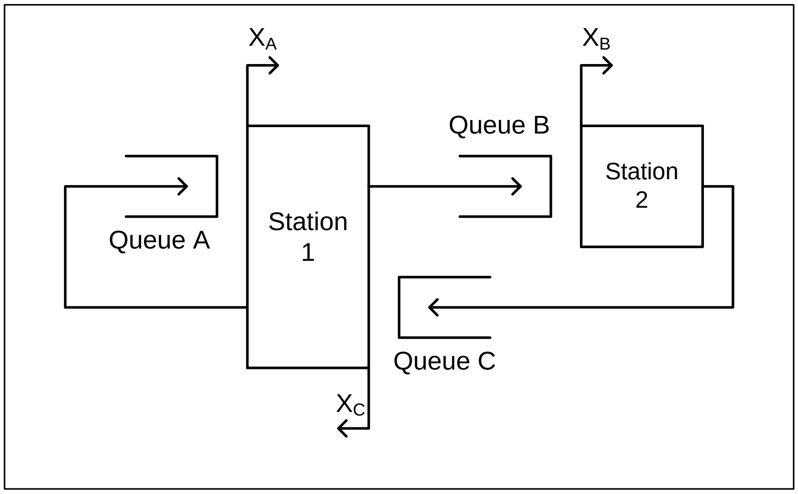


Figure 5.2 Re-entrant Manufacturing System (2 machines / 3 Processes)

The decision point in this system is located at station 1 when there are pallets waiting in queues *A* and *C* for processing on station 1. Different scheduling rules can be used to regulate the pallets flow including FCFS (First Come First Serve), FBFS (First Buffer First Serve), and LBFS (Last Buffer First Serve). In this, the LBFS policy will be assumed which gives priority to jobs in queue C. This policy results in the least cycle-time for the system according to (Kumar 1993).

5.4. System Modeling

Let $XA_{i,k}$, $XB_{i,k}$, and $XC_{i,k}$ be the starting time of process *A*, *B* and *C* respectively for pallet i for the k^{th} time, t_A, t_B, and t_C be the processing times for processes *A*, *B* and *C* respectively, and let $\boldsymbol{XA}_k = [XA_{1,k}\ XA_{2,k}\ \ldots XA_{n,k}]^T$, $\boldsymbol{XB}_k = [XB_{1,k}\ XB_{2,k}\ \ldots XB_{n,k}]^T$, and $\boldsymbol{XC}_k = [XC_{1,k}\ XC_{2,k}\ \ldots XC_{n,k}]^T$ where n is the total number of pallets in the line. Accordingly, $XA_{2,3}$ is the starting time of process *A* on pallet number 2 for the third time and $\boldsymbol{XB_2}$ is the vector representing the starting time of process *B* for all pallets for the second time.

Let $UA_{i,k}$, $UB_{i,k}$, and $UC_{i,k}$ be the arrival time of pallet i for the k^{th} time to queue *A*, *B* and *C* respectively. Since transport time is negligible then $\boldsymbol{UB_k} = t_A \otimes \boldsymbol{XA_k}$, $\boldsymbol{UC_k} = t_B \otimes \boldsymbol{XB_k}$, and $\boldsymbol{UA_k} = t_C \otimes \boldsymbol{XC_{k-1}}$. For simplicity and without loss of generality, the system will be analyzed for only 2 pallets circulating the line. The same procedure can be used for larger numbers of pallets.

If n is equal to 2, then we have:

$$\boldsymbol{XA_k} = \begin{bmatrix} XA_1 \\ XA_2 \end{bmatrix}_{\boldsymbol{k}} = \begin{bmatrix} \varepsilon & \varepsilon \\ t_A & \varepsilon \end{bmatrix} \begin{bmatrix} XA_1 \\ XA_2 \end{bmatrix}_{\boldsymbol{k}} \oplus \begin{bmatrix} t_C & \eta 1 \\ \varepsilon & t_C \end{bmatrix} \begin{bmatrix} XC_1 \\ XC_2 \end{bmatrix}_{\boldsymbol{k-1}} \tag{5.1}$$

$$\boldsymbol{XB_k} = \begin{bmatrix} XB_1 \\ XB_2 \end{bmatrix}_{\boldsymbol{k}} = \begin{bmatrix} \varepsilon & \varepsilon \\ t_B & \varepsilon \end{bmatrix} \begin{bmatrix} XB_1 \\ XB_2 \end{bmatrix}_{\boldsymbol{k}} \oplus \begin{bmatrix} \varepsilon & t_B \\ \varepsilon & \varepsilon \end{bmatrix} \begin{bmatrix} XB_1 \\ XB_2 \end{bmatrix}_{\boldsymbol{k-1}} \oplus \begin{bmatrix} t_A & \varepsilon \\ \varepsilon & t_A \end{bmatrix} \begin{bmatrix} XA_1 \\ XA_2 \end{bmatrix}_{\boldsymbol{k}} \tag{5.2}$$

$$\begin{aligned} \boldsymbol{XC_k} = \begin{bmatrix} XC_1 \\ XC_2 \end{bmatrix}_{\boldsymbol{k}} &= \begin{bmatrix} \varepsilon & \varepsilon \\ t_C & \varepsilon \end{bmatrix} \begin{bmatrix} XC_1 \\ XC_2 \end{bmatrix}_{\boldsymbol{k}} \oplus \begin{bmatrix} \varepsilon & t_A \\ \varepsilon & \varepsilon \end{bmatrix} \begin{bmatrix} XA_1 \\ XA_2 \end{bmatrix}_{\boldsymbol{k}} \oplus \begin{bmatrix} t_B & \varepsilon \\ \varepsilon & t_B \end{bmatrix} \begin{bmatrix} XB_1 \\ XB_2 \end{bmatrix}_{\boldsymbol{k}} \\ &\oplus \begin{bmatrix} \varepsilon & \varepsilon \\ \eta 2 & \varepsilon \end{bmatrix} \begin{bmatrix} XA_1 \\ XA_2 \end{bmatrix}_{\boldsymbol{k+1}} \end{aligned} \tag{5.3}$$

where $\eta 1$and $\eta 2$ are parameters used to enforce the scheduling rule as will be demonstrated later.

Let:

$$A_1 = \begin{bmatrix} \varepsilon & \varepsilon \\ t_A & \varepsilon \end{bmatrix}, \quad A_2 = \begin{bmatrix} t_C & \eta 1 \\ \varepsilon & t_C \end{bmatrix}, \quad B_1 = \begin{bmatrix} \varepsilon & \varepsilon \\ t_B & \varepsilon \end{bmatrix}, \quad B_2 = \begin{bmatrix} \varepsilon & t_B \\ \varepsilon & \varepsilon \end{bmatrix}, \quad B_3 = \begin{bmatrix} t_A & \varepsilon \\ \varepsilon & t_A \end{bmatrix}$$

$$C_1 = \begin{bmatrix} \varepsilon & \varepsilon \\ t_C & \varepsilon \end{bmatrix}, \quad C_2 = \begin{bmatrix} \varepsilon & t_A \\ \varepsilon & \varepsilon \end{bmatrix}, \quad C_3 = \begin{bmatrix} t_B & \varepsilon \\ \varepsilon & t_B \end{bmatrix}, \quad C_4 = \begin{bmatrix} \varepsilon & \varepsilon \\ \eta 2 & \varepsilon \end{bmatrix}$$

Then equation (*5.1)* can be written as:

$$\boldsymbol{XA_k} = A_1\, \boldsymbol{XA_k} \oplus A_2\, \boldsymbol{XC_{k-1}} \tag{5.4}$$

which, according to theorem (2.1), is equal to:

$$\boldsymbol{XA_k} = {A_1}^*\, A_2\, \boldsymbol{XC_{k-1}} \tag{5.5}$$

where:

$${A_1}^* = \begin{bmatrix} e & \varepsilon \\ t_A & e \end{bmatrix} \tag{5.6}$$

Similarly:

$$\boldsymbol{XB_k} = B_1\, \boldsymbol{XB_k} \oplus B_2\, \boldsymbol{XB_{k-1}} \oplus B_3\, \boldsymbol{XA_k} \tag{5.7}$$

$$\boldsymbol{XC_k} = C_1\, \boldsymbol{XC_k} \oplus C_2\, \boldsymbol{XB_k} \oplus C_3\, \boldsymbol{XA_k} \oplus C_4\, \boldsymbol{XA_{k+1}} \tag{5.8}$$

Substituting *(5.5)* in *(5.8)* yields:

$$\boldsymbol{XC_k} = (C_1 \oplus C_4\, A_1{}^{*}\, A_2)\, \boldsymbol{XC_k} \oplus C_2\, \boldsymbol{XB_k} \oplus C_3\, \boldsymbol{XA_k} \tag{5.9}$$

and by letting:

$$C_5 = C_4\, A_1{}^{*}\, A_2 = \begin{bmatrix} \varepsilon & \varepsilon \\ t_C\, \eta 2 & \eta 1\, \eta 2 \end{bmatrix} \tag{5.10}$$

equations (*5.4)*, (*5.7)*, and (*5.9)* can by combined in one equation in the form of:

$$\boldsymbol{X_k} = \begin{bmatrix} \boldsymbol{XA} \\ \boldsymbol{XB} \\ \boldsymbol{XC} \end{bmatrix}_{\boldsymbol{k}} = \begin{bmatrix} A_1 & E & E \\ B_3 & B_1 & E \\ C_3 & C_2 & C_1 \oplus C_5 \end{bmatrix} \begin{bmatrix} \boldsymbol{XA} \\ \boldsymbol{XB} \\ \boldsymbol{XC} \end{bmatrix}_{\boldsymbol{k}} \oplus \begin{bmatrix} E & E & A_2 \\ E & B_2 & E \\ E & E & E \end{bmatrix} \begin{bmatrix} \boldsymbol{XA} \\ \boldsymbol{XB} \\ \boldsymbol{XC} \end{bmatrix}_{\boldsymbol{k-1}} \tag{5.11}$$

where E is a 2×2 matrix with each element equal to ε. Thus, $\boldsymbol{X_k}$ holds the starting time of all processes for all pallets for the k^{th} time.

By letting:

$$G = \begin{bmatrix} A_1 & E & E \\ B_3 & B_1 & E \\ C_3 & C_2 & C_1 \oplus C_5 \end{bmatrix}, and\ \ H = \begin{bmatrix} E & E & A_2 \\ E & B_2 & E \\ E & E & E \end{bmatrix},$$

equation *(5.11)* becomes:

$$\boldsymbol{X_k} = G\, \boldsymbol{X_k} \oplus H\, \boldsymbol{X_{k-1}} = G^{*}H\, \boldsymbol{X_{k-1}} \tag{5.12}$$

Equation *(5.12)* represents the system's behavior in the k^{th} cycle as a function of the preceding cycle and matrix $G^{*}H$ which can be calculated by knowing only the processing times. The Eigen-value of the matrix $G^{*}H$ represents the growth rate of each of the parameters in $\boldsymbol{X}$ at steady state and thus the inter-arrival rate can be deduced from it.

5.5. Max-Plus System Model Analysis

Analysis should start by examining the parameters $\eta 1$ and $\eta 2$. These parameters enforce the scheduling rule by taking one of two values: t_c or ε for $\eta 1$, and t_A or ε for $\eta 2$. If pallet number 2 arrives at Queue C before pallet number 1 arrives at Queue A ($UC_{2,k-1} < UA_{1,k}$), then pallet number 1 cannot start process A for the k^{th} time except after pallet number 2 finishes process C. Thus, the value of $\eta 1$ can be summarized by:

$$\eta 1 = \{ \begin{matrix} t_c, & if\ UC_{2,k-1} < UA_{1,k} \\ \varepsilon, & if\ UC_{2,k-1} > UA_{1,k} \end{matrix} \tag{5.13}$$

Similarly, the value of $\eta 2$ can be summarized by:

$$\eta 2 = \{ \begin{matrix} \varepsilon, & if\ UC_{2,k-1} < UA_{1,k} \\ t_A, & if\ UC_{2,k-1} > UA_{1,k} \end{matrix} \tag{5.14}$$

and equations *(5.13)* and *(5.14)* can be combined as:

$$\begin{matrix} if\ UC_{2,k-1} < UA_{1,k} \rightarrow \eta 1 = t_c\ \ and\ \eta 2 = \varepsilon \\ if\ UC_{2,k-1} > UA_{1,k} \rightarrow \eta 1 = \varepsilon\ and\ \eta 2 = t_A \end{matrix} \tag{5.15}$$

Recalling that $UC_{2,k-1} = t_B\ \ XB_{2,k-1}$ and $UA_{1,k} = t_C\ \ XC_{1,k-1}$, and then substituting the value of XB_2 and XC_1 from *(5.2)* and *(5.3)*, we arrive at:

$$\begin{matrix} UC_{2,k-1} = t_B\, XB_{2,k-1} = t_B\, t_A\ \ XA_{2,k-1} \oplus t_B{}^2\, XB_{1,k-1} \\ UA_{1,k}\ = t_C\, XC_{1,k-1} = t_C\, t_A\ \ XA_{2,k-1} \oplus t_C\, t_B\, XB_{1,k-1} \end{matrix} \tag{5.16}$$

From equations *(5.15)* and *(5.16)*, it is obvious that the value of $\eta 1$and $\eta 2$ depends solely on the values of t_B and t_C according to:

$$\begin{matrix} if\ t_B < t_C \rightarrow \eta 1 = TC\ and\ \eta 2 = \varepsilon \\ if\ t_B > t_C \rightarrow \eta 1 = \varepsilon\ and\ \eta 2 = t_A \end{matrix} \tag{5.17}$$

With the values of $\eta 1$ and $\eta 2$ determined, analysis of the system can be performed based on the values of the processing times. In the rest of this section, the system will be analyzed for all possible scenarios.

It should be noted that equation *(5.12)* along with equation *(5.17)* fully describes the dynamics of the system for any processing times and the following analysis can be done only for the scenarios of importance. Thus for larger systems with more scenarios to be considered, the following analysis is not required for all possible scenarios.

5.5.1. Processing Times $\boldsymbol{t_C < t_B < t_A}$

According to equation *(5.17)*, in this case $\eta 1 = \varepsilon$ and $\eta 2 = t_A$. Using the values of $\eta 1$ and $\eta 2$ and simplifying for $t_C < t_B < t_A$ we get:

$$G^*H \quad = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_A\, t_C & t_C \\ \varepsilon & \varepsilon & \varepsilon & t_B & t_A\, t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & {t_A}^2 t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & {t_A}^2 t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & t_A\, {t_B}^2\, t_C & {t_A}^3 t_C & {t_A}^2 {t_C}^2 \end{bmatrix} \qquad (5.18)$$

In this case, the Eigen-value of Matrix G^*H is ${t_A}^2{t_C}^2$. The Eigen-value represents the difference between the k^{th} and the $k-1^{th}$ instances of any of the parameters of $\boldsymbol{X}$ at steady state. Thus, the Eigen-value gives the time between $XC_{1,k}$ and $XC_{1,k-1}$and the time between $XC_{2,k}$ and $XC_{2,k-1}$, while for inter-arrival rate, the time between $XC_{1,k}$ and $XC_{2,k-1}$ is required.

From equations *(5.12)* and *(5.18)* it can be shown that:

$$XC_{1,k} = {t_B}^2\, XB_{2,k-1} \oplus {t_A}^2\, t_C\, XC_{1,k-1} \oplus t_A\, t_C\, XC_{2,k-1} \qquad (5.19)$$

and given that $XC_{1,k} - XC_{1,k-1} = {t_A}^2{t_C}^2$, equation *(5.19)* becomes:

$$XC_{1,k} = {t_B}^2\, XB_{2,k-1} \oplus t_A\, t_C\, XC_{2,k-1} \qquad (5.20)$$

Since $XC_{2,k-1} \geq t_B\, XB_{2,k-1}$ and $t_A > t_B$, then:

$$XC_{1,k} = t_A\, t_C\, XC_{2,k-1} \qquad (5.21)$$

and:

$$XC_{2,k} = t_A\, t_C\, XC_{1,k} \qquad (5.22)$$

Thus, the inter-arrival time for the case of processing times $t_C < t_B < t_A$ is $t_A\, t_C$, which in conventional algebra is equal to $t_A + t_C$. Thus in this case, one product will be finished every $t_A + t_C$ time units.

5.5.2. Processing Times $\boldsymbol{t_B < t_C < t_A}$

In this case $\eta 1 = \varepsilon$ and $\eta 2 = t_A$. Using the values of $\eta 1$ and $\eta 2$ and simplifying for $t_C < t_B < t_A$ gives:

$$G^*H = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_A\, t_C & t_C \\ \varepsilon & \varepsilon & \varepsilon & t_B & t_A\, t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & {t_A}^2 t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & {t_A}^2 t_C & {t_A}^2 t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2\, t_C & {t_A}^2 {t_C}^2 & {t_A}^2 {t_C}^2 \end{bmatrix} \qquad (5.23)$$

Again in this case, the Eigen-value of Matrix G^*H is ${t_A}^2{t_C}^2$, however, by examining $XC_{1,k}$ it is found that:

$$XC_{1,k} = {t_B}^2\, XB_{2,k-1} \oplus {t_A}^2\, t_C\, XC_{1,k-1} \oplus {t_A}^2\, t_C XC_{2,k-1} \qquad (5.24)$$

and similar to above, it can be shown that:

$$XC_{1,k} = {t_A}^2\, t_C XC_{2,k-1} \qquad (5.25)$$

and:

$$XC_{2,k} = t_C XC_{1,k} \qquad (5.26)$$

From equations *(5.25)* and *(5.26)*, it can be seen that the inter-arrival rate in this case fluctuates between $2t_A + t_C$ and t_C.

5.5.3. Processing Times $\boldsymbol{t_B < t_A < t_C}$

This case is exactly the same as case II with:

$$G^*H = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_A\, t_C & t_C \\ \varepsilon & \varepsilon & \varepsilon & t_B & t_A\, t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & {t_A}^2 t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & {t_A}^2 t_C & {t_A}^2 t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2\, t_C & {t_A}^2 {t_C}^2 & {t_A}^2 {t_C}^2 \end{bmatrix} \qquad (5.27)$$

The Eigen-value of Matrix G^*H is ${t_A}^2{t_C}^2$, and again the inter-arrival rate in this case fluctuates between $2t_A + t_C$ and t_C according to:

$$XC_{1,k} = {t_A}^2\, t_C XC_{2,k-1} \qquad (5.28)$$

and:

$$XC_{2,k} = t_C XC_{1,k} \tag{5.29}$$

5.5.4. Processing Times $\boldsymbol{t_A < t_B < t_C}$

In this case $\eta 1 = \varepsilon$ and $\eta 2 = t_A$. Using the values of $\eta 1$ and $\eta 2$ and simplifying for $t_C < t_B < t_A$ gives:

$$G^*H \quad = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_C & t_C \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_A\, t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & t_B & t_A\, t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & t_A\, t_B t_C & t_A\, t_B t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & t_A\, t_B t_C & t_A\, t_B t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2\, t_C & t_A\, t_B {t_C}^2 & t_A\, t_B {t_C}^2 \end{bmatrix} \tag{5.30}$$

The Eigen-value of Matrix G^*H is $t_A\, t_B {t_C}^2$, and the equations for process C are:

$$XC_{1,k} = t_A\, t_B t_C XC_{2,k-1} \tag{5.31}$$

and:

$$XC_{2,k} = t_C XC_{1,k} \tag{5.32}$$

Thus the inter-arrival time fluctuates between $t_A + t_B + t_C$ and t_C.

5.5.5. Processing Times $\boldsymbol{t_C < t_A < t_B}$

In this case $\eta 1 = \varepsilon$ and $\eta 2 = t_A$. Using the values of $\eta 1$ and $\eta 2$ and simplifying for $t_C < t_A < t_B$ gives:

$$G^*H = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_A\, t_C & t_C \\ \varepsilon & \varepsilon & \varepsilon & t_B & t_A\, t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & t_A\, t_B t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & t_A\, t_B t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & \begin{cases} {t_B}^3, t_A\, t_C \le t_B \\ t_A\, {t_B}^2\, t_C, t_A\, t_C > t_B \end{cases} & \begin{cases} t_A\, {t_B}^2\, t_C, t_A\, t_C \le t_B \\ {t_A}^2 t_B\, {t_C}^2\, t_C, t_A\, t_C > t_B \end{cases} & \begin{cases} t_A\, t_B t_C, t_A\, t_C \le t_B \\ {t_A}^2\, {t_C}^2, t_A\, t_C > t_B \end{cases} \end{bmatrix} \tag{5.33}$$

For the case $t_A + t_C \le t_B$, equation *(5.33)* becomes:

$$G^*H = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_A\, t_C & t_C \\ \varepsilon & \varepsilon & \varepsilon & t_B & t_A\, t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & t_A\, t_B t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & t_A\, t_B t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^3 & t_A\, {t_B}^2 t_C & t_A\, t_B t_C \end{bmatrix} \qquad (5.34)$$

The Eigen-value of Matrix G^*H is ${t_B}^2$, and the equations for process C are:

$$XC_{1,k} = {t_B}^2\, XB_{2,k-1} \qquad (5.35)$$

and:

$$XC_{2,k} = t_B XB_{2,k-1} \qquad (5.36)$$

Thus the inter-arrival time is constant and equal to t_B.

For the case $t_B < t_A + t_C$, equation *(5.33)* becomes:

$$G^*H = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_A\, t_C & t_C \\ \varepsilon & \varepsilon & \varepsilon & t_B & t_A\, t_C & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & t_A\, t_B t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & {t_B}^2 & t_A\, t_B t_C & t_A\, t_C \\ \varepsilon & \varepsilon & \varepsilon & t_A\, {t_B}^2 t_C & {t_A}^2\, t_B\, {t_C}^2 & {t_A}^2\, {t_C}^2 \end{bmatrix} \qquad (5.37)$$

with Eigen-value ${t_A}^2\, {t_C}^2$, and the equations for process C are:

$$XC_{1,k} = t_A\, t_C XC_{2,k-1} \qquad (5.38)$$

and:

$$XC_{2,k} = t_A\, t_C XC_{1,k} \qquad (5.39)$$

Thus the inter-arrival time is constant and equal to $t_A + t_C$.

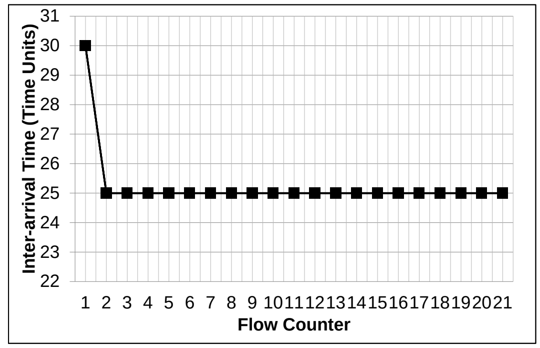
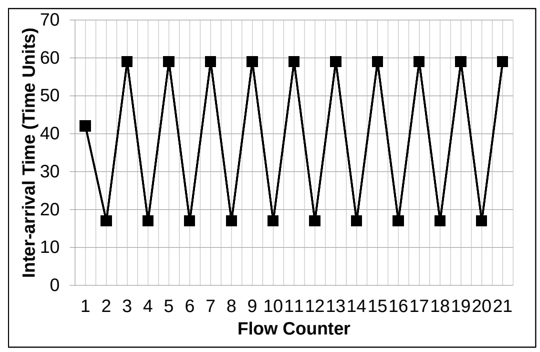
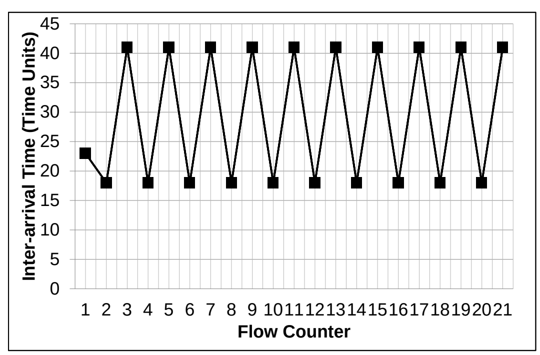
5.5.6. Processing Times $\boldsymbol{t_A < t_C < t_B}$

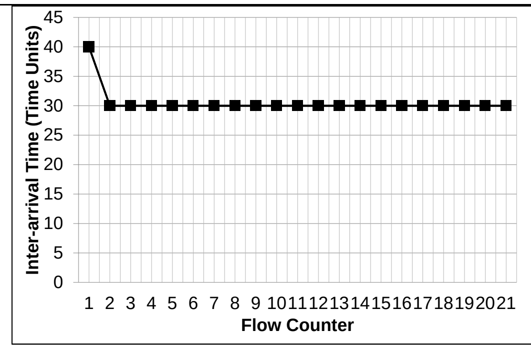
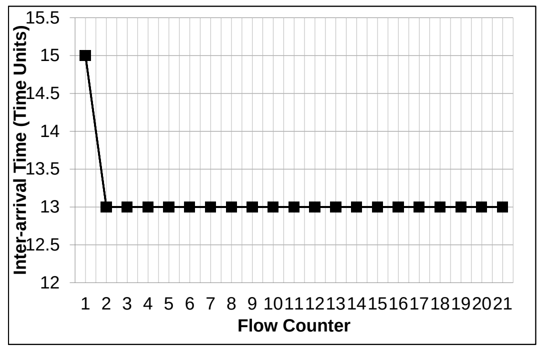
This case is exactly the same as case V with inter-arrival time equal to t_B for $t_A + t_C \le t_B$ and equal to $t_A + t_C$ for $t_B < t_A + t_C$.

A digital discrete events simulation model was created using the discrete-event simulation software FlexSim (Beaverstock, Greenwood et al. 2011) to verify the results of the max-plus model.

Table 5.1 includes a summary of the results along with a plot of the inter-arrival times of process C obtained from the simulation model.

Table 5-1 Summary of inter-arrival time behavior for different scenarios.

Case number	Processing times	Inter-arrival time for process C	Inter-arrival time plot from simulation
I	$TC < TB < TA$	$TA + TC$	$TA = 15,\ TB = 12,\ TC = 10.$ 
II & III	$TB < TC < TA$ & $TB < TA < TC$	$\begin{cases} 2TA + TC \\ TC \end{cases}$	$TA = 21,\ TB = 13,\ TC = 17.$ 
IV	$TA < TB < TC$	$\begin{cases} TA + TB + TC \\ TC \end{cases}$	$TA = 10,\ TB = 13,\ TC = 18.$ 
V & VI	$TC + TA < TB$	TB	$TA = 10,\ TB = 13,\ TC = 30.$

<table>
<tr><td></td><td></td><td></td><td>
</td></tr>
<tr><td></td><td>$\{TC, TA\}$
$< TB$
$< TA + TC$</td><td>$TA + TC$</td><td>$TA = 6,\ TB = 7,\ TC = 9.$

</td></tr>
</table>

5.6. Discussion and Conclusions

A max-plus algebraic model for representing re-entrant manufacturing systems such as automotive fuel injector assembly system, a semiconductor manufacturing cell, or an automated storage-retrieval system has been developed. Unlike other methods for modeling re-entrant lines, the max-plus model equations describe the system for any schedule and machine processing times. To account for decision points, extra variables were inserted and their values were analyzed as a function of the processing times, and thus the system was completely defined for any processing time scenario. The developed model offers an algebraic state-space equation in the form of $\boldsymbol{X_k} = \boldsymbol{A}\,\boldsymbol{X_{k-1}}$ where $\boldsymbol{X_k}$ is a vector of starting times on the different machines and A is a constant matrix which is function of the processing times of the machines only. Analyzing matrix $\boldsymbol{A}$ using the developed model; the steady state inter-arrival time can be found as a function of the processing times and the effect of changing each processing time can be obtained directly without the need for numerous simulation runs. The analysis yielded a complete description of all possible patterns of inter-arrival. In some cases, the steady state inter-arrival time was a constant number while in other cases it fluctuated between two values. As the studied re-entrant system can be a subsystem of a larger automated manufacturing system, a full understanding of the inter-arrival pattern is important in balancing the next stages especially for facilities following a Just-In-Time production policy.

Development of a max-plus model of a manufacturing system is relatively easy and straight-forward. Sub-systems can be modeled as independent modules and connected together in a block diagram like model (Imaev and Judd 2009). This feature allows overcoming the complexity and difficulty that can arise in systems larger than the one modeled in this. The only difficulty is in determining where to insert the variables that change due to the decision points and to determine the relation between their values and the values of the system parameters (processing times in this case). This presented a model for a small system with only two pallets and the dynamics of the system were captured by a 6×6 matrix with three variables representing the processing times of the three processes. Increasing the number of pallets would increase the size of the model; however, the calculations and analytical procedures remain the same.

Discrete event simulation remains a more versatile and easier to use modeling tool when the requirement is to evaluate the system under a given set of conditions. However, when full analysis and understanding of the effect of changing system parameters on the output over time, a parametric model is more useful and efficient.

CHAPTER 6: DISCUSSION AND CONCLUSIONS

6.1. Discussion and Overview

There is a need for a mathematical tool that can relate changes in manufacturing systems parameters to the overall system performance in a quick and efficient manner. Such a tool would be very useful to decision makers on all levels during both design and operation phases of any manufacturing system. Available tools for modeling manufacturing systems offering insights into the effect of different system parameters on the overall system behavior include *Queuing Theory, Markov Chains, Petri Nets, Discrete Event Simulation, and Max-plus Algebra*.

Both queuing theory and Markov chains are tools that deal with the *average* system performance over long time periods and thus are not very useful in short-term system analysis and control and give little insight into the system's dynamics and behavior. Petri Nets is more of a logical tool that gives qualitative analysis of the system such as detecting deadlocks but cannot give quantitative analysis on the system behavior.

Discrete event simulation is an excellent tool for the analysis of manufacturing systems' behavior and can give detailed picture of the system, however it is time consuming and can give information on the system only for the given simulated system parameters. In order to use discrete event simulation to gain insight into the effect of a given system parameter on the overall behavior, numerous simulation runs would be required.

Max-plus algebra is an algebraic mathematical formulation that can be used to model manufacturing systems in linear state-space like equations. By modeling manufacturing systems using max-plus algebra, one can arrive at mathematical equations that can be used in the analysis and control of manufacturing systems. The use of max-plus algebra in modeling and analysis of manufacturing systems started in the nineteen eighties, however it use both commercially and academically has been limited. This is mainly because using the tool requires special mathematical background and because there are no user-friendly tools that facilitate the use of max-plus algebra in modeling and analysis of systems.

In this research different models and tools have been developed to make max-plus algebra more accessible to engineers and managers with little or no background in its mathematical foundation.

They can enable engineers and managers to use max-plus equations in analyzing manufacturing systems and testing different what-if scenarios efficiently in both design and operation stages.

The methods developed in this research could be further developed into a commercial analysis tool for use by engineers and managers. Development of such commercial tools requires a user interface which could be similar to those found in discrete event simulation tools.

While no tool is suitable for modeling all systems, a practical tool should be capable of modeling and providing analysis to a wide range of systems types. In this thesis max-plus algebra was used to model and analyze manufacturing systems that are different in their structure and theory of operation. For each system, the max-plus equations was capable of simulating the behavior of the system and providing insights that are useful in both the design and operation phases of a manufacturing system.

In this thesis, transfer times and setup times were not included in the models. Such details can be easily represented by adding an extra station with a given setup time. More details can be added to the models, and these details add to accuracy of the models. For all the developed models in this thesis, discrete event simulation was used for verification. For every system modeled by max-plus equations, an identical model was developed using discrete event simulation and the results from both models were compared. This verification process is required when developing new methods or system models, but will not be necessary each time max-plus algebra is used in modeling or analysis.

Like any other tool, max-plus algebra has some limitations. The two most important limitations are requiring deterministic processing times, and its inability to handle decisions made during operating the line. The first limitation prevents modeling stochastic processing times, manual processes, and machine breakdowns. While some research has been done on incorporating stochastic times in max-plus algebra, it still requires more research. The second limitation causes the difficulty of modeling systems with scheduling decisions, like re-entrant systems and job shops. There exists some max-plus algebra models of job shops, but these models determine the complete schedule apriori then uses max-plus algebra just as a simulation tool. The modeling method used in chapter 5 forms a basis for overcoming this limitation. These two limitations increase the model complexity when using max-plus algebra, while for example an increase in the number of stations would not increase the model complexity.

6.2. Research Significance

The research presented in this dissertation aims at transforming max-plus algebra from a tool used by mathematicians and academics into a practical engineering tool that can be easily used by engineers and managers in the industry. This was achieved by working in two directions, the first direction aimed at making max-plus algebraic models more accessible to end users in industry and allowing them to use these models without the need to learn and understand the max-plus algebra. That is achieved through developing a method for automatically generating the max-plus equations for manufacturing systems and allowing end users to use these equations in analyzing the system and tuning its parameters. The second part aimed at increasing the appeal and practicality of the tool by developing max-plus models for manufacturing system types that were never before modeled using max-plus algebra.

The first direction (chapter 3) presented a method for automatically generating max-plus equations for given manufacturing lines, which can be then easily used in simulating what-if scenarios quickly and efficiently and gaining insight into the effect of each system parameter on the overall system behavior. All what is needed as input is the structure of the line presented in an adjacency matrix and the output is a parametric closed form max-plus equations of the system that can be used to evaluate several system parameters such as the effect of buffer sizes on the line idle time or the effect of the processing time of each station on the total line make-span. The ease of generating these equations can also make it very useful in comparing different possible system configurations.

The second direction (chapters 4 and 5) used max-plus algebra to model mixed-model assembly lines and Re-entrant lines. The standard problem of mixed model assembly lines is an optimization problem where the different assembly tasks need to be distributed among the stations to achieve some given criteria. Modeling such systems using closed form parametric equations allows for analyzing the effect of changes in the system parameters on the performance without the need to perform numerous optimizations. This can be used to test the robustness of a given solution to changes in the system parameters.

In chapter 5, Re-entrant manufacturing lines were modeled using max-plus algebra and produced system equations that enabled analyzing the dynamics of such systems in transient and steady state stages. The equations can be used to determine the ranges of values for stations processing times for which the steady state inter-arrival time would be a constant or oscillating. This can be most useful when a constant inter-arrival time is required in steady-state operation.

Models that can provide better insight into system performance and the effect of changes to system parameters allow better decision making, which translates to lower cost, shorter lead times, and higher efficiency.

6.3. Research Contributions and Novelty

Contributions can be summarized in these points:

- Developing a new method for automatic generation of max-plus equations of manufacturing lines of any structure while taking into account finite buffers and parallel identical stations. The produced equations were then used in parametric analysis both in design and operation phases. The developed method was used to model and analyze a real industrial system and provided valuable insight such as the optimal buffer size after which more buffer space does not decrease the line's idle time, and the relationship between the processing time of any station and the total idle time of the line.

- For the first time, modeling MMALs and Re-entrant manufacturing systems using max-plus equations. For MMALs, the developed equations are used in comparing given sequences of demand mix over a range of processing times of assembly tasks and thus can be used in robustness analysis of given sequences. The equations are also used in analyzing different line performance measures such as length of line or total workers idle time while considering one (or more) of the line parameters as a variable. For Re-entrant manufacturing systems, the developed equations can be used to tune the system to achieve steady state faster, which could be very important when such a system feeds other conventional systems.

6.4. Future Work

Max-plus algebra is a mathematical tool that has a great potential in modeling, simulation, and control of manufacturing systems. The work done in this research can be expanded in many ways to further the benefit of this elegant tool. The following points are possible work extensions that are useful and possible:

- Expanding the method for generating max-plus equations to cover manufacturing systems other than flow lines, such as job shops and cellular manufacturing systems.
- Include machine breakdowns in Max-plus models using variable processing times. This can be used in comparing the loss due to breakdown in different system configurations.
- Investigating new applications of max-plus algebra in modeling, simulating, and controlling manufacturing systems such as flexible manufacturing systems.
- Modeling complex re-entrant manufacturing lines featuring large number of stations and more than one re-entrance.
- Applying the already available control theoretic tools of max-plus algebra in actual manufacturing systems.
- Expanding the developed models and methods to accommodate stochastic systems in which processing times are not deterministic.

REFERENCES

Addad, B., S. Amari, et al. (2010). Linear time-varying (Max,+) representation of conflicting timed event graphs. 10th Int. Workshop on Discrete Event Systems, Berlin, Germany.

Alfaro, M. D. and J. M. Sepulveda (2006). "Chaotic behavior in manufacturing systems." International Journal of Production Economics **101**(1): 150-158.

Amari, S., I. Demongodin, et al. (2005). Sizing, cycle time and plant control using dioid algebra. Supply Chain Optimisation, Springer**:** 71-85.

Baccelli, F. and D. Hong (2000). TCP is Max-Plus Linear and what it tells us on its throughput. ACM SIGCOMM Computer Communication Review, ACM.

Baccelli, F. L., G. Cohen, et al. (1992). Synchronization and linearity, Wiley New York.

Bard, J. F., E. Dar-El, et al. (1992). "An analytic framework for sequencing mixed model assembly lines." The International Journal of Production Research **30**(1): 35-48.

Beaumariage, T. and K. Kempf (1994). The nature and origin of chaos in manufacturing systems. IEEE/SEMI Advanced Semiconductor Manufacturing Conference, Cambridge, MA, IEEE.

Beaverstock, M., A. Greenwood, et al. (2011). Applied Simulation. Modeling and Analysis using Flexsim. Flexsim Software Products, FlexSim Software Products, Inc.

Bispo, C. F. and S. Tayur (2001). "Managing simple re-entrant flow lines: theoretical foundation and experimental results." IIE Transactions **33**(8): 609-623.

Bogdan, S., Z. Kovacic, et al. (2004). "A matrix approach to an FMS control design [flexible manufacturing systems]." Robotics & Automation Magazine, IEEE **11**(4): 92-109.

Bohnen, F., M. Buhl, et al. (2013). "Systematic procedure for leveling of low volume and high mix production." CIRP Journal of Manufacturing Science and Technology **6**(1): 53-58.

Boukra, R., S. Lahaye, et al. (2013). "New representations for (max,+) automata with applications to performance evaluation and control of discrete event systems." Discrete Event Dynamic Systems **25**(1-2): 1-28.

Boysen, N., M. Fliedner, et al. (2008). "Assembly line balancing: which model to use when?" International Journal of Production Economics **111**(2): 509-528.

Boysen, N., M. Fliedner, et al. (2009). "Sequencing mixed-model assembly lines: Survey, classification and model critique." European Journal of Operational Research **192**(2): 349-373.

Boysen, N., M. Kiel, et al. (2011). "Sequencing mixed-model assembly lines to minimise the number of work overload situations." International Journal of Production Research **49**(16): 4735-4760.

Cassandras, C. G. and S. Lafortune (2007). Introduction to discrete event systems, Springer.

Choi, S.-W. and Y.-D. Kim (2006). "Minimizing makespan on a two-machine re-entrant flowshop." Journal of the Operational Research Society **58**(7): 972-981.

Chryssolouris, G., N. Giannelos, et al. (2004). "Chaos theory in production scheduling." CIRP Annals-Manufacturing Technology **53**(1): 381-383.

Chutima, P., W. Nimmano, et al. (2003). "Application of fuzzy genetic algorithm for sequencing in mixed-model assembly line with processing time." International Journal of Industrial Engineering-Theory Applications and Practice **10**(4): 325-331.

Cohen, G., D. Dubois, et al. (1985). "A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing." Automatic Control, IEEE Transactions on **30**(3): 210-220.

Cohen, G., S. Gaubert, et al. (1999). "Max-plus algebra and system theory: where we are and where to go now." Annual reviews in control **23**: 207-219.

Correia, A., A. Abbas-Turki, et al. (2009). "A dioid model for invariant resource sharing problems." Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **39**(4): 770-781.

De Schutter, B. and T. Van Den Boom (2001). "Model predictive control for max-plus-linear discrete event systems." Automatica **37**(7): 1049-1056.

Delta-Tech. "Delta - Tech Assembly Lines." from http://www.deltatech.hu/.

Di Febbraro, A., R. Minciardi, et al. (1994). Performance optimization in manufacturing systems by use of max-plus algebraic techniques. Systems, Man, and Cybernetics, 1994.'Humans, Information and Technology'., 1994 IEEE International Conference on, IEEE.

Diaz-Rivera, I., D. Armbruster, et al. (2000). "Periodic orbits in a class of re-entrant manufacturing systems." Mathematics of Operations Research **25**(4): 708-725.

Dini, G., F. Failli, et al. (1999). "Generation of optimized assembly sequences using genetic algorithms." CIRP Annals - Manufacturing Technology **48**(1): 17-20.

Dong, M. and F. He (2012). "A new continuous model for multiple re-entrant manufacturing systems." European Journal of Operational Research **223**(3): 659-668.

Dorot ((2001)). Dorot automatic control valves.

Doustmohammadi, A. and E. W. Kamen (1995). Direct generation of event-timing equations for generalized flow shop systems. Photonics East'95, International Society for Optics and Photonics.

ElMaraghy, H. and M. Manns (2009). "Synchronisation of inter-arrival times in manufacturing systems with main and side loops." International Journal of Production Research **47**(7): 1931-1954.

ElMaraghy, W., H. ElMaraghy, et al. (2012). "Complexity in engineering design and manufacturing." CIRP Annals-Manufacturing Technology **61**(2): 793-814.

Fattahi, P. and M. Salehi (2009). "Sequencing the mixed-model assembly line to minimize the total utility and idle costs with variable launching interval." The International Journal of Advanced Manufacturing Technology **45**(9-10): 987-998.

Gökcen, H. and E. Erel (1998). "Binary integer formulation for mixed-model assembly line balancing problem." Computers & Industrial Engineering **34**(2): 451-461.

Goto, H. (2013). "Model predictive control-based scheduler for repetitive discrete event systems with capacity constraints." International Journal of Optimization & Control: Theories & Applications **3**(2).

Goto, H., Y. Hasegawa, et al. (2007). Efficient scheduling focusing on the duality of MPL representation. Computational Intelligence in Scheduling, 2007. SCIS'07. IEEE Symposium on, IEEE.

Goto, H., K. Shoji, et al. (2007). On the Max-Plus Linear representation for event-varying systems with buffer and order constraints. Industrial Engineering and Engineering Management, 2007 IEEE International Conference on, IEEE.

Groover, M. P. (2007). Automation, production systems, and computer-integrated manufacturing, Prentice Hall Press.

Grzymkowski, R., A. Kapusta, et al. (2008). Mathematica 6, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego.

He, F., M. Dong, et al. (2011). "Modeling and Analysis of Reentrant Manufacturing Systems: Micro-and Macroperspectives." Mathematical Problems in Engineering **2011**: 17.

Heidergott, B., G. J. Olsder, et al. (2006). Max Plus at work: modeling and analysis of synchronized systems: a course on Max-Plus algebra and its applications, Princeton University Press.

Ho, Y. C. (1989). "Introduction to special issue on dynamics of discrete event systems." Proceedings of the IEEE **77**(1): 3-6.

Houssin, L. (2011). Cyclic jobshop problem and (max, plus) algebra. World IFAC Congress (IFAC 2011).

Houssin, L., S. Lahaye, et al. (2007). "Just in time control of constrained (max,+)-linear systems." Discrete Event Dynamic Systems **17**(2): 159-178.

Hu, S. J., J. Ko, et al. (2011). "Assembly system design and operations for product variety." CIRP Annals-Manufacturing Technology **60**(2): 715-733.

Hyun, C. J., Y. Kim, et al. (1998). "A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines." Computers & Operations Research **25**(7): 675-690.

Imaev, A. and R. P. Judd (2008). Hierarchial modeling of manufacturing systems using max-plus algebra. American Control Conference, 2008, IEEE.

Imaev, A. and R. P. Judd (2009). Block diagram-based modeling of manufacturing systems using max-plus algebra. American Control Conference, 2009. ACC'09., IEEE.

Jean-Marie, A. and G. J. Olsder (1996). "Analysis of stochastic min-max-plus systems: Results and conjectures." Mathematical and computer modelling **23**(11): 175-189.

Kashkoush, M. and H. ElMaraghy (2014). "Consensus tree method for generating master assembly sequence." Production Engineering **8**(1-2): 233-242.

Kilbridge, L. W. M. and L. Wester (1963). The Assembly Line Mixed Model Sequencing Problem. Proceedings of the Third International Conference on Operation research.

Kumar, P. (1993). "Re-entrant lines." Queueing systems **13**(1): 87-110.

Kumar, S. and P. Kumar (2001). "Queueing network models in the design and analysis of semiconductor wafer fabs." Robotics and Automation, IEEE Transactions on **17**(5): 548-561.

Lahaye, S., J. L. Boimond, et al. (2004). "Linear periodic systems over dioids." Discrete Event Dynamic Systems **14**(2): 133-152.

Lee, T.-E. (2000). "Stable earliest starting schedules for cyclic job shops: a linear system approach." International Journal of Flexible Manufacturing Systems **12**(1): 59-80.

Li, J., D. E. Blumenfeld, et al. (2009). "Throughput analysis of production systems: recent advances and future topics." International Journal of Production Research **47**(14): 3823-3851.

Lu, S. H. and P. Kumar (1991). "Distributed scheduling based on due dates and buffer priorities." Automatic Control, IEEE Transactions on **36**(12): 1406-1416.

Maia, C. A., L. Hardouin, et al. (2013). Some results on the feedback control of max-plus linear systems under state constrains. Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, IEEE.

Manns, M. and H. ElMaraghy (2009). "Inter-arrival time patterns in manufacturing systems with main and side loops." International Journal of Production Research **47**(10): 2721-2744.

Matyas, K. and S. Auer (2012). "Combination of planning methods in a comprehensive production planning approach for sequenced production lines." CIRP Annals-Manufacturing Technology **61**(1): 445-448.

Miltenburg, J. and G. Sinnamon (1989). "Scheduling mixed-model multi-level just-in-time production systems." The International Journal of Production Research **27**(9): 1487-1509.

Morrison, J. R. (2010). "Deterministic flow lines with applications." Automation Science and Engineering, IEEE Transactions on **7**(2): 228-239.

Nait-Sidi-Moh, A., M.-A. Manier, et al. (2005). "max-plus algebra Modeling for a Public Transport System." Cybermatics and Systems: An international Journal **36**(2): 165-180.

Narahari, Y. and L. Khan (1996). "Performance analysis of scheduling policies in re-entrant manufacturing systems." Computers & Operations Research **23**(1): 37-51.

Ott, E. (2002). Chaos in dynamical systems, Cambridge university press.

Park, K. and J. R. Morrison (2010). Performance evaluation of deterministic flow lines: Redundant modules and application to semiconductor manufacturing equipment. Automation Science and Engineering (CASE), 2010 IEEE Conference on, IEEE.

Reddy, Y., G. R. Janardhana, et al. (2009). "Modular modelling and performance evaluation of manufacturing systems using Max–Plus Algebra." International Journal of Industrial and Systems Engineering **4**(5): 577-596.

Ren, S., D. Xu, et al. (2007). "Timed event graph-based cyclic reconfigurable flow shop modelling and optimization." International Journal of Production Research **45**(1): 143-156.

Schmitz, J., D. Van Beek, et al. (2002). "Chaos in discrete production systems?" Journal of Manufacturing Systems **21**(3): 236-246.

Seleim, A. and H. ElMaraghy (2014). "Parametric analysis of Mixed-Model Assembly Lines using max-plus algebra." CIRP Journal of Manufacturing Science and Technology **7**(4): 305-314.

Seleim, A. and H. ElMaraghy (2015). "Generating max-plus equations for efficient analysis of manufacturing flow lines." Journal of Manufacturing Systems **37**: 426-436.

Singh, M. and R. P. Judd (2014). "Efficient calculation of the makespan for job-shop systems without recirculation using max-plus algebra." International Journal of Production Research **52**(19): 5880-5894.

Suk, J. B. and C. G. Cassandras (1989). Optimal control of a storage-retrieval queuing system. Decision and Control, 1989., Proceedings of the 28th IEEE Conference on, IEEE.

Tanaka, A., S. Masuda, et al. (2009). Process pogressing scheduling for repetitive systems using max-plus algebra under multiple output constraints. ICCAS-SICE, 2009, IEEE.

Tavakoli, N. B. and P. Fattahi (2012). "Sequencing Mixed-Model Assembly Line under a JIT-Approach." Applied Mechanics and Materials **110**: 4324-4329.

Thomopoulos, N. T. (1967). "Line balancing-sequencing for mixed-model assembly." Management Science **14**(2): B-59-B-75.

Uddin, M. K., M. C. Soto, et al. (2010). "An integrated approach to mixed-model assembly line balancing and sequencing." Assembly Automation **30**(2): 164-172.

van den Boom, T. J. and B. De Schutter (2006). "Modelling and control of discrete event systems using switching max-plus-linear systems." Control Engineering Practice **14**(10): 1199-1211.

Wang, C. and J. Li (2010). "Approximate analysis of reentrant lines with Bernoulli reliability model." Automation Science and Engineering, IEEE Transactions on **7**(3): 708-715.

Wiendahl, H. P. and H. Scheffczyk (1999). "Simulation based analysis of complex production systems with methods of nonlinear dynamics." CIRP Annals-Manufacturing Technology **48**(1): 357-360.

APPENDIX A

Theorem 2.1:

An equation is the general form:

$$\boldsymbol{X} = \boldsymbol{A}\,\boldsymbol{X} \oplus \boldsymbol{B}\,\boldsymbol{U} \qquad (2.1)$$

where $\boldsymbol{X}$ is an $n \times 1$ vector of variables, $\boldsymbol{U}$ is an $m \times 1$ vector of inputs, $\boldsymbol{A}$ is an $n \times n$ square matrix and $\boldsymbol{B}$ is an $n \times m$ matrix, has a solution :

$$\boldsymbol{X} = \boldsymbol{A}^{*}\,\boldsymbol{B}\,\boldsymbol{U}$$

where $\boldsymbol{A}^{*}$ is defined as:

$$\boldsymbol{A}^{*} = \boldsymbol{E} \oplus \boldsymbol{A} \oplus \boldsymbol{A}^{2} \oplus \ldots \oplus \boldsymbol{A}^{\infty}$$

and if all the circuit weights of the directed communication graph (Heidergott, Olsder et al. 2006) of matrix **A** are negative, then:

$$\mathbf{A}^{*} = (\boldsymbol{E} \oplus \boldsymbol{A} \oplus \boldsymbol{A}^{\boldsymbol{2}} \oplus \ldots \oplus \boldsymbol{A}^{\boldsymbol{n-1}})$$

Proof (Heidergott, Olsder et al. 2006):

By substituting $\boldsymbol{X}$ in the R.H.S. of equation (2.1) by the whole R.H.S. of the same equation, we get:

$$\boldsymbol{X} = \boldsymbol{A}(\boldsymbol{A}\,\boldsymbol{X} \oplus \boldsymbol{B}\,\boldsymbol{U}) \oplus \boldsymbol{B}\,\boldsymbol{U}$$

expanding, we get:

$$\boldsymbol{X} = \boldsymbol{A}^{\boldsymbol{2}}\boldsymbol{X} \oplus \boldsymbol{A}\,\boldsymbol{B}\,\boldsymbol{U} \oplus \boldsymbol{B}\,\boldsymbol{U} = \boldsymbol{A}^{\boldsymbol{2}}\boldsymbol{X} \oplus (\boldsymbol{A} \oplus \boldsymbol{E})\boldsymbol{B}\,\boldsymbol{U}$$

where $\boldsymbol{E}$ is identity of the matrix product. Iterating the substitution n-1 times we get:

$$\boldsymbol{X} = \boldsymbol{A}^{\boldsymbol{n}}\boldsymbol{X} \oplus (\boldsymbol{E} \oplus \boldsymbol{A} \oplus \boldsymbol{A}^{\boldsymbol{2}} \oplus \ldots \oplus \boldsymbol{A}^{\boldsymbol{n-1}})\,\boldsymbol{B}\,\boldsymbol{U}$$

If A is the incidence matrix of an acyclic matrix, then $lim_{\mathrm{n}\to\infty}\,\boldsymbol{A}^{\boldsymbol{n}} =$ Null and we get:

$$\boldsymbol{X} = (\boldsymbol{E} \oplus \boldsymbol{A} \oplus \boldsymbol{A}^{\boldsymbol{2}} \ldots \oplus \boldsymbol{A}^{\boldsymbol{n-1}})\,\boldsymbol{B}\,\boldsymbol{U} = \mathbf{A}^{*}\,\mathbf{B}\,\mathbf{U}$$

APPENDIX B

This section presents the *Mathematica* code used to define max-plus operations used throughout the dissertation. Sentences between “(*” and “*)” are comments to explain the code.

```
(*
Declaring the symbol "∊" to be equal to negative infinity
*)
∊=-Infinity;

(*
Declaring Function for Matrix Max-Plus Addition "⊕"
The function takes two matrices "x" and "y" and returns the matrix "x ⊕ y"
*)
MMPlus[x_,y_]:=  Simplify [Table [ Max [ x [[i,j]] , y [[i,j]] ] , {i , First [Dimensions
[x]] } , {j  , Last [Dimensions [y]] } ]]

(*
Declaring Function for Matrix Max-Plus Multiplication "⊗"
The function takes two matrices "x" and "y" and returns the matrix "x ⊗ y"
*)
MMMult [x_,y_]:= Simplify [Table  [ Max [ x [[i]]+ y [[All,j]]]] , {I , First [Dimensions
[x]] } , {j , Last [Dimensions [y]] } ]]

(*
Declaring Function for Matrix Power
The function takes a matrix "x" and an integer "n" and returns the matrix "x^n", i.e.
x⊗x⊗...x(for n times)
*)
Mpower [x_,n_]:=
        Module [{M=x , power = n} ,
        For [i=1 , i<power , i++ , M = MMMult [M,x] ];
 M]

(*
Declaring Function for existence of x^+ for a maxtrix "x"
*)
MPlus [x_]:=
        Module [{M=x , power=1000},
        mplus = M;
        For [i=1 , i<power>0 ,  i++ ,
                {M = MMMult [M,x] , If [Max[ Flatten[M]] <0 , Break [] ];
                mplus = MMPlus [mplus , M]} ];
 mplus]

(*
Declaring Function for existence of x^* for a matrix "x"
*)
```

```
MStar [x_]:=
        Module [{M=x},
        ME = Table [If [i==j,0,ϵ] , {i , Length[M]},{j , Length[M]}];
        M = MMPlus [MPlus[M] , ME];
 M]

(*
Returns Eigenvalue of Matrix "A" given an EigenVector "x" (x has to be a column!)
The Eigenvalue is calculated according to Karp's algorithm (Heidergott et al.2006)
*)
EigenA [A_,X_]:=
        Module [{n = Length[A] , x = Transpose[X]},
        For [i = 0 , i< n , i++,
                x= Join [x , Transpose [MMMult [A,Transpose [{Last[x]}] ] ] ];
         ];
        L = {};
        For [j=1 , j<n+1 , j++ ,
                l={};
                For [i=0, i<n , i++,
                        If [x [[n+1 , j]] <0 , l = Join [l,{0}] ; Break[]];
                        l = Join [l , {(x [[n+1,j]] –x [[i+1,j]] ) / (n-i) } ];
                ];
                L=Join[L,{Min[l]}];
         ];
        L=Max[L];
 L]
```

APPENDIX C

The values of $\widehat{A}$, $\widehat{AB}$ and $\widehat{B}$ in equation 3.19 for each of the configurations in figure 3.11 are given as:

for configuration 1

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} t_C & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ t_C^2 & t_D & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_B & \varepsilon & \varepsilon & \varepsilon \\ t_C^2 t_D & \varepsilon & t_B^2 & t_E & \varepsilon & \varepsilon \\ \varepsilon & t_D^2 & \varepsilon & \varepsilon & t_A & \varepsilon \\ t_C^2 t_D t_E & t_D^2 t_E & t_B^2 t_E & t_E^2 & t_A^2 & t_F \end{bmatrix}, \widehat{\boldsymbol{B}} = \begin{bmatrix} 0 & \varepsilon & \varepsilon \\ t_C & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon \\ t_C t_D & t_B & \varepsilon \\ \varepsilon & \varepsilon & 0 \\ t_C t_D t_E & t_B t_E & t_A \end{bmatrix} \textit{and}$$

$$\widehat{\boldsymbol{AB}} = \begin{bmatrix} \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_C & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & t_C t_D & \varepsilon & t_D \oplus t_B & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_C t_D t_E & \varepsilon & t_D t_E \oplus t_B t_E & \varepsilon & t_A \end{bmatrix}, \qquad (B.1)$$

for configuration 2

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} t_{C^*} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_B & \varepsilon & \varepsilon & \varepsilon \\ t_{C^*}^2 & t_B^2 & t_E & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & t_A & \varepsilon \\ t_{C^*}^2 t_E & t_B^2 t_E & t_E^2 & t_A^2 & t_F \end{bmatrix}, \widehat{\boldsymbol{B}} = \begin{bmatrix} 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ t_{C^*} & 0 & \varepsilon \\ \varepsilon & t_B & 0 \\ t_{C^*} t_E & t_B t_E & t_A \end{bmatrix} \textit{and}$$

$$\widehat{\boldsymbol{AB}} = \begin{bmatrix} \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_C \oplus t_B & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & (t_C \oplus t_B) t_D & \varepsilon & t_D \oplus t_A \end{bmatrix}, \qquad (B.2)$$

and for configuration 3

$$\hat{\boldsymbol{A}} = \begin{bmatrix} t_C & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ t_C^2 & t_D & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_B & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ t_C^2 t_D & t_D^2 & t_B^2 & t_{E^*} & \varepsilon & \varepsilon & \varepsilon \\ t_C^2 t_D t_{E^*} & t_D^2 t_{E^*} & t_B^2 t_{E^*} & t_{E^*}^2 & t_G & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & t_A & \varepsilon \\ t_C^2 t_D t_{E^*} t_G & t_D^2 t_{E^*} t_G & t_B^2 t_{E^*} t_G & t_{E^*}^2 t_G & t_G^2 & t_A^2 & t_F \end{bmatrix}, \quad \hat{\boldsymbol{B}} = \begin{bmatrix} 0 & \varepsilon & \varepsilon \\ t_C & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon \\ t_C t_D & t_B & \varepsilon \\ t_C t_D t_{E^*} & t_B t_{E^*} & \varepsilon \\ \varepsilon & \varepsilon & 0 \\ t_C t_D t_E t_G & t_B t_{E^*} t_G & t_A \end{bmatrix} \textit{ And } \widehat{\boldsymbol{AB}} =$$

$$\begin{bmatrix} \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_C & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_C t_D & \varepsilon & t_D \oplus t_B & 0 & \varepsilon & \varepsilon \\ \varepsilon & t_C t_D t_{E^*} & \varepsilon & (t_D \oplus t_B) t_{E^*} & t_{E^*} & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & t_C t_D t_{E^*} t_G & \varepsilon & (t_D \oplus t_B) t_{E^*} t_F & t_{E^*} t_F & \varepsilon & t_F \oplus t_A \end{bmatrix}. \quad \textit{(B.3)}$$

APPENDIX D

The following AMPL code was developed following the mathematical programming model presented in (Bard, Dar-El et al. 1992).

```
“

param vc;
param M;
param I;
param J;
param d {m in 1..M};
param t {j in 1..J,m in 1..M};
param w;
var x {i in 1..I,m in 1..M} binary;
var z {i in 1..I,j in 1..J} >=0;
var y {j in 1..J} >=0;

minimize OF:
  sum{j in 1..J} y[j];
subject to c_1 {i in 1..I}:
 sum {m in 1..M} x[i,m] = 1;
subject to c_2 {m in 1..M}:
 sum {i in 1..I} x[i,m] = d[m];
subject to c_3 {i in 1..I-1,j in 1..J}:
 z[i+1,j] >= z[i,j] + vc * sum {m in 1..M} x[i,m]*t[j,m] - w;
subject to c_4 {i in 1..I,j in 1..J}:
 y[j] >= z[i,j] + vc * sum {m in 1..M} x[i,m]*t[j,m];

data;
param vc:= 1;
param M:= 4;
param I:= 24;
param J:= 5;
param w:= 42;
param t:=
```

[*,*]: 1 2 3 4:=

1	45	46	50	53
2	38	38	42	44
3	38	40	43	45
4	41	44	47	47
5	47	40	40	40;

param d:=

1	10
2	8
3	4
4	2;

“

VITA AUCTORIS

NAME: Abdulrahman Seleim

PLACE OF BIRTH: Windsor, Ontario, Canada

YEAR OF BIRTH: 1985

EDUCATION: B.Sc. in Mechanical Engineering from Ain Shams University, Cairo, Egypt, 2007

M.Sc. in Systems Design Engineering from the University of Waterloo, Waterloo, Ontario, Canada 2010

PUBLICATIONS: **Journal Publications:**

- *Seleim, A.,* and H. ElMaraghy (2015). "Generating max-plus equations for efficient analysis of manufacturing flow lines." Journal of Manufacturing Systems 37: 426-436.
- *Seleim, A.*, and H. ElMaraghy (2014). "Parametric analysis of Mixed-Model Assembly Lines using max-plus algebra." CIRP Journal of Manufacturing Science and Technology 7(4): 305-314.
- *Seleim, A.*, et al. (2012) Nonlinear Dynamics. Dynamics of a close-loop controlled MEMS resonator. Vol. 69, pp. 615-633.
- Towfighian, S., *Seleim, A.*, Abdel-Rahman, E., and Heppler, G. (2011) Journal of Micromechanics and Microengineering. A large-stroke electrostatic micro-actuator. Vol. 21 075023.