

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ

Διατριβή

**ΕΚΤΙΜΗΣΗ ΑΠΟΔΟΣΗΣ ΚΑΙ ΙΔΙΟΤΗΤΕΣ ΠΟΛΙΤΙΚΩΝ ΤΥΠΟΥ
ΚΑΝΒΑΝ ΓΙΑ ΤΟΝ ΣΥΝΤΟΝΙΣΜΟ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ-
ΑΠΟΘΕΜΑΤΩΝ ΠΟΛΛΑΠΛΩΝ ΣΤΑΔΙΩΝ**

υπό

ΣΤΥΛΙΑΝΟΥ ΚΟΥΚΟΥΜΙΑΛΟΥ

Διπλωματούχου Μηχανολόγου Μηχανικού Βιομηχανίας Πανεπιστημίου Θεσσαλίας, 1997

Υπεβλήθη για την εκπλήρωση μέρους των

απαιτήσεων για την απόκτηση του

Διδακτορικού Διπλώματος

2003



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΥΠΗΡΕΣΙΑ ΒΙΒΛΙΟΘΗΚΗΣ & ΠΛΗΡΟΦΟΡΗΣΗΣ
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 1157/1
Ημερ. Εισ.: 27-01-2004
Δωρεά: Συγγραφέως
Ταξιθετικός Κωδικός: Δ
658.5
ΚΟΥ


© 2003 Στυλιανός Κουκούμιαλος



Η έγκριση της διδακτορικής διατριβής από το Τμήμα Μηχανολόγων Μηχανικών Βιομηχανίας της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).

Εγκρίθηκε από τα Μέλη της Επταμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής
(Επιβλέπων)



Δρ. Γεώργιος Λυμπερόπουλος
Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών
Βιομηχανίας, Πανεπιστήμιο Θεσσαλίας

Δεύτερος Εξεταστής



Δρ. Γεώργιος Ταγαράς
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Αριστοτέλειο
Πανεπιστήμιο Θεσσαλονίκης

Τρίτος Εξεταστής



Δρ. Χρυσολέων Παπαδόπουλος
Καθηγητής, Τμήμα Μηχανικών Σχεδίασης Προϊόντων και
Συστημάτων, Πανεπιστήμιο Αιγαίου

Τέταρτος Εξεταστής



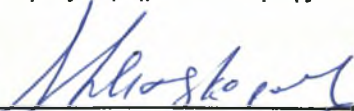
Δρ. Βασίλειος Κουικόγλου
Αναπληρωτής Καθηγητής, Τμήμα Μηχανικών Παραγωγής και
Βιομηχανικής Διοίκησης, Πολυτεχνείο Κρήτης

Πέμπτος Εξεταστής



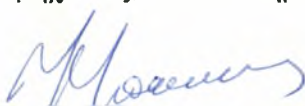
Δρ. Μιχαήλ Βιδάλης
Λέκτορας, Τμήμα Διοίκησης Επιχειρήσεων, Πανεπιστήμιο Αιγαίου

Έκτος Εξεταστής



Δρ. Αθανάσιος Ζηλιασκόπουλος
Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών
Βιομηχανίας, Πανεπιστήμιο Θεσσαλίας

Έβδομος Εξεταστής



Δρ. Ιωάννης Μπακούρος
Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών
Βιομηχανίας, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Πρώτα απ' όλα, θέλω να ευχαριστήσω τον επιβλέποντα της διατριβής μου, Αναπληρωτή Καθηγητή κ. Γιώργο Λυμπερόπουλο, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια της διδακτορικής μου δουλειάς. Επίσης, είμαι ευγνώμον στα υπόλοιπα μέλη της εξεταστικής επιτροπής της διατριβής μου, Καθηγητές κκ. Γιώργο Ταγαρά, Χρυσολέων Παπαδόπουλο, Βασίλη Κουϊκόγλου, Μιχάλη Βιδάλη, Θανάση Ζηλιασκόπουλο και Γιάννη Μπακούρο, για την προσεκτική ανάγνωση της εργασίας μου και για τις πολύτιμες υποδείξεις τους. Οφείλω ευχαριστίες στον Καθηγητή κ. Yves Dallery της Ecole Centrale του Παρισιού που μου υπέδειξε την προσεγγιστική μέθοδο που ανέπτυξα στο Κεφάλαιο 3. Ευχαριστώ τους συναδέλφους μου Νίκο Γουρνεζάκη και Σπύρο Παπασπύρου για την πολύτιμη βοήθειά τους στον προγραμματισμό με Matlab, και τους Ανδρέα Δώριζα, Ισίδωρο Τσική, Τριαντάφυλλο Λιάπη και Πέτρο Τρυφονόπουλο για την συνδρομή τους στις προσομοιώσεις του Κεφαλαίου 2. Ευχαριστώ τον Χρήστο Πανούτσο για τις φιλικές συμβουλές του, και τις φίλες μου Ελένη Καμούτση και Αθηνά Οικονόμου για την ηθική υποστήριξή τους. Επίσης, ευχαριστώ την Μαρία Κατέρη για την κατανόησή της, ιδιαίτερα κατά τη διάρκεια των τελευταίων μηνών της προσπάθειάς μου. Πάνω απ' όλα, είμαι ευγνώμων στους γονείς μου, Γιάννη και Αγγελική Κουκούμιαλου για την ολόψυχη αγάπη και υποστήριξή τους όλα αυτά τα χρόνια. Αφιερώνω αυτή την διατριβή στην μητέρα μου και στην μνήμη του πατέρα μου, ο οποίος ήθελε τόσο πολύ να με δει να την τελειώνω. Τέλος, ευχαριστώ τον μικρότερο αδελφό μου, Γιώργο Κουκούμιαλο, για την συμπαράστασή του, και του εύχομαι να συνεχίζει να προσπαθεί με τον δικό του τρόπο και πάντα να πετυχαίνει.

Στέλιος Κουκούμιαλος

ΕΚΤΙΜΗΣΗ ΑΠΟΔΟΣΗΣ ΚΑΙ ΙΔΙΟΤΗΤΕΣ ΠΟΛΙΤΙΚΩΝ ΤΥΠΟΥ KANBAN ΓΙΑ ΤΟΝ ΣΥΝΤΟΝΙΣΜΟ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ- ΑΠΟΘΕΜΑΤΩΝ ΠΟΛΛΑΠΛΩΝ ΣΤΑΔΙΩΝ

ΣΤΥΛΙΑΝΟΣ ΚΟΥΚΟΥΜΙΑΛΟΣ

Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών Βιομηχανίας, 2003

Επιβλέπων Καθηγητής: Δρ. Γεώργιος Λυμπερόπουλος, Αναπληρωτής Καθηγητής Διοίκησης
Παραγωγής

Περίληψη

Οι πολιτικές τύπου Kanban που χρησιμοποιούνται για τον συντονισμό συστημάτων παραγωγής-αποθεμάτων πολλαπλών σταδίων έχουν προσελκύσει μεγάλη προσοχή εξαιτίας της απλότητας και αποδοτικότητάς τους. Σε αυτήν την διατριβή διερευνούμε ιδιότητες πολιτικών τύπου kanban για τον συντονισμό συστημάτων παραγωγής-αποθεμάτων πολλαπλών σταδίων με και χωρίς πρότερη πληροφόρηση της ζήτησης και αναπτύσσουμε μια αναλυτική μέθοδο για την εκτίμηση της απόδοσης ενός συστήματος ελέγχου κλιμακωτών kanban.

Αρχικά, διερευνούμε αριθμητικά ανταλλαγές μεταξύ των βέλτιστων επιπέδων αποθέματος βάσης (base stock), των αριθμών των kanban, και των προγραμματισμένων χρόνων υστέρησης προμηθειών σε πολιτικές base stock και σε υβριδικές πολιτικές base stock/kanban με πρότερη πληροφόρηση της ζήτησης που χρησιμοποιούνται για τον έλεγχο συστημάτων παραγωγής-αποθεμάτων πολλαπλών σταδίων. Αναφέρουμε υπολογιστική εμπειρία από την προσομοίωση συστημάτων παραγωγής-αποθεμάτων ενός και δύο σταδίων σχετικά με τέτοιες ανταλλαγές και την σημασία τους για την διοίκηση παραγωγής.

Στην συνέχεια αναπτύσσουμε μια αναλυτική μέθοδο γενικού σκοπού για την εκτίμηση της απόδοσης ενός συστήματος ελέγχου κλιμακωτών kanban πολλαπλών σταδίων. Η βασική αρχή της προτεινόμενης προσεγγιστικής μεθόδου είναι να αποσυντεθεί το αρχικό σύστημα ελέγχου κλιμακωτών kanban σε ένα σύνολο φωλιασμένων υποσυστημάτων, όπου κάθε υποσύστημα σχετίζεται με ένα συγκεκριμένο κλιμάκιο (echelon) σταδίων. Κάθε υποσύστημα αναλύεται μεμονωμένα χρησιμοποιώντας μια τεχνική προσέγγισης που βασίζεται σε λύση που έχει την μορφή γινομένου (product form solution). Στην συνέχεια, χρησιμοποιούμε μια επαναληπτική διαδικασία για να προσδιορίσουμε τις άγνωστες παραμέτρους κάθε υποσυστήματος. Τα αριθμητικά αποτελέσματα δείχνουν ότι η μέθοδος είναι αρκετά ακριβής.

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF MECHANICAL & INDUSTRIAL ENGINEERING

Dissertation

**PERFORMANCE EVALUATION AND PROPERTIES OF KANBAN-
TYPE POLICIES FOR THE COORDINATION OF MULTI-STAGE
PRODUCTION-INVENTORY SYSTEMS**

by

STILIANOS (STELIOS) KOUKOUUMIALOS

Diploma in Mechanical & Industrial Engineering, University of Thessaly, 1997

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2003

© 2003 Stelios Koukoumialos

The approval of this Ph.D. Dissertation by the Department of Mechanical and Industrial Engineering of the School of Engineering of the University of Thessaly does not imply acceptance of the writer's opinions. (Law 5343/32, article 202 par. 2).

Approved by

- First Reader: Dr. George Liberopoulos
(Supervisor) Associate Professor, Department of Mechanical & Industrial Engineering, University of Thessaly
- Second Reader: Dr. George Tagaras
Professor, Department of Mechanical Engineering, Aristotle University of Thessaloniki
- Third Reader: Dr. Chrissoleon Papadopoulos
Professor, Department of Product and Systems Design Engineering, University of the Aegean
- Fourth Reader: Dr. Vassilis Kouikoglou
Associate Professor, Department of Production Engineering and Management, Technical University of Crete
- Fifth Reader: Dr. Michael Vidalis
Lecturer, Department of Business Administration, University of the Aegean
- Sixth Reader: Dr. Athanasios Ziliaskopoulos
Associate Professor, Department of Mechanical & Industrial Engineering, University of Thessaly
- Seventh Reader: Dr. Ioannis Bakouros
Assistant Professor, Department of Mechanical & Industrial Engineering, University of Thessaly

Acknowledgements

First and foremost, I want to thank my thesis supervisor, Associate Professor George Liberopoulos, for his valuable help and guidance throughout my doctoral work. I am also grateful to the readers of my thesis, Professors George Tagaras, Chrissoleon Papadopoulos, Vassilis Kouikoglou, Michael Vidalis, Athanasios Ziliaskopoulos and Ioannis Bakouros, for carefully reading my work and making valuable suggestions. Due thanks are extended to Professor Yves Dallery of Ecole Centrale Paris, France, who suggested to me the decomposition approach developed in Chapter 3. I thank my colleagues Nikos Gournezakis and Spiros Papaspirou for their valuable help in Matlab programming, and Andreas Dorizas, Isidoros Tsikis, Triantafilos Liapis and Petros Trifonopoulos for their assistance in the simulations of Chapter 2. I thank Christos Panoutsos for his friendly advice, and my friends Eleni Kamoutsi and Athina Oikonomou for their moral support. I also thank Maria Kateri for her understanding, especially during the last months of my effort. Most of all, I am grateful to my parents, Yiannis and Aggeliki Koukoumialos for their wholehearted love and support all these years. I dedicate this thesis to my mother and to the memory of my father, who wanted so much to see me finish it. Finally, I thank my younger brother, Giorgos Koukoumialos for his support, and I wish to him to continue trying his own way and always succeed.

Stelios Koukoumialos

PERFORMANCE EVALUATION AND PROPERTIES OF KANBAN-TYPE POLICIES FOR THE COORDINATION OF MULTI-STAGE PRODUCTION-INVENTORY SYSTEMS

STILIANOS (STELIOS) KOUKOUMLALOS

University of Thessaly, Department of Mechanical & Industrial Engineering, 2003

Major Professor: Dr. George Liberopoulos, Associate Professor of Production Management

Abstract

Kanban-type policies used for the coordination of multi-stage production-inventory systems have attracted a lot of attention because of their simplicity and effectiveness. In this thesis we investigate properties of kanban-type policies for the coordination of multi-stage production-inventory systems with and without advance demand information and we develop an analytical method for the performance evaluation of an echelon kanban control system.

First, we numerically investigate tradeoffs between optimal base stock levels, numbers of kanbans and planned supply lead times in base stock policies and hybrid base stock/kanban policies with advance demand information used for the control of multi-stage production-inventory systems. We report simulation-based computational experience regarding such tradeoffs and the managerial insights behind them for single-stage and two-stage production-inventory systems.

Then, we develop a general purpose analytical method for the performance evaluation of a multi-stage echelon kanban control system. The basic principle of the proposed approximation method is to decompose the original echelon kanban control system into a set of nested subsystems, each subsystem being associated with a particular echelon of stages. Each subsystem is analyzed in isolation using an approximation technique which is based on a product-form solution. We then use an iterative procedure to determine the unknown parameters of each subsystem. Numerical results show that the method is fairly accurate.

Contents

CHAPTER 1	INTRODUCTION.....	1
1.1	Motivation and Background.....	1
1.2	Literature Review.....	7
1.3	Thesis Organization.....	12
CHAPTER 2	TRADEOFFS BETWEEN BASE STOCK LEVELS, NUMBERS OF KANBANS AND PLANNED SUPPLY LEAD TIMES IN PRODUCTION-INVENTORY SYSTEMS WITH ADVANCE DEMAND INFORMATION.....	13
2.1	Introduction	13
2.2	Single-Stage Base Stock Policy with ADI	18
2.2.1	The Case where There Is No ADI.....	20
2.2.2	The Case where There Is ADI	21
2.3	Single-Stage Hybrid Base Stock/Kanban Policy with ADI	28
2.3.1	The Case where There Is No ADI.....	31
2.3.2	The Case where There Is ADI	36
2.4	Two-Stage Base Stock Policy with ADI	39
2.4.1	The Case where There Is No ADI.....	41
2.4.2	The Case where There Is ADI	41
2.5	Two-Stage Hybrid Base Stock/Kanban Policy with ADI	45
2.6	Conclusions	47
CHAPTER 3	AN ANALYTICAL METHOD FOR THE PERFORMANCE EVALUATION OF ECHELON KANBAN CONTROL SYSTEMS....	50
3.1	Introduction	50
3.2	Description of an Echelon Kanban Control System	52
3.3	Queuing Network Model of an Echelon Kanban Control System.....	55
3.3.1	Modeling.....	55
3.3.2	Performance Measures.....	59
3.3.3	Methodology for the Analysis of an Echelon Kanban Control System	59
3.4	Decomposition of an Echelon Kanban Control System.....	60
3.5	Analysis of Each Subsystem in Isolation.....	63
3.5.1	Determination of the Load-Dependent Service Rates in Each Subsystem.....	64
3.5.2	Analysis of the Open Queuing Systems.....	67
3.6	Analysis of the Entire Echelon Kanban Control System.....	68

3.6.1	Equations of the Decomposition	68
3.6.2	Iterative Procedure for Determining the Unknown Parameters	69
3.7	Numerical Results	72
3.7.1	Influence of Parameters.....	72
3.7.2	Optimization of Parameters.....	81
3.8	Conclusions.....	85
Appendix 3A Analysis of synchronization station O^y		86
Appendix 3B Analysis of synchronization station I^f		88
CHAPTER 4 THESIS SUMMARY		90
BIBLIOGRAPHY		93

List of Tables

Table 2-1: Parameter values for cases 1-4 of the single-stage base stock policy with ADI.....	24
Table 2-2: S^* and $C(S^*, L^*)$ versus T , for $L = L^*$, for the single-stage base stock policy with ADI.....	27
Table 2-3: S_K^* and $C(S_K^*, L^*)$ versus K for the single-stage hybrid base stock/kanban policy with no ADI.....	32
Table 2-4: S_K^* and $C(S_K^*, L^*)$ versus T and K , for $L = L^*$, for the single-stage hybrid base stock/kanban policy with ADI.....	37
Table 2-5: Parameter values for the case of the two-stage base stock policy with ADI.....	42
Table 2-6: S_1^* , S_2^* and $C(S_1^*, S_2^*)$ versus T , for $L_1 = L_1^*$ and $L_2 = L_2^*$, for the two-stage base stock policy with ADI.....	42
Table 2-7: S_1^* , S_2^* and $C(S_1^*, S_2^*)$ versus T , L_1 and L_2 for the two-stage base stock policy with ADI.....	43
Table 2-8: S_1^* , S_2^* and $C(S_1^*, S_2^*)$ versus T , K_1 and K_2 , for $L_1 = L_1^*$ and $L_2 = L_2^*$, for the two-stage hybrid base stock/kanban policy with ADI.....	48
Table 3-1: Production capacity of the saturated echelon kanban control system (Example 1).	74
Table 3-2: Production capacity of the saturated conventional kanban system (Example 1)....	75
Table 3-3: Average number of backordered demands, mean waiting time of a backordered demand and proportion of backordered demands for the echelon kanban control system (Example 1).	76
Table 3-4: Average number of backordered demands, mean waiting time of a backordered demand and proportion of backordered demands for the conventional kanban control system (Example 1).	77
Table 3-5: Average work in process (WIP) and average number of finished parts (FP) in each stage for the echelon kanban control system (Example 1).	78
Table 3-6: Average work in process (WIP) and average number of finished products (FP) in each stage for the conventional kanban system (Example 1).	79
Table 3-7: Production capacity of the echelon kanban control system (Example 2).	81
Table 3-8: Production capacity of the conventional kanban control system (Example 2).	81
Table 3-9: Optimal configuration and associated costs for different values of h_1, \dots, h_5 and $\lambda_D = 0.5$ for the echelon kanban control system.	84
Table 3-10: Optimal configuration and associated costs for different values of h_1, \dots, h_5 and $\lambda_D = 0.5$ for the CONWIP system.	85

List of Figures

Figure 2-1: Single-stage base stock policy with ADI.	19
Figure 2-2: S^* versus T , for $L = L^*$, for the single-stage base stock policy with ADI.	27
Figure 2-3: Single-stage hybrid base stock/kanban policy with ADI.	30
Figure 2-4: S_k^* and $C(S_k^*, L^*)$ versus K for the single-stage hybrid base stock/kanban policy with no ADI.	33
Figure 2-5: Two-stage base stock policy with ADI.	40
Figure 2-6: S_1^* , S_2^* and $C(S_1^*, S_2^*)$ versus T , for $L_1 = L_1^*$ and $L_2 = L_2^*$, for the two-stage base stock policy with ADI.	43
Figure 2-7: Two-stage hybrid base stock/kanban policy with ADI.	46
Figure 3-1: Example of a serial production system.	53
Figure 3-2: Production system of Figure 3-1 decomposed into three stages in series.	53
Figure 3-3: Queuing network model of the echelon kanban control system of Figure 3-2.	56
Figure 3-4: Queuing network model of the saturated echelon kanban control system associated with the queuing network of Figure 3-3.	58
Figure 3-5: Illustration of the decomposition of a 3-stage echelon kanban control system. ...	61
Figure 3-6: Proportion of backordered demands versus the average arrival rate of demands for different values of the squared coefficient of variation (Example 2).	80
Figure 3-7: Markov chain describing the state (n_o^N, n_p) of synchronization station O^N	87
Figure 3-8: Markov chain describing the state (n_l^i, n_u^i) of queuing network I^i	89

Chapter 1 Introduction

In this chapter, we provide some background information which supports the motivation behind this thesis. We also review the literature that is related to the thesis, and we give a brief description of the two main parts of the thesis, which occupy Chapters 2 and 3, respectively.

1.1 Motivation and Background

Production-inventory systems consist of workstations/transport equipment and buffers where parts are processed/transferred and stored repeatedly until final products are created and delivered to customers. The effective production control of any production-inventory system, i.e. the management of the total flow of goods through the system, from the acquisition of raw parts to the delivery of final products to customers, is paramount to the competitiveness of the system. Production control is an optimization problem that typically addresses the question of when and how much to produce in order to achieve a satisfactory customer service level (measured by how quickly customer demands are satisfied), while keeping low in process inventories. Difficulties in production control arise because of queuing delays due to variability in production capacity and demand for final or intermediate products. Things may get even more complicated when it is possible to obtain end-item *advance demand information* – henceforth referred to as ADI – in the form of actual orders, commitments, forecasts, etc.

A practical approach to tackling production control is to restrict the search for a production control policy to a class of simple, sub-optimal policies that are easy to implement and try to determine the optimal policy within this class. Much of the research effort in this area has focused on developing and evaluating simple production control policies that depend on a small number of parameters and have often emerged from actual industrial practice. In many such policies encountered in the literature, production control is applied at a selected number of points in the manufacturing system. This is done by functionally aggregating several production activities into different production stages and then coordinating the release of parts into each stage with the arrival of customer demands for final products.

There are many reasons for wanting to aggregate production activities into stages and control the material flow in between stages. First, in most manufacturing systems production activities are naturally grouped into well identifiable production stages. In practice, these stages operate independently from one another and what couples them is the release of parts from one stage to the next. Second, when dealing with multi-product systems, set ups to change from one product to another are often performed on whole sub-systems of machines (e.g., on a production line) rather than on individual machines. Controlling the production of each individual machine may, therefore, not be appropriate in such cases. Finally, having fewer points to control makes the production control problem simpler and the implementation of a production control policy easier.

The simplest production-inventory control policy in case there is no ADI is a so-called base stock policy (e.g. see [76]). Base stock policies were originally developed for non-capacitated inventory systems. In a base stock policy, production orders to release parts into each stage are issued as soon as customer demand arrives to the system. The advantage of this policy is that it responds rapidly to demand. Its disadvantage is that it provides a very loose

coordination between stages and that it does not guarantee any limit on the number of parts that may enter the system, since any time a customer demand arrives to the system authorizes the release of a new part into the first stage.

A policy that has attracted considerable attention and has particular appeal in JIT capacitated production environments is the kanban policy, according to which, a production order to release a part into a stage is issued only when a finished part of that stage is consumed by the downstream stage (e.g. see [10]). The advantage of this policy is that it limits the number of parts that may enter each stage. A disadvantage is that the system may not immediately respond to customer demands, since a customer demand may not immediately be transferred to all stages upon its arrival to the system. Another drawback of the kanban policy is that it provides a very tight coordination between stages since the transfer of demands, production authorizations, and parts is completely coupled. The classical kanban policy was originally developed and implemented in the Toyota production line in the mid-seventies. Since then, several variants and extensions of the kanban policy have been developed. We refer to such policies as kanban-type policies.

Kanban-type policies used for the coordination of multi-stage production-inventory systems have attracted a lot of attention because of their simplicity and effectiveness. In this thesis we investigate properties of base stock and kanban-type policies for the coordination of multi-stage production-inventory systems with and without ADI and we develop an analytical method for the performance evaluation of an echelon kanban control system.

When studying a production control policy used for the coordination of multi-stage production-inventory systems, one is generally interested in performing the following tasks:

1. **Describe the dynamics of the policy.** In this thesis we employ the modeling framework used in [50], [51], [53], which is based on a queuing network representation with synchronization stations, to describe the dynamics of the policies that we study. We use this framework because we find that it describes in a clear and exact manner the operation of these policies.
2. **Uncover properties of the dynamics of the policy.** Examples of properties of the dynamics of the policy are invariants and bounds on the contents of various queues in the queuing network representation of the policy, as well as monotonicity properties of a performance measure (e.g. throughput) with respect to the design parameters (e.g. number of kanbans) of the policy. Examples of such properties for the extended kanban control system can be found in [23].
3. **Develop tools for evaluating the performance of the policy.** The performance measures of a production control policy that are of primary interest are the average number of work in process and finished parts and the customer service level. The customer service level is usually included in the objective function as a cost of backorders or is treated as a constraint (e.g. on the minimum percentage or average number of demands that are met from stock) that must be satisfied. The first approach is taken in Chapter 2, whereas the second approach is taken in Chapter 3. Traditionally, there are three types of tools for evaluating the performance measures of a policy: exact analysis, approximate analysis, and simulation.

Exact analytical solutions can be obtained for a class of networks known as separable for which the steady-state joint probability has a product form solution, and in general only for small and simple systems, e.g., for closed queuing networks with population constraints an exact analytical solution can only be obtained by analyzing the underlying

Markov chains using numerical techniques [70] but this is feasible only for networks of small sizes.

Unfortunately, the assumption of separable networks is often too restrictive for the analysis and modeling of real systems. Therefore, much work has been devoted to the approximate analysis of more complex systems and non-separable networks. Generally, most of the approximate methods are based on decomposition. In open networks, the network is decomposed into a set of subsystems, which are analyzed in isolation. In particular, this technique has been applied with success to open networks with general service time distributions. For closed queuing networks, two main approaches have been followed in order to derive approximation methods. The first one is based on heuristic extensions of the mean value analysis (MVA) algorithm, and the second one uses two main techniques, the aggregation technique and Marie's method, in order to approximate the performance of the original network by that of an equivalent product-form network. The second approach is very general since it applied to queuing networks with complex behaviors such as blocking, non-exponential service times, population constraints, and fork/join mechanisms [5]. In this thesis, the second approach and especially Marie's method is used in Chapter 3.

When the development of exact or approximate analytical methods is not possible, simulation may be the only solution. Simulation refers to a collection of methods for studying a wide variety of models of real-world systems by numerical evaluation using software design to imitate the system's operations or characteristics, often over time. From a practical view point, simulation is the process of designing and creating a computerized model of a real or proposed system for the purpose of conducting numerical experiments to give us a better understanding of the behavior of that system for a given set

of conditions. Although it can be used to study simple systems, the real power of this technique is fully realized when we use it to study complex systems. However, simulation isn't quite paradise, either. Because many real systems are affected by uncontrollable and random inputs, many simulation models involve random, or stochastic, input components, causing their output to be random too. A solution to that problem comes from longer time frame since if the time frame becomes longer most results averaged over the run will tend to settle down and becomes less variable. Discrete-event simulation is used in Chapter 2 as well as in Chapter 3.

4. **Use the performance evaluation tools to optimize the design parameters of the policy.** Once a tool for evaluation the performance of the policy for a given set of design parameters has been developed, an optimization algorithm can be developed to optimize these parameters. In most cases, a gradient search method combined with intuition about where to search for the optimal parameters will do the job. In some cases, constraints on the design parameter values must be taken into account. A problem that often arises in the policies that we examine is that the performance is very insensitive to some of the parameters especially around their optimal values.
5. **Uncover structural properties of the optimal parameters of the policy.** The optimal parameters of the policy may have certain structural properties which are not obvious and are difficult to uncover unless a number of numerical studies have been performed. Such properties may involve tradeoffs between different design parameters, which can provide very useful managerial insights. Most of Chapter 2 is devoted to uncovering such tradeoffs in production-inventory systems with ADI.
6. **Compare the dynamics and the optimal parameters and measures of the policy against these of other policies.** Ultimately, we are interested in picking a control policy

which is simple, robust and performs well compared to other policies. For this reason we should be able to compare different policies. In Chapter 3 we compare an echelon kanban control policy against a conventional kanban control policy.

1.2 Literature Review

There exist several studies that analyze and compare different production-inventory control policies in the case where there is no ADI. Spearman [67] uses stochastic ordering arguments to compare customer service in pull production-inventory systems operating under kanban and CONWIP policies and presents some comparative numerical results for a single-stage system consisting of three stations in tandem. Veach and Wein [73] employ dynamic programming to compute the optimal control policy for a make-to-stock production-inventory system consisting of two stations in tandem and compare it to the optimal base stock, kanban, and fixed-buffer policies. One of their results is that base stock policies are never optimal for such a system. Karaesmen and Dallery [43] follow the same approach to analyze a similar two-station system operating under single-stage and two-stage base stock, kanban, and generalized kanban policies. Bonvik et al. [11] employ simulation to compare kanban, minimal blocking, base stock, and CONWIP policies with a hybrid kanban-CONWIP policy in a four-machine tandem production line. Rubio and Wein [63] employ queuing theory to analyze production-inventory systems that operate under a classical base stock policy. Frein et al. [30] model production-inventory systems that operate under a generalized kanban policy as open queuing networks with restricted capacity and employ approximate analysis to evaluate their performance. Duri et al. [29] compare base stock, kanban, and generalized kanban policies for production-inventory systems consisting of one, two, three, and four-stage stations, using approximation techniques. Zipkin [76] (sec. 8.8.2) compares a two-stage

production-inventory system, where each stage consists of a single server, operating under base stock, kanban, and generalized kanban policies.

The issue of ADI has attracted quite a bit of attention in the context of uncapacitated pure inventory systems (e.g. see [12], [19], [27], [31], [40], [57], [58], [72]). Yet, there exist few studies that investigate the benefits of ADI in capacitated production-inventory systems. Güllü [38] and Toktay and Wein [71] study production-inventory systems with ADI in the form of forecasts, using a model of forecast evolution developed by Graves et al. [36], [37] and Heath and Jackson [41]. Gavirneni and Tayur [32] consider a stochastic, capacitated production-inventory system where customers use a so-called target reverting policy, which gives rise to a non-stationary demand process as viewed by their supplier. They find that for such a system, an order up-to policy is optimal. Gillbert and Ballou [34] investigate the capacity planning problem of a make-to-order supplier that can receive advance demand commitments through a pricing policy. Benjaafar and Kim [9] investigate ADI for a make-to-stock queue in the context of demand variability. Buzacott and Shanthikumar [16] (sec. 4.5.2), [17] analyze a single-stage, single-server make-to-stock queue for which they explicitly characterize the tradeoff between safety stock and safety lead time. Karaesmen et al. [45] further interpret this tradeoff and analyze the value of ADI. Karaesmen et al. [44] employ dynamic programming to compute the optimal control policy for a discrete time make-to-stock queue with ADI in the form of constant demand lead times, and compare it to the optimal base stock policy with a planned supply lead time parameter. To the best of our knowledge, there have been no studies on hybrid base stock/kanban policies with ADI to date.

Some of the most widely-used methods that have been developed for the analysis of kanban-type production-inventory policies are approximation methods for the performance evaluation of queuing networks with an emphasis on closed queuing networks. Exact

analytical solutions exist for a class of queuing networks known as separable. The steady-state joint probabilities for these networks have a product-form solution. Jackson [42] showed that the steady-state joint probability of an open queuing network with Poisson arrivals, exponential service times, probabilistic routing and first-come-first-served (FCFS) service disciplines has a product-form solution. Each station of such a network can be analyzed in isolation as an M/M/1 queue. For the case of closed queuing networks of Jackson type, Gordon and Newell [35] proved that the product-form solution also holds. The BCMP theorem [4] summarizes extensions of these results to incorporate alternative service disciplines and several classes of customers. The performance parameters of separable networks can be obtained using efficient algorithms such as the mean value analysis (MVA) algorithm [62] and the convolution algorithm [18].

Unfortunately, since the class of queuing networks for which an exact solution is known (separable networks) appears too restrictive for modeling and analyzing real systems, much work has been devoted to the approximate analysis of non-separable networks. By non-separable networks we mean general queuing networks, especially queuing networks with general service time distributions and FCFS disciplines. Most of the approximation methods are based on decomposition. In open networks, the queuing network is decomposed into a set of GI/GI/1 queues that are analyzed in isolation [47], [48], [74]. For closed queuing networks, in which the total number of customers in the network is a constant, an exact solution can be obtained by analyzing the underlying Markov chains using numerical techniques [70] but this is feasible only for networks of small sizes. For networks of realistic size, approximation methods have been developed. These approximation methods are for the most part based on two approaches. The first approach relies on heuristic extensions of the MVA algorithm. In order to handle non-BCMP stations, especially stations with priority disciplines and FCFS

exponential stations with class-dependent service rates, a variety of MVA heuristic extensions have been proposed [65]. The second approach approximates the performance of the original network by that of an equivalent product-form network. Among the different approximation techniques that have been proposed, two have proved to be of high interest, namely Marie's method [54] and the aggregation technique [1], [13], [49] and [64]. The common idea of these techniques is to approximate the performance of the original network by that of an equivalent product-form queuing network with load-dependent service rates. The difference between the two methods is in the way that these service rates are estimated. A third technique is to aggregate several states of the underlying Markov chain and adjust some service rates using Norton's Theorem for closed queuing networks to obtain a product-form solution [25].

Approximation methods, with extensions in some cases, have been applied to a variety of queuing networks. Dallery [21] considers an open queuing network with restricted capacity and general service time distributions. In that network, the total number of customers cannot exceed a given number, called the capacity of the network. If the arrival of a customer finds the network full, it is forced to wait in an external queue. A different approximation technique is used for the analysis of this system. In particular, the roles of the jobs and the resources, i.e. the empty slots in the network, are exchanged and the resulting model is a closed queuing network whose population is equal to the capacity of the open network. An extension of Marie's method is developed to analyze this closed queuing network, and the performance parameters of the open model are derived from those of the equivalent closed queuing network. Di Mascolo, Frein and Dallery [28] develop a general-purpose analytical method for the performance evaluation of kanban-controlled, single-part-type, serial, multistage production systems. The approximation method that they propose decomposes the original system into a set of subsystems, each subsystem being associated with a particular stage. Each

subsystem is then analyzed in isolation using a product-form approximation technique. The performance parameters of each subsystem are determined using an iterative procedure. In a later work [30], the same authors apply an extension of this method to evaluate the performance of generalized kanban control systems.

Baynat and Dallery [6] consider closed queuing networks with subnetworks having population constraints and general service time distributions. A population constraint means that the total number of customers that are simultaneously present in a subnetwork is constant and equal to a given number. Each subnetwork consists of a subset of stations of the original network. If the subnetwork is full, the arrival of a customer has to wait in an external FCFS queue. They analyze such networks by applying both the aggregation technique and Marie's method, and they also provide a comparison between those two techniques. In another work, Baynat and Dallery [8] consider general closed queuing networks with several classes of customers and propose a new method for obtaining approximate solutions for such networks. The concept of this method is to associate with each class of customers a single-class closed queuing network with load-dependent exponential service stations. The interaction of customers that belong to other classes is taken into account in the estimation of the load-dependent service rates, which are obtained by analyzing each station in isolation under the assumption that the arrival process of each class is a state-dependent Markovian process. In order to reduce the complexity of the analysis, they also propose a class aggregation technique. Finally, Baynat and Dallery [7] propose an approximation method based on the use of a product-form approximation technique in order to analyze queuing networks with synchronization mechanisms and multiple classes of customers.

More generally, queueing network models with finite capacity queues and blocking have been introduced and applied as more realistic models of systems with finite capacity

resources and population constraints. Three textbooks that focus on the analysis of such networks are [3], [60] and [61]. Finally, different decomposition-based methodologies applied to manufacturing systems can be found in the textbooks [1], [16] and [33] as well as in the survey papers [22] and [59].

1.3 Thesis Organization

The remainder of this thesis is organized into two main parts which occupy Chapters 2 and 3, respectively. More specifically:

In Chapter 2, we numerically investigate tradeoffs between optimal base stock levels, numbers of kanbans and planned supply lead times in base stock policies and hybrid base stock/kanban policies with ADI used for the control of multi-stage production-inventory systems. Simulation-based computational experience regarding such tradeoffs and the managerial insights behind them are reported for single-stage and two-stage production-inventory systems.

In Chapter 3, we develop a general purpose analytical method for the performance evaluation of a multi-stage echelon kanban control system. The system is modeled as a queuing network with synchronization stations. The basic principle of the proposed approximation method is to decompose the original echelon kanban control system into a set of nested subsystems, each subsystem being associated with a particular echelon of stages. Each subsystem is analyzed in isolation using a product-form approximation technique. An iterative procedure is then used to determine the unknown parameters of each subsystem. Numerical results show that the method is fairly accurate.

Finally, a summary of the thesis is presented in Chapter 4.

Chapter 2 Tradeoffs between Base Stock Levels, Numbers of Kanbans and Planned Supply Lead Times in Production-Inventory Systems with Advance Demand Information

2.1 Introduction

Recent developments in information technology and the emphasis on supply chain system integration have significantly reduced the cost of obtaining end-item advance demand information (ADI) in the form of actual orders, commitments, forecasts, etc., and diffusing it among all stages of the system. This has created opportunities for developing effective production-inventory control policies that exploit such information. The implementation of such policies may result in significant cost savings throughout the entire system through inventory reductions as well as improvements in customer service [12], [17], [19], [27], [31], [34], [38], [40], [45], [58], [72].

In this chapter we investigate policies that use ADI for production-inventory control of a multi-stage serial system that produces a single type of parts in a make-to-stock mode. We make the following specific assumptions. Every stage in the system consists of a facility where parts are processed, and an output store where finished parts are stored. Parts in the facility are referred to as *work-in-process* (WIP), and parts in the output store are referred to

as *finished goods* (FG). Finished goods of the last stage are referred to as *end-items*. There is an infinite supply of raw parts feeding the first stage. Customer demands arrive randomly for one end-item at a time, with a constant *demand lead time* in advance of their due dates. Once a customer demand arrives, it cannot be cancelled, i.e. the ADI is assumed to be perfect. Demands that cannot be satisfied on their due dates are backordered and are referred to as *backordered demands* (BD). The arrival of a customer demand for an end-item triggers the issuing of a production order to replenish FG inventory at every stage. FG inventory levels are followed continuously at all stages, and replenishment production orders may be issued at any time. There is no setup cost or setup time for issuing a production order and no limit on the number of orders that can be placed per unit time. Under the above assumptions, there is no incentive to replenish FG inventory by anything other than a continuous review, one-for-one replenishment policy. The model described above is simple but it captures some of the basic elements of the operation of a serial capacitated system.

When there is no ADI, demand due dates coincide with demand arrival times. In this case, the replenishment production orders at every stage, which are triggered by the arrival of customer demands, may be issued only at (or after) the demand due dates. The simplest production-inventory control policy in case there is no ADI is a *base stock* policy. Base stock policies were originally developed for non-capacitated inventory systems. In a base stock policy, production orders to replenish FG inventory at each stage are issued as soon as customer demand arrives to the system. A policy that has attracted considerable attention and has particular appeal in a JIT capacitated production environment is the kanban policy. In the case of a single-stage system, a kanban policy is equivalent to a make-to-stock CONWIP policy [68]. In a kanban policy, a production order to replenish FG inventory at a stage is issued only when a finished part of that stage is consumed by the downstream stage. Base

stock and kanban policies may be combined to form more sophisticated hybrid base stock/kanban policies such as generalized kanban policies [15], [16], [75], [76] or extended kanban policies [23]. In a generalized kanban policy, a production order to replenish FG inventory at a stage is issued only when the inventory in that stage is below a given inventory-cap level. In an extended kanban policy, production orders to replenish FG inventory at each stage are issued as soon as customer demand arrives to the system but are authorized to go through a stage only when the inventory in that stage is below a given inventory-cap level. A detailed description of these and other similar policies can be found in [50], [51] and [53]. A brief literature review on studies that analyze and compare different production-inventory control policies in the case where there is no ADI is given in Section 1.2 of Chapter 1.

When there is ADI, the production orders to replenish FG inventory at every stage, which are triggered by the arrival of a customer demand, may be issued before the due date of the demand. Base stock and hybrid base stock/kanban policies can be easily modified to take advantage of ADI by offsetting requirements due dates by *stage planned supply lead times* to determine the issue times of production replenishment orders at every stage, as is done in the time-phasing step of the MRP procedure. The planned supply lead time of a stage is a fixed parameter of the control policy which, in an MRP system, is typically set so as to guarantee that the actual flow time (a random variable) of a part through the facility of the stage falls within the planned supply lead time most of the time (e.g. 95% of the time) [44]. A kanban policy can not exploit ADI, because in a kanban policy a production order is issued after a part in FG inventory is consumed and therefore at (or after) the due date of the demand that triggered it. When ADI is available, it is therefore reasonable to consider only base stock and hybrid base stock/kanban policies and not pure kanban policies, where by hybrid base stock/kanban policies we mean different variants of combinations of base stock and kanban

policies. Hybrid base stock/kanban policies are of particular interest because they fuse together reorder point inventory control policies, JIT, and MRP, three widely practiced approaches for controlling the flow of material in multi-stage production-inventory systems.

The aim of our investigation is to reveal tradeoffs between optimal base stock levels, numbers of kanbans, and planned supply lead times in multi-stage base stock and hybrid base stock/kanban policies with ADI. Some of the more specific issues that we will try to study are the following.

In both base stock and hybrid base stock/kanban policies with ADI, the base stock level of FG inventory represents parts that have been produced before any demands have arrived to the system to protect the system against possible stockouts. Intuitively, there is a tradeoff between the demand lead time and the optimal base stock levels of FG inventory. Namely, the larger the demand lead time, the smaller the optimal base stock level. But is there a structure to this tradeoff? More specifically, do the optimal base stock levels decrease at a constant rate or at a diminishing rate and until which point as the demand lead time increases? Do the optimal base stock levels at different stages all decrease at once until they drop to zero or to some constant level, or do they decrease one after the other in a certain order as the demand lead time increases? If the latter is true, is it more beneficial to first decrease the base stock level at upstream stages, where FG inventory is usually less expensive to hold but also less important, or at downstream stages, where FG inventory is usually more expensive to hold but also more important with respect to customer service?

The planned supply lead times are control parameters that determine how much (if any) to delay the issuing of production replenishment orders triggered by the arrival of customer demands. Intuitively, if the demand lead time is short, the issuing of production replenishment orders should not be delayed, whereas if the demand lead time is long, the

issuing of production replenishment orders should be delayed. But what is the maximum critical demand lead time below which the issuing of production replenishment orders should not be delayed? Does it make sense to delay the issuing of production replenishment orders and at the same time have positive base stock levels at some stages?

In hybrid base stock/kanban policies the number of kanbans at each stage represents an inventory cap and determines the production capacity of the stage. Intuitively, there is a tradeoff between the number of kanbans and the base stock level at every stage. Specifically, the smaller the number of kanbans, the smaller the production capacity, and the larger the production replenishment time and consequently the base stock level of FG inventory. But what is the optimal number of kanbans and therefore the optimal base stock level, and how are they affected by the demand lead time? More specifically, as the demand lead time increases, should the optimal number of kanbans be decreased, and if so, by how much?

Since exact analytical tools for evaluating the performance of multi-stage base stock and hybrid base stock/kanban policies with ADI are limited and approximation-based analytical tools yield inaccurate results, which may be misleading when trying to reveal tradeoffs between parameters, we use simulation and brute force optimization to investigate such tradeoffs and report the results of this investigation for single-stage and two-stage systems. The main contribution of this chapter is the managerial insights that these results bring to light.

The rest of the chapter is organized as follows. In Sections 2.2 and 2.3 we numerically investigate single-stage base stock and hybrid base stock/kanban policies, respectively. In Sections 2.4 and 2.5 we numerically investigate two-stage base stock and hybrid base stock/kanban policies, respectively. Finally in Section 2.6 we draw conclusions.

2.2 Single-Stage Base Stock Policy with ADI

In this section we consider a single-stage base stock policy with ADI, which is similar to that considered in [44]. Customer demands arrive for one end-item at a time according to a Poisson process with rate λ , with a constant demand lead time, T , in advance of their due dates. The arrival of every customer demand eventually triggers the consumption of an end-item from FG inventory and the issuing of a production order to the facility of the one and only stage to replenish FG inventory. More specifically, the consumption of an end-item from FG inventory is triggered T time units after the arrival time of the demand. If no end-items are available at that time, the demand is backordered. The control policy depends on two design parameters, the base stock level of end-items in FG inventory, denoted by S , and the stage planned supply lead time, denoted by L . The base stock level S has the same meaning as in a classical base stock policy. The only difference is that in the presence of ADI, the inventory position can exceed the base stock level. In fact, for this reason, Hariharan and Zipkin [40] and Chen [19] use the term “order base stock” instead of “base stock” to describe the target FG inventory. The stage planned supply lead time L has the same meaning as the fixed lead time parameter in an MRP system. Initially, the system starts with a base stock of S end-items in FG inventory. The time of issuing the replenishment production order is determined by offsetting the demand due date by the stage planned supply lead time, L , as is done in the time-phasing step of the MRP procedure. This means that the order is issued immediately with no delay, if $L \geq T$ (in this case, the order is already late), or with a delay equal to $T - L$ with respect to the demand arrival time, if $L < T$. In other words, the delay in issuing an order is equal to $\max[0, T - L]$. When the order is issued, a new part is immediately released into the facility. If there is no ADI, i.e. if $T = 0$, both the consumption of an end-item from FG inventory and the replenishment production order are triggered at the demand arrival time,

and the resulting policy is a classical base stock policy. A queuing network model of a base stock policy with ADI is shown in Figure 2-1.

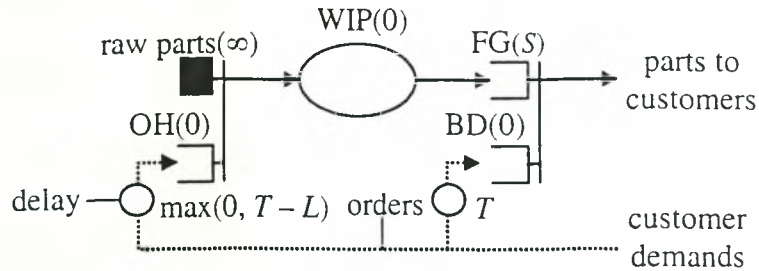


Figure 2-1: Single-stage base stock policy with ADI.

The symbolism used in Figure 2-1 (and all other similar figures that follow in the rest of the chapter) is the same as that used in [23], [29], [44], [30], [50], [51], [53], and has the following interpretation. The oval represents the facility, and the circles represent time delays. The queues followed by vertical bars represent *synchronization stations* linking the queues. A synchronization station is a server with instant service time that “fires” (serves customers) as soon as there is at least one customer in each of the queues that it synchronizes. Queues are labeled according to their content, and their initial value is indicated inside parentheses. Queue OH stands for replenishment *orders on hold*. In the single-stage base stock policy, this queue is always equal to zero, because of the underlying assumption that there is an infinite number of raw parts. Recall that BD stands for *backordered demands*.

We consider a classical optimization problem where the objective is to find the values of S and L that minimize the long run expected average cost of holding and backordering inventory,

$$C(S,L) = hE[\text{WIP} + \text{FG}(S, L)] + bE[\text{BD}(S, L)], \quad (2.1)$$

where h is the unit cost of holding WIP + FG inventory per unit time and b is the unit cost of backordering FG inventory per unit time. Here, we are more interested in the effect of ADI on inventory holding and backordering costs than on the cost of ADI itself, so we assume that there is no cost of obtaining ADI. The effect of buying ADI is considered in [46]. It is not difficult to see that control parameters S and L affect the expected average FG and BD only and not the expected average WIP. We explicitly express these dependencies in the cost function (2.1). In what follows, we will study the above optimization problem, first for the case where there is no ADI and then for the case where there is ADI.

2.2.1 The Case where There Is No ADI

If there is no ADI, i.e. if $T = 0$, the planned supply lead time parameter L is irrelevant, because a replenishment order is always issued at the time of a customer demand arrival. After some algebraic manipulations, the long run expected average cost (2.1) can then be expressed as a function solely of S as follows:

$$C(S) = (h + b) \left\{ E[\text{WIP}] - \sum_{n=0}^S nP(\text{WIP} = n) + S[P(\text{WIP} \leq S) - b/(b + h)] \right\}. \quad (2.2)$$

Moreover, the optimal base stock level, S^* , is given by the well-known critical fraction rule of the newsvendor problem, i.e. it is the smallest integer that satisfies (see [63])

$$P(\text{WIP} \leq S^*) \geq b/(b + h). \quad (2.3)$$

If the facility consists of a single-server station with exponential service rate μ , the long run expected average cost (excluding the cost of WIP, which is equal to $h\rho/(1-\rho)$ and is therefore independent of the design parameters S and L) is given by (see [16] (sec. 4.3.1) and [63])

$$C(S) = h[S - \rho(1 - \rho^S)/(1 - \rho)] + b[\rho^{S+1}/(1 - \rho)],$$

where $\rho = \lambda/\mu$, and $S^* = \lfloor \hat{S} \rfloor$, where $\hat{S} = \ln[h/(h + b)]/\ln \rho$ and $\lfloor x \rfloor$ denotes the “floor” of x (i.e. the largest integer which is smaller than or equal to x).

If the facility consists of a Jackson network of servers, S^* satisfies a non-closed-form expression that can be solved numerically. For instance, in the case of a balanced Jackson network consisting of M identical single-server stations, each server having an exponential service rate μ , S^* is the smallest integer that satisfies (2.3), where the WIP has a negative binomial steady state distribution given by (see [63])

$$P(\text{WIP} = n) = \binom{M + n - 1}{n} (1 - \rho)^M \rho^n, \quad (2.4)$$

where $\rho = \lambda/\mu$.

2.2.2 The Case where There Is ADI

If there is ADI, i.e. if $T > 0$, there is a time lag between issuing an order and demanding an end-item from FG inventory. This time lag is equal to $T - \max[0, T - L] = \min[T, L]$, which implies that any system with demand lead time $T > L$ behaves exactly like a system with demand lead time $T = L$.

The case of a single-server exponential station

If the facility consists of a single-server station with exponential service rate μ , the long run expected average cost (excluding the cost of WIP, which is equal to $h\rho/(1 - \rho)$) is given by (see [16] (sec. 4.5.2), [17], and [45])

$$C(S, L) = h[S + \lambda \min(L, T) - \rho/(1 - \rho)] + (h + b)[\rho^{S+1}/(1 - \rho)]e^{-\mu(1 - \rho)\min(L, T)},$$

where $\rho = \lambda/\mu$. One can then optimize $C(S,L)$ with respect to parameters S and L to gain insight into the behavior of the system under the optimal parameters. Specifically, it can be shown [45] that for a fixed L , the optimal based stock level, $S^*(L)$, is given by

$$S^*(L) = \begin{cases} \lfloor \hat{S}(L) \rfloor & \text{if } L \leq L^*, \\ 0 & \text{if } L \geq L^*, \end{cases}$$

where

$$\hat{S}(L) = \ln[(h+b)/h]/\ln \rho - [(\mu - \lambda)/\ln \rho]L$$

and the optimal planned supply lead time, L^* , is given by

$$L^* = \ln[(h+b)/h]/(\mu - \lambda).$$

The overall optimal base stock, $S^* = S^*(L^*)$, is then equal to the integer $\lfloor \hat{S} \rfloor$, where

$$\hat{S} = \max\{0, (\ln[(h+b)/h]/\ln \rho) - [(\mu - \lambda)/\ln \rho]T\}.$$

The above analysis implies that L^* is independent of T and is equal to $cE[W]$, where W is the waiting (or flow) time of a part in the facility if the system were operated in make-to-order mode (since $E[W] = 1/(\mu - \lambda)$), and c is a factor equal to $\ln[(h+b)/h]$. \hat{S} and hence S^* , on the other hand, are functions of T . More specifically, \hat{S} decreases linearly with T and reaches zero at $T = L^*$. Thus, for demand lead times T such that $T < L^*$, $\hat{S} > 0$ and production orders are issued upon the arrival of demands with no delay. For demand lead times T such that $T > L^*$, however, $\hat{S} = 0$ and production orders are issued upon the arrival of demands with a delay of $T - L^*$. This means that for $T > L^*$, the system *switches its operation from a make-to-stock mode into a make-to-order mode*. The minimum long run expected average cost $C(S^*, L^*)$ decreases with T and attains its minimum value at $T = L^*$. Since L^* is the smallest value of T

for which $\hat{S} = 0$, and $S^* = \lfloor \hat{S} \rfloor$, it follows that the smallest value of T for which $S^* = 0$, is just below L^* .

The case of a Jackson network of servers

If the facility consists of a Jackson network of servers, there are no general analytical results available for the optimal parameter values. Intuitively, we would expect that as T increases, the optimal base stock level should decrease, as is the case with the single-server station. The question is how exactly does it decrease? Does it decrease linearly until it drops to zero, as in the case of a single-server station, or does it decrease in some sort of non-linear way (e.g. a diminishing way)? What is the smallest value of T , for which the optimal base stock level becomes zero, switching the system operation from a make-to-stock mode into a make-to-order mode? Is it equal to the average flow time of a part through the facility multiplied by the factor $c = \ln[(h + b)/h]$, as in the case of a single-server station?

The only general analytical result related to the above question is Proposition 1 in [45] which states that for any supply system satisfying the following assumption, if the system operates in a make-to-order mode (i.e. with zero base stock level) and $T \geq L^*$, then the optimal planned supply lead time L^* is the smallest real number L that satisfies

$$P(W \leq L^*) \geq b/(b + h), \tag{2.5}$$

where W is the order replenishment time, i.e. the waiting or flow time of a part in the facility.

Assumption 1: All replenishment orders enter the supply system one at the time, remain in the system until they are fulfilled (there is no blocking, balking or renegeing), leave one at a time in the order of arrival (FIFO) and do not affect the flow time of previous replenishment orders (lack of anticipation).

Assumption 1 concerns systems similar to those considered by Haji and Newell [39] in a pioneering paper in which they address the issue of relating the queue length distribution and the waiting time distribution in a queuing system, when the discipline of the system is first-in-first-out (FIFO), and prove a general distributional Little's law. Notice the similarity between expressions (2.3) and (2.5). These two expressions demonstrate explicitly the interchangeability of safety stock and safety time.

The implication of the above result is that if the system in Figure 2-1 satisfies Assumption 1, then when $T \geq L^*$, where L^* is given by (2.5), the system switches from make-to-stock mode into a make-to-order mode with optimal planned supply lead time L^* .

In summary, when $T = 0$, S^* is given by (2.3), and when $T \geq L^*$, $S^* = 0$. A question that remains unanswered is what happens when $0 < T < L^*$? To shed some light into this issue, we numerically investigated a particular but representative instance of the system, in which the facility consists of a Jackson network of $M = 4$ identical single-server stations in series, each server having a mean service time $1/\mu$. Thus, the system instance that we considered has the FIFO property. For this instance, we considered four sets of parameter values shown in Table 2-1. In cases 1 and 2, the service time distribution of each machine is exponential, whereas in cases 3 and 4, it is Erlang with two phases. The parameters values, for the four cases are as follows.

Case	$1/\lambda$	Service time distribution	$1/\mu$	$\rho = \lambda/\mu$	h	b
1	1.25	exponential	1.0	0.8	5	1
2	1.1	exponential	1.0	0.90909...	1	9
3	1.25	Erlang-2	1.0	0.8	5	1
4	1.1	Erlang-2	1.0	0.90909...	1	9

Table 2-1: Parameter values for cases 1-4 of the single-stage base stock policy with ADI.

For $T = 0$, L is irrelevant, and S^* can be determined from (2.3). In cases 1 and 2, $P(\text{WIP} = n)$ can be computed analytically from (2.4) for $M = 4$. S^* can then be substituted into (2.2) to determine $C(S^*)$. The results are: $S^* = 8$ and $C(S^*) = 90.8954$, for case 1, and $S^* = 68$ and $C(S^*) = 83.6966$, for case 2. In cases 3 and 4, the optimal base stock level S^* was obtained by evaluating the cost for different values of S using simulation and picking the value that yielded the lowest cost. The results are: $S^* = 4$ and $C(S^*) = 48.07865$, for case 3, and $S^* = 34$ and $C(S^*) = 42.83576$, for case 4, respectively. Before discussing the results, let us say a few words about the simulation experiments.

For this and for all the other examples that follow in the rest of this chapter we ran over 10,000 simulations using the simulation software Arena. In each simulation we used a simulation run length of 60 million time units. This yielded 95% confidence intervals on the estimated values of $E[\text{WIP}]$, $E[\text{FG}]$ and $E[\text{BD}]$ with half width values less than 0.5% of their respective estimated values in the cases of $E[\text{WIP}]$ and $E[\text{FG}]$ and less than 4% in the case of $E[\text{BD}]$. The results are discussed next.

In all cases, the optimal planned supply lead time L^* can be determined from (2.5). In cases 1 and 2, it is well-known that the distribution of the order replenishment time W is Erlang with M phases and mean $M/(\mu - \lambda)$, so it can be computed analytically. This is because W is the sum of M iid M/M/1-system waiting times, each time having an exponential distribution with mean $1/(\mu - \lambda)$. Specifically, the cumulative distribution of W is given by

$$P(W \leq w) = 1 - \sum_{k=0}^{M-1} \frac{[(\mu - \lambda)w]^k}{k!} e^{-(\mu - \lambda)w}.$$

Substituting the above expression into (2.5) yields $L^* = 10.6396$ and $L^* = 73.4886$, for cases 1 and 2, respectively. Incidentally, a quick computation of $cE[W]$ yields $\ln[(5 + 1)/5]/[4/(1 - 0.8)] = 3.6464$ in case 1 and $\ln[(1 + 9)/1]/[4/(1 - 0.90909)] = 101.3137$ in case 2. Therefore, in

either case, $L^* \neq cE[W]$ (recall that in the case of a single-server exponential station, $L^* = cE[W]$). In fact, in case 1, $L^* (= 10.6396) > cE[W] (= 3.6464)$, whereas in case 2, $L^* (= 73.4886) < cE[W] (= 101.3137)$. Therefore, the fact that $L^* = cE[W]$ for the single-server exponential server case is special to this case and that does not hold in general.

In cases 3 and 4, L^* was obtained by evaluating the cost for different integers values of L using simulation and picking the value that yielded the lowest cost. The result is $L^* = 6$ and $L^* = 35$ for the two cases, respectively.

For values of T in the interval $(0, L^*)$ we used simulation to evaluate the cost of the system for the four sets of parameter values. In each case we optimized the control parameters S and L for different values of T , using exhaustive search. In this chapter we only present the optimal results due to space considerations. For all four sets of parameter values shown in Table 2-1, the optimization yielded the following general results.

As T increases from zero, the optimal base stock S^* appears to decrease linearly with T and reaches zero just below $T = L^*$, as in the case of the single-server station. The insight behind this behavior is that there appears to be a linear tradeoff between S^* and T and that L^* is just above the smallest value of T for which S^* is equal to zero. The optimal results are shown in Table 2-2 for the four cases. Plots of S^* versus T are shown in Figure 2-2 for the four cases.

From Figure 2-2 it can be seen that the smallest values of T for which $S^* = 0$ are approximately equal to 10, 73, 6 and 35, for cases 1-4 respectively. We say “approximately” because we only examined integer values of T , whereas T really is a continuous parameter. Recall that the analytically obtained optimal planned supply lead times L^* are 10.6396 and

73.4886 for cases 1 and 2, respectively. As in the case of the single-server station, the optimal planned supply lead times, L^* , are independent of T .

Case 1			Case 2			Case 3			Case 4		
T	S^*	$C(S^*, L^*)$	T	S^*	$C(S^*, L^*)$	T	S^*	$C(S^*, L^*)$	T	S^*	$C(S^*, L^*)$
0	8	90.8954	0	68	83.6966	0	4	48.0787	0	34	42.8358
2	6	90.5209	10	59	83.3837	2	3	47.6811	10	24	42.3458
4	5	90.3351	20	50	83.0256	4	1	47.2363	20	16	41.6722
6	3	90.0959	30	40	82.7439	6	0	46.8434	30	5	41.3582
8	2	89.9054	40	31	82.3999	∞	0	46.8434	35	0	41.1147
10	0	89.6463	50	22	82.1246				∞	0	41.1147
∞	0	89.6463	60	12	81.8376						
			73	0	81.6226						
			∞	0	81.6226						

Table 2-2: S^* and $C(S^*, L^*)$ versus T , for $L = L^*$, for the single-stage base stock policy with ADI.

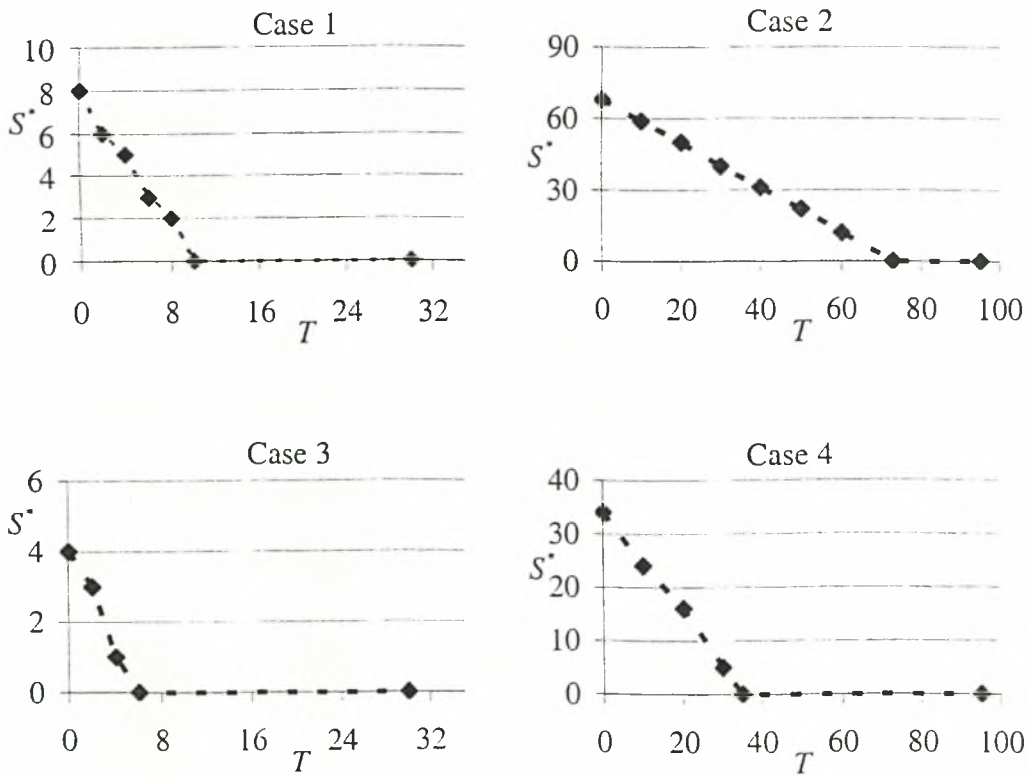


Figure 2-2: S^* versus T , for $L = L^*$, for the single-stage base stock policy with ADI.

From Table 2-2 it can be seen that the minimum long run expected average cost $C(S^*, L^*)$ decreases very little with T and attains its minimum value at $T = L^*$. The drop in $C(S^*, L^*)$ between the cases $T = 0$ and $T = L^*$ is only 1.37%, 2.48%, 2.29% and 4.50%, for cases 1-4, respectively. The insensitivity of the long run expected average cost with respect to the demand lead time T is to a certain extent due to the fact that a significant part of that cost given by (2.1) is due to the term $hE[\text{WIP}]$, which is independent of T . Had we omitted this term from the long run expected average cost, the drop in $C(S^*, L^*)$ between the cases $T = 0$ and $T = L^*$ would have been 10.06%, 4.38%, 20.91% and 7.09%, for cases 1-4, respectively. To summarize, the basic insights behind the results are the following.

In a single-stage system controlled by a single-stage base stock policy: (a) there appears to be a linear tradeoff between the demand lead time and the optimal base stock level, and (b) the optimal planned supply lead time appears is the smallest demand lead time for which the optimal base stock level is zero. This means that if the demand lead time is smaller than the optimal planned supply lead time, the optimal base stock level is positive and a production replenishment order is issued immediately after the arrival of the customer demand that triggered it. On the other hand, if the demand lead time is greater than the optimal planned supply lead time, the optimal base stock level is zero and a production replenishment order is issued with a delay equal to the difference between the demand lead time and planned supply lead time after the arrival of the customer demand that triggered it.

2.3 Single-Stage Hybrid Base Stock/Kanban Policy with ADI

A single-stage hybrid base stock/kanban policy with ADI behaves exactly like a single-stage base stock policy with ADI as far as the issuing of replenishment production orders is concerned. The difference is that in a single-stage hybrid base stock/kanban policy,

when a replenishment production order is issued, it is not immediately authorized to go through (as is the case in a base stock policy) unless the inventory in the facility (i.e. WIP) or in the entire system (i.e. WIP + FG) is below a given *inventory cap* level.

Setting an inventory cap in any section of a production-inventory system makes sense if this section and/or the section downstream of it have limited processing capacity. This is because releasing a part in an already congested section of the system with limited processing capacity, or in a section without limited processing capacity (e.g. a buffer) but which is followed by a section with limited processing capacity, will increase the inventory in that section with little or no decrease in the part's completion time. In the kind of multi-stage serial systems that we study in this chapter, where each stage consists of a facility containing WIP and an output store containing FG inventory, all facilities have limited processing capacity, and all output buffers, except the output buffer of the last stage, are followed by facilities which have limited processing capacity. In such systems, therefore, it makes sense to set a (WIP + FG) *cap* on the (WIP + FG) inventory of all stages except the last one and to set a *WIP cap* on the WIP of the last stage. In the case of a single-stage system considered in this section, the one and only stage is the last stage; therefore for a single-stage system we will only consider a base stock/kanban policy where a *WIP cap* is set on the WIP of the stage.

With the above discussion in mind, in a single-stage hybrid base stock/kanban policy, when a replenishment production order is issued, it is not immediately authorized to go through unless the WIP in the system is below a given *WIP cap* of K parts. If the WIP in the system is at or above K , the order is put on hold until the WIP drops below K (the inventory drops as parts exit the facility). Once the order is authorized to go through, a new part is immediately released into the facility. This policy can be implemented by requiring that every part entering the facility be granted a production authorization card known as *kanban*, where

the total number of kanbans is equal to the WIP cap level. Once a part leaves the facility, the kanban that was granted (and attached) to it is detached and is used to authorize the release of a new part into the facility. Notice that a single-stage hybrid base stock/kanban policy with no ADI is equivalent to a single-stage generalized kanban policy [15], [75].

The system starts with a base stock of S end-items in FG inventory and K free kanbans that are available to authorize an equal number of replenishment production orders. The number of free kanbans represents the number of parts that can be released into the facility before the WIP in the system reaches the WIP cap level K . A queuing network model of a hybrid base stock/kanban policy with ADI is shown in Figure 2-3, where queue FK contains *free kanbans*.

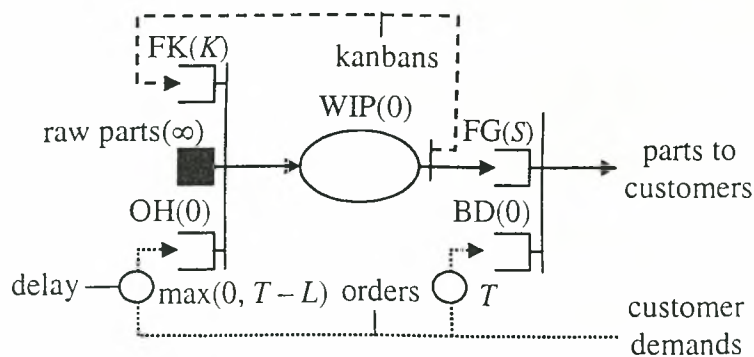


Figure 2-3: Single-stage hybrid base stock/kanban policy with ADI.

From Figure 2-3 it can be seen that kanbans trace a loop within a closed network linking FK and WIP. The constant population of this closed network is K , i.e. at all times, $FK + WIP = K$. The throughput of this closed network, denoted TH_K , depends on K and determines the processing capacity of the system, i.e. the maximum demand rate λ that the system can meet in the long run. Under some fairly general conditions (that essentially require that the facility exhibits “max-plus” behavior in the sense that the timings of events in the system can be expressed as functions of the timings of other events involving the operators

“max” and “+” only), TH_K is an increasing concave function of K , such that $TH_0 = 0$ and $TH_\infty < \infty$. For every feasible demand rate λ , such that $\lambda < TH_\infty$, there is a finite minimum value of K , K_{\min} , such that for any $K \geq K_{\min}$, $TH_K > \lambda$, which means that the system has enough capacity to meet demand in the long run.

A single-stage hybrid base stock/kanban policy includes single-stage base stock and kanban policies as special cases. Namely, a single-stage hybrid base stock/kanban policy with $K = \infty$ and $S < \infty$ is equivalent to a single-stage base stock policy with base stock S . A single-stage hybrid base stock/kanban policy with $K = S < \infty$ is equivalent to a single-stage kanban policy or a make-to-stock CONWIP policy with K (or equivalently, S) kanbans [28].

We consider an optimization problem similar to that in Section 2.2, where the objective is to find the values of K , S , and L that minimize the long run expected average cost of holding and backordering inventory,

$$C(K,S,L) = hE[WIP_K + FG_K(S,L)] + bE[BD_K(S,L)], \quad (2.6)$$

where h and b are defined as in Section 2.2. It is not difficult to see that control parameters S and L affect only the expected average FG and BD and not the expected average WIP or OH, whereas parameter K affects the expected average FG, BD as well as WIP and OH. We explicitly express these dependencies in the cost function (2.6).

2.3.1 The Case where There Is No ADI

If there is no ADI, i.e. if $T = 0$, the planned supply lead time parameter L is irrelevant, and the optimal base stock level for any given value of K , such that $K \geq K_{\min}$, S_K^* , is the smallest integer that satisfies (see [52])

$$P(OH_K + WIP_K \leq S_K^*) \geq b/(b+h). \quad (2.7)$$

If the facility consists of a Jackson network of servers, there is no analytical expression (not even in non-closed form) to determine the steady-state distribution of OH_K and WIP_K and therefore S_K^* , and only approximate methods exist (e.g. see [30]). To shed some light into this case, we numerically investigated the same instance of the system that we investigated in Section 2.2.2, i.e. an instance in which the facility consists of a Jackson network of $M = 4$ identical single server stations in series, for the same four sets of parameter values shown in Table 2-1. For cases 1 and 2, TH_K can be calculated analytically as $TH_K = \mu/[1 + (M - 1)/K]$ [30]. Since K_{\min} is the smallest integer for which $TH_{K_{\min}} > \lambda$, it follows that K_{\min} is the smallest integer that satisfies $\mu/[1 + (M - 1)/K_{\min}] > \lambda$, i.e. $K_{\min} > (M - 1)\rho/(1 - \rho)$, where $\rho = \lambda/\mu$. This implies that K_{\min} is equal to 13 and 30, for cases 1 and 2, respectively.

We used simulation to evaluate the long run expected average cost of the system for the four sets of parameter values, and in each case we found the optimal base stock levels for different values of K , S_K^* , using exhaustive search. The optimal results are shown in Table 2-3. Plots of S_K^* versus K are shown in Figure 2-4 for the four cases. The values for $K = \infty$ are taken from Table 2-2 for the case where $T = 0$.

Case 1			Case 2			Case 3			Case 4		
K	S_K^*	$C(K, S_K^*)$	K	S_K^*	$C(K, S_K^*)$	K	S_K^*	$C(K, S_K^*)$	K	S_K^*	$C(K, S_K^*)$
13	15	104.7284	33	227	246.6715	6	5	40.4173	13	54	57.4847
14	10	88.4036	40	96	109.8939	7	4	38.3957	15	43	47.4977
15	11	84.0617	45	81	94.3872	8	4	39.3074	20	36	41.9549
16	9	82.9463	50	75	88.3950	9	4	40.6576	22	35	41.4670
17	8	82.7963	60	71	84.4438	10	4	41.9764	23	34	41.1375
18	8	83.1151	68	69	83.6047	11	4	43.0968	24	34	41.1463
∞	8	90.8954	69	68	82.8599	∞	4	48.0787	25	34	41.3125
			70	68	82.9287				∞	34	42.8358
			∞	68	83.6966						

Table 2-3: S_K^* and $C(S_K^*, L^*)$ versus K for the single-stage hybrid base stock/kanban policy with no ADI.

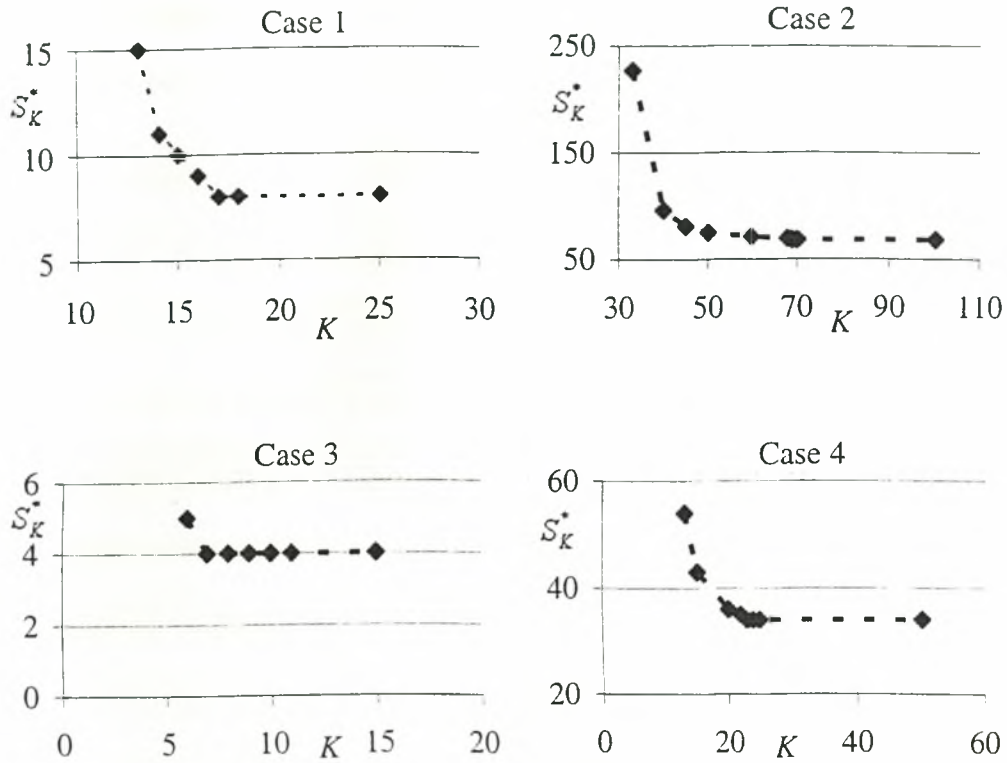


Figure 2-4: S_K^* and $C(S_K^*, L^*)$ versus K for the single-stage hybrid base stock/kanban policy with no ADI.

From Table 2-3 and Figure 2-4, it can be seen that the optimal base stock level S_K^* is non-increasing in K , i.e. $S_{K+1}^* \leq S_K^*$, for $K \geq K_{\min}$. Moreover, there exists a finite critical value of K , K_c , such that $S_K^* = S_\infty^*$, for $K \geq K_c$, where S_∞^* is the optimal base stock level for the same system operating under a pure base stock policy, i.e. a hybrid base stock/kanban policy with $K = \infty$. This means that there is a tradeoff between K and S_K^* and that this tradeoff holds for up to a finite critical value of K , K_c . This critical value is equal to 17, 69, 7 and 23, for cases 1-4, respectively. The same result is proven analytically in [52] for a slightly different but equivalent system where the objective is to minimize the long run expected average cost of

holding inventory subject to a specified fill rate constraint. The insight behind it is the following.

As K increases, parts are released in the facility earlier and depart from it earlier, causing the average FG inventory to increase too. At the same time, the congestion in the system also increases, and therefore parts stay in the facility longer. This implies that the rate of increase of the average FG inventory is diminishing in K . At $K = K_c$, the facility reaches a critical congestion level. That is, for values of K below K_c , the system is *under-congested* in the sense that increasing K , causes an increase in the average FG inventory that is enough to warrant a decrease in S_K^* . For values of K at or above K_c , however, the system is *over-congested* in the sense that increasing K , does not cause an increase in the average FG inventory that is enough to warrant a decrease in S_K^* . An important question that remains to be answered is what is the overall optimal number of kanbans K^* and the resulting optimal base stock level S_K^* ?

The most striking result of the optimization is that in all four cases the overall optimal number of kanbans, K^* , is equal to K_c , and therefore the overall optimal base stock level, S^* , is equal to S_∞^* . More specifically, K^* is equal to 17, 69, 7 and 23, for cases 1-4, respectively, and S^* is equal to 8, 68, 4 and 34, for cases 1-4, respectively. This is not an obvious result. The insight behind it is that the optimal base stock level of the hybrid base stock/kanban policy, S^* , appears to be equal to the optimal base stock level of a pure base stock policy, S_∞^* , which is the smallest possible value of S_K^* . Moreover, the optimal number of kanbans, K^* , is the smallest value of K for which $S_K^* = S_\infty^*$. In other words, it is optimal to set K to a value that is just big enough so that the corresponding optimal base stock level is equal to the optimal base stock level of a pure base stock policy, S_∞^* . This means that a pure base stock policy is never

optimal but the optimal base stock level of a pure base stock policy is the optimal base stock level for a hybrid base stock/kanban policy too. Computational experience reported in [29], [43], [76] (sec. 8.8.2) for simpler single-machine systems also confirms this result. The difficulty in proving it stems from the fact that no analytical expression for the steady-state distribution of OH_K and WIP_K exists, except for a trivial system where the facility consists of a single-server station with exponential service rate μ , in which case $K_{\min} = K_c = 1$. Nevertheless, an indication of the validity of this result is given in [52].

The results also suggest that the long run expected average cost increases more steeply as K decreases from K^* than it does as K increases from K^* . This means that it is more costly to underestimate K relatively to the optimal value K^* than to overestimate it. Of course, as $K \rightarrow \infty$, the long run expected average cost approaches $C(\infty, S_n^*)$, i.e. the minimum cost of a pure base stock policy. In our numerical example, the minimum long run expected average cost under the optimal base stock policy is 90.8954, 83.6966, 48.0787 and 42.8358, for cases 1-4, respectively, as is seen in Table 2-2, whereas, the minimum long run expected average cost under the optimal hybrid base stock/kanban policy is 82.7963, 82.8599, 38.3957 and 41.1375, for cases 1-4, respectively, as is seen in Table 2-3. This means that the minimum long run expected average cost is 8.91%, 1%, 20.14% and 3.96% smaller under the optimal hybrid base stock/kanban policy than it is under the optimal base stock policy, for cases 1-4, respectively. The fact that the reduction in the cost is more dramatic in case 1 than in case 2 (and similarly in case 3 than in case 4) is due to two reasons. The first reason is that the cost ratio h/b is higher in case 1 than in case 2, so reducing the WIP with a WIP-cap is more useful in case 1 than in case 2, since every part in WIP costs relatively more. The second reason is that the utilization coefficient, ρ , is higher in case 2 than in case 1, so the distribution of the inter-departure times from the facility is less sensitive to the distribution of the inter-arrival

times to the facility in case 2 than in case 1. This implies that S_K^* and $C(K, S_K^*)$ are less sensitive to K in case 2 than in case 1. Finally, the fact that the reduction in cost is more dramatic in case 3 than in case 1 (and similarly in case 4 than in case 2) implies that imposing a WIP-cap mechanism is more beneficial in a system with lower flow time variability.

2.3.2 The Case where There Is ADI

If there is ADI, i.e. if $T > 0$, and the facility consists of a Jackson network of servers, there are no analytical results available for the optimal parameter values. Intuitively, we would expect that as T increases, the optimal base stock level of a hybrid base stock/kanban policy should decrease. The question is how exactly does it decrease, in particular with respect to the optimal base stock level of a pure base stock policy? Also, does the optimal number of kanbans decrease too?

To shed some light into this case, we numerically investigated the same instance of the system that we investigated in Sections 2.2.2 and 2.3.1, i.e. an instance in which the facility consists of a Jackson network of $M = 4$ identical single-server stations in series, for the same four sets of parameter values shown in Table 2-1. We used simulation to evaluate the long run expected average cost of the system for the four cases, and in each case we optimized the control parameters, K , S and L , for different values of T , using exhaustive search. The optimal results are shown in Table 2-4 for selected values of K around the optimal values and $L = L^*$.

From the results in Table 2-4, it appears that in all cases, the optimal number of kanbans, K^* , is equal to K_c for all values of T , i.e. K^* is independent of T . Namely, K^* is equal to 17, 69, 7 and 23, for cases 1-4, respectively. Moreover, L^* and S^* have the same values as in the single-stage base stock policy with ADI discussed in Section 2.2.2. Namely, L^* is approximately equal to 10, 73, 6 and 35, for cases 1-4, respectively, and S^* has the same

values as those shown in Table 2-2. This is not an obvious result. The insight behind it is the following.

Case 1				Case 2			
T	K	S_K^*	$C(K, S_K^*)$	T	K	S_K^*	$C(K, S_K^*)$
4	16	6	82.6148	40	67	33	82.2823
	17	5	82.4321		69	31	81.7778
	18	5	82.7398		71	31	82.0570
8	16	3	82.2708	60	67	14	81.7176
	17	2	82.0512		69	12	81.2404
	18	2	82.3473		71	12	81.5260
10	16	1	82.0972	73	67	2	81.3691
	17	0	81.9289		69	0	80.8782
	18	0	82.1906		71	0	81.1567

Case 3				Case 4			
T	K	S_K^*	$C(K, S_K^*)$	T	K	S_K^*	$C(K, S_K^*)$
2	6	4	40.1270	10	21	26	41.2114
	7	3	38.0275		23	24	40.7101
	8	3	38.9747		25	24	40.8552
4	6	2	39.8828	20	21	17	40.7926
	7	1	37.6979		23	16	40.2179
	8	1	38.5364		25	16	40.4074
6	6	1	39.5846	35	21	2	40.0908
	7	0	37.2553		23	0	39.5676
	8	0	38.1188		25	0	39.6699

Table 2-4: S_K^* and $C(S_K^*, L^*)$ versus T and K , for $L = L^*$, for the single-stage hybrid base stock/kanban policy with ADI.

When $T = 0$, then $S^* > 0$. When $S^* > 0$, it appears to be optimal to issue a replenishment production order immediately upon the arrival of a customer demand to the system, irrespectively of the value of T (as long as T is small enough so that $S^* > 0$). Whenever a replenishment production order is issued immediately upon the arrival of a customer demand to the system, T does not affect what goes on in the facility but only affects FG and BD. As such, T is a tradeoff for S^* , where S^* also affects only FG and BD. Therefore, the value of K that determines the optimal processing capacity and congestion level in the

facility when $T = 0$, namely K_c , is also optimal when $T > 0$. The results in Section 2.3.1 showed that for $T = 0$, S^* is equal to the optimal base stock level of a pure base stock policy, S_{∞}^* . For $T > 0$, it appears that the tradeoff which exists between T and S^* is exactly the same as the tradeoff between T and S_{∞}^* in a pure base stock policy. In other words, $S^* = S_{\infty}^*$, for $T > 0$. This means that as T increases from zero, S^* decreases and reaches zero at T just below L^* , exactly as in a pure base stock policy. To summarize, the basic insights behind the results are the following.

In a single-stage system controlled by a single-stage hybrid base stock/kanban policy, when there is no ADI, i.e. when the demand lead time is zero, there is a tradeoff between the optimal base stock level and the number of kanbans. This tradeoff holds for up to a finite critical number of kanbans. This means that whenever the number of kanbans is above this critical number, the optimal base stock level is at a minimum value, which is equal to the optimal base stock level of a single-stage base stock policy with no ADI. The critical number of kanbans and the corresponding minimum base stock level appear to be the optimal parameters of the single-stage hybrid base stock/kanban policy when the demand lead time is zero. Moreover, the same critical number of kanbans appears to be also optimal for all demand lead times. In addition, the optimal base stock level of a pure base stock policy appears to be also optimal for a hybrid base stock/kanban policy for all demand lead times. This means that the same linear tradeoff between the optimal base stock level and the demand lead time that appears to hold for a pure base stock policy also holds for a hybrid base stock/kanban policy.

2.4 Two-Stage Base Stock Policy with ADI

In this section we extend the single-stage base stock policy with ADI considered in Section 2.2 to two stages. The two-stage base stock policy with ADI is similar to the policy considered in [44]. In the two-stage policy, customer demands arrive for one end item at a time according to a Poisson process with rate λ , with a constant demand lead time, T , in advance of their due dates, as is the case with the single-stage policy. The arrival of every customer demand eventually triggers the consumption of an end-item from FG inventory and the issuing of a replenishment production order to the facility of each of the two stages in the system. More specifically, the consumption of an end-item from FG inventory is triggered T time units after the arrival time of the demand, as is the case in the single-stage policy. If no end-items are available at that time, the demand is backordered. The control policy depends on four design parameters, the base stock level of end-items in FG inventory at stage n , $n = 1, 2$, denoted by S_n , and the planned supply lead time of stage n , $n = 1, 2$, denoted by L_n . Initially, the system starts with a base stock of S_n end-items in FG inventory at stage n , $n = 1, 2$. The time of issuing the replenishment order at stage 2 is determined by offsetting the demand due date by the stage planned supply lead time, L_2 . The time of issuing the replenishment order at stage 1 is determined by offsetting the demand due date by the sum of the planned supply lead times of stages 1 and 2, $L_1 + L_2$. This means that the delay in issuing an order at stage 2 is equal to $\max[0, T - L_2]$, as is the case in the single-stage policy, whereas the delay in issuing an order at stage 1 is equal to $\max[0, T - (L_1 + L_2)]$. In general, in a system with N stages, the delay in issuing an order at stage n is equal to $\max[0, T - L_n^c]$, where L_n^c denotes the *echelon planned supply lead time* at stage n , which is defined as $L_n^c = L_n + L_{n+1} + \dots + L_N$, $n = 1, 2, \dots, N$. When an order is issued at stage 1, a new part is

immediately released into the facility of stage 1. When an order is issued at stage 2, a new part is also immediately released into the facility of stage 2, provided that such a part is available in the FG output store of stage 1. Otherwise, the order remains on hold until a part becomes available in the FG output store of stage 1. If there is no ADI, i.e. if $T = 0$, both the consumption of an end-item from FG inventory and the replenishment orders are triggered at the demand arrival time, and the resulting policy is a classical base stock policy. A queuing network model of a base stock policy with ADI is shown in Figure 2-5.

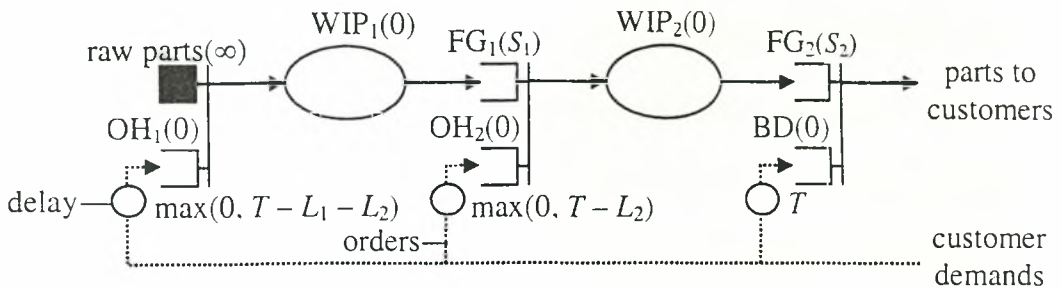


Figure 2-5: Two-stage base stock policy with ADI.

We consider an optimization problem similar to that in Section 2.2, where the objective is to find the values of S_1 , S_2 , L_1 , and L_2 that minimize the long run expected average cost of holding and backordering inventory,

$$C(S_1, S_2, L_1, L_2) = h_1 E[WIP_1 + FG_1(S_1, L_1, L_2)] + h_2 E[WIP_2(S_1, L_1, L_2) + FG_2(S_1, S_2, L_1, L_2)] + b E[BD(S_1, S_2, L_1, L_2)] \quad (2.8)$$

where h_n is the unit cost of holding $WIP + FG$ inventory per unit time at stage n and b is the unit cost of backordering end-item inventory per unit time. In expression (2.8), we explicitly express the dependencies of WIP_1 , FG_1 , WIP_2 , FG_2 , and BD on parameters S_1 , S_2 , L_1 , and L_2 .

2.4.1 The Case where There Is No ADI

If there is no ADI, i.e. if $T = 0$, the planned supply lead time parameters L_1 and L_2 are irrelevant. Unfortunately, even in this case there are no analytical results available for the optimal base stock levels S_1^* and S_2^* , even when each facility consists of a Jackson network of servers. Some approximation methods have been developed in [16] (sec. 10.7), [29], [1] (sec. 8.3.4.3). The only analytically tractable case is the case where $S_1 = 0$. In this case, the two-stage policy is equivalent to a single-stage policy where the facilities of stages 1 and 2 are merged into a single facility. This is useful to know because in case $h_1 \geq h_2$, i.e. in case holding FG inventory at stage 1 is at least as expensive as holding FG inventory at stage 2, it does not make sense to hold FG inventory at stage 1, given that FG inventory is mostly needed at stage 2 to better respond to customer demand, hence $S_1^* = 0$. Therefore, if $T = 0$, the only interesting case to look at is the case where $h_1 < h_2$. In what follows, we will therefore assume that $h_1 < h_2$.



2.4.2 The Case where There Is ADI

If there is ADI, i.e. if $T > 0$, there are no analytical results available for the optimal parameter values. As in the single-stage base stock policy with ADI considered in Section 2.2 intuitively, we would expect that as T increases, the optimal base stock levels of both stages should decrease. The question is how exactly do they decrease in T ?

To shed some light into these issues, we numerically investigated a particular instance of the system, in which each facility consists of a Jackson network of $M = 2$ identical single-server stations in series, each server having an exponential service rate μ . For this instance, we considered the set of parameter values shown in Table 2-5. We only looked at one case

because there are four parameters to optimize and optimization via simulation is computationally very demanding. The inventory holding cost rates are such that $h_1 < h_2$, so that $S_1^* > 0$ (recall that if $S_1 = 0$, the two-stage policy is equivalent to a single-stage policy where the facilities of stages 1 and 2 are merged into a single facility).

Case	$1/\lambda$	$1/\mu$	$P = \lambda/\mu$	h_1	h_2	b
1	1.1	1.0	0.90909...	1	3	9

Table 2-5: Parameter values for the case of the two-stage base stock policy with ADI.

For this set of parameter values, we used simulation to evaluate the long run expected average cost of the system, and we optimized the control parameters, S_1 , S_2 , L_1 and L_2 , for different values of T , using exhaustive search. The optimization yielded the following results.

T	S_1^*	S_2^*	$C(S_1^*, S_2^*)$
0	24	32	158.7183
10	24	23	157.8996
20	24	13	157.0982
25	24	9	156.6986
33	24	1	156.0171
34	24	0	155.9370
40	17	0	155.7578
50	10	0	154.8409
60	1	0	155.2108
61	0	0	154.9056
95	0	0	155.0616

Table 2-6: S_1^* , S_2^* and $C(S_1^*, S_2^*)$ versus T , for $L_1 = L_1^*$ and $L_2 = L_2^*$, for the two-stage base stock policy with ADI.

For $T = 0$, L_1 and L_2 are irrelevant and $S_1^* = 24$ and $S_2^* = 32$. As T increases from zero, S_1^* remains constant, while S_2^* decreases—apparently linearly—with T and reaches zero just below $T = L_2^*$, where $L_2^* = 34$. As T increases from L_2^* , S_2^* remains zero, while S_1^* decreases linearly with T and reaches zero just below $T = L_1^* + L_2^*$, where $L_1^* + L_2^* = 61$, therefore

$L_1^* = 27$. The optimal values of S_1^* and S_2^* and the resulting cost $C(S_1^*, S_2^*)$ versus T are shown in Table 2-6 and are plotted in Figure 2-6, for the optimal planned supply lead times $L_1^* = 27$ and $L_2^* = 34$. Table 2-7 shows the optimal values of S_1^* and S_2^* and the resulting cost $C(S_1^*, S_2^*)$ versus T and selected values of L_1 and L_2 around the optimal values.

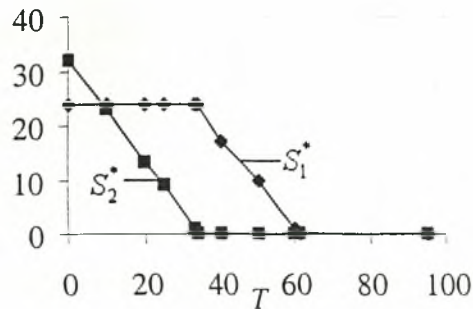


Figure 2-6: S_1^* , S_2^* and $C(S_1^*, S_2^*)$ versus T , for $L_1 = L_1^*$ and $L_2 = L_2^*$, for the two-stage base stock policy with ADI.

T	L_1	L_2	S_1^*	S_2^*	$C(S_1^*, S_2^*)$
34	27	33	25	0	156.1356
	27	34	24	0	155.9370
	27	35	24	0	155.9370
40	27	33	18	0	156.0771
	27	34	17	0	155.7578
	27	35	15	0	156.5931
61	26	34	6	0	155.9039
	27	34	0	0	154.9056
	28	34	0	0	154.9056

Table 2-7: S_1^* , S_2^* and $C(S_1^*, S_2^*)$ versus T , L_1 and L_2 for the two-stage base stock policy with ADI.

The insight behind the results is the following. When $T = 0$, $S_1^* > 0$ and $S_2^* > 0$. When $S_1^* > 0$ and $S_2^* > 0$, it seems that it is optimal to issue a replenishment production order

immediately upon the arrival of a customer demand to each stage irrespectively of the value of T (as long as T is small enough so that $S_1^* > 0$ and $S_2^* > 0$). Whenever a replenishment production order is issued immediately upon the arrival of a customer demand to each stage, T does not affect what goes on in either facility but only affects the FG inventory of stage 2 and BD. In this case, T is a tradeoff for S_2^* , where S_2^* also affects only the FG inventory of stage 2 and BD. Therefore, as T increases from zero, it is optimal to reduce only S_2^* and not S_1^* . When T is just below L_2^* , S_2^* becomes zero. As T increases beyond L_2^* , S_2^* remains at zero, and orders are issued at stage 2 with a delay of $T - L_2^*$. At the same time, S_1^* starts decreasing with T , while orders are still issued at stage 1 with no delay. When T is just below $L_1^* + L_2^*$, S_1^* becomes zero too. As T increases beyond $L_1^* + L_2^*$, both S_1^* and S_2^* remain at zero, while orders are issued at stages 2 and 1 with delays of $T - L_2^*$ and $T - (L_1^* + L_2^*)$, respectively. As in the case of the single-server station, the optimal planned supply lead times are independent of T . The minimum long run expected average cost decreases very little with T and attains its minimum value at $T = L_1^* + L_2^*$.

The results imply that as T increases and therefore more demand information becomes available in advance, the optimal base stock levels of all stages seem to drop to zero one after the other, starting from the last stage. An alternative way of looking at this is that as T increases, the optimal *echelon* base stock level of every stage drops to zero, where by echelon base stock of a stage we mean the sum of the base stock levels of the stage and all its downstream stages. Moreover, replenishment production orders are issued with a delay at a stage only when T is large enough so that the optimal echelon base stock level of the stage is zero. To summarize, the basic insights behind the results are the following.

In a two-stage system controlled by a single-stage base stock policy, at every stage: (a) there appears to be a linear tradeoff between the demand lead time and the optimal echelon base stock level, and (b) the optimal echelon planned supply lead time appears to be the smallest demand lead time for which the optimal echelon base stock level is zero.

2.5 Two-Stage Hybrid Base Stock/Kanban Policy with ADI

A two-stage hybrid base stock/kanban policy with ADI is an extension of a single-stage hybrid base stock/kanban policy with ADI presented in Section 2.3 to two stages. Recall from our discussion in the second paragraph of Section 2.3 that in the kind of multi-stage serial systems that we study in this chapter, it makes sense to set a $(WIP + FG)$ cap on the $(WIP + FG)$ inventory in all but the last stage and to set a WIP cap on the WIP of the last stage; therefore for the two-stage system considered in this section, we will only consider a hybrid base stock/kanban policy where a $(WIP + FG)$ cap is set on the $(WIP + FG)$ inventory of the first stage and WIP cap is set on the WIP of the second stage.

With the above discussion in mind, a two-stage hybrid base stock/kanban policy with ADI behaves exactly like a two-stage base stock policy with ADI as far as the issuing of replenishment production orders is concerned. The difference is that in a two-stage hybrid base stock/kanban policy, when a replenishment production order is issued to the facility of stage 1, it is not immediately authorized to go through unless the $(WIP + FG)$ inventory at stage 1 is below a given $(WIP + FG)$ cap of K_1 parts. If the $(WIP + FG)$ inventory at stage 1 is at or above K_1 , the order is put on hold until the $(WIP + FG)$ inventory drops below K_1 (the $(WIP + FG)$ inventory drops as FG parts from stage 1 are consumed by stage 2). Once the order is authorized to go through, a new part is immediately released into the facility. This policy can be implemented by requiring that every part entering the facility be granted a

kanban, where the total number of kanbans is equal to the (WIP + FG) cap level. Once a part leaves the FG output store, the kanban that was granted (and attached) to it is detached and is used to authorize the release of a new part into the facility. A similar mechanism is in place at stage 2, except that it is the WIP rather than the (WIP + FG) inventory that is constrained, i.e. when a replenishment production order is issued to the facility of stage 2, it is not immediately authorized to go through unless the WIP at stage 2 is below a given WIP cap of K_2 parts.

Notice that a two-stage hybrid base stock/kanban policy with no ADI is equivalent to a mixture of an extended kanban policy [23] at stage 1 and a generalized kanban policy [15], [75] at stage 2. A queuing network model of a two-stage hybrid base stock/kanban policy with ADI is shown in Figure 2-7.

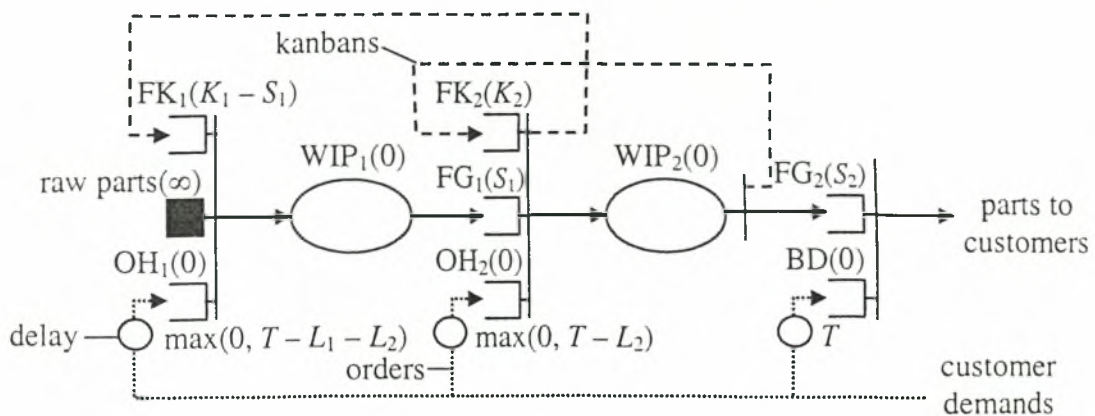


Figure 2-7: Two-stage hybrid base stock/kanban policy with ADI.

If there is ADI, i.e. if $T > 0$, there are no analytical results available for the optimal parameter values. To shed some light into this case, we numerically investigated the same instance of the system as that in Section 2.4, for the same set of parameter values shown in Table 2-5. For this set of parameter values, we used simulation to evaluate the long run

expected average cost of the system, and we set out to optimize the control parameters K_1 , K_2 , S_1 , S_2 , L_1 , and L_2 for different values of T , using exhaustive search.

The results of the optimization indicate that the properties of the optimal parameter values are similar to those of the optimal parameter values in the single-stage hybrid base stock/kanban policy. Namely, for $T = 0$, L_1 and L_2 are irrelevant, and S_1^* and S_2^* are equal to the optimal base stock levels for the two-stage pure base stock policy, i.e. $S_1^* = 24$ and $S_2^* = 32$. Moreover, the optimal numbers of kanbans K_1^* and K_2^* are the smallest values of K_1 and K_2 for which the optimal base stock levels are equal to the optimal base stock levels in the two-stage pure base stock policy. These values are $K_1^* = 44$ and $K_2^* = 28$.

For $T > 0$, K_1^* and K_2^* remain constant for all values of T , whereas L_1^* , L_2^* , S_1^* and S_2^* have the exact same values as in the two-stage base stock policy with ADI discussed in Section 2.4. The insight behind these results is the same as that behind the results for the single-stage hybrid base stock/kanban policy.

The optimal results are shown in Table 2-8: for selected values of K_1 and K_2 around the optimal values and $L_1 = L_1^*$, $L_2 = L_2^*$.

2.6 Conclusions

We numerically investigated the tradeoffs between optimal base stock levels, numbers of kanbans, and planned supply lead times in single-stage and two-stage production-inventory systems operating under base stock and hybrid base stock/kanban policies with ADI. The results of our investigation lead to the following conjectures.

T	K_1	K_2	S_1^*	S_2^*	$C(S_1^*, S_2^*)$
0	42	26	31	32	155.7978
	42	28	26	32	155.0267
	42	30	26	32	155.0935
	44	26	27	32	155.1304
	44	28	24	32	154.8046
	44	30	24	32	154.9027
	46	26	26	32	154.9793
	46	28	24	32	156.6758
	46	30	24	32	155.7128
20	42	26	31	13	154.1961
	42	28	26	13	153.5168
	42	30	26	13	153.5238
	44	26	27	13	153.6286
	44	28	24	13	153.3745
	44	30	24	13	153.3911
	46	26	26	13	153.4994
	46	28	24	13	155.2802
	46	30	24	13	154.2590
40	42	26	21	0	153.3327
	42	28	19	0	153.0503
	42	30	19	0	153.2261
	44	26	20	0	153.1847
	44	28	17	0	152.9341
	44	30	17	0	154.1602
	46	26	19	0	153.0693
	46	28	17	0	153.0853
	46	30	17	0	153.1135
95	42	26	0	0	153.1015
	42	28	0	0	151.6153
	42	30	0	0	152.4563
	44	26	0	0	152.9508
	44	28	0	0	151.4917
	44	30	0	0	152.8287
	46	26	0	0	153.0745
	46	28	0	0	152.9572
	46	30	0	0	152.6148

Table 2-8: S_1^* , S_2^* and $C(S_1^*, S_2^*)$ versus T , K_1 and K_2 , for $L_1 = L_1^*$ and $L_2 = L_2^*$, for the two-stage hybrid base stock/kanban policy with ADI.

In multi-stage make-to-stock production-inventory control policies in which a base stock level of FG inventory is set at every stage, that base stock level represents finished goods that have been produced before any demands have arrived to the system to protect the

system against uncertainties in production or demand that may cause costly backorders. Holding inventory, however, is itself costly.

The results in this chapter indicate that the optimal base stock level at every stage should be as low as possible. The lowest possible optimal base stock level is attained when the replenishment policy adopted is such that a replenishment order is issued and released into the facility of every stage immediately after the arrival of the customer demand that triggered it. This can be achieved by setting the echelon planned supply lead time at the first stage greater than or equal to the demand lead time, and by setting the number of kanbans equal to infinity at every stage, so that no inventory limit is imposed at any stage. Delaying or postponing the issuing of a replenishment production order by means of an offsetting by the echelon planned supply lead time mechanism or a kanban mechanism appears to lower the total cost as long as it does not cause an increase in the optimal base stock level of FG inventory above its lowest possible value.

Moreover, for a fixed demand lead time, the more downstream a stage is, the less advance demand information is available and so the higher the need to keep a base stock of FG inventory of that stage. As the demand lead time increases, the amount of advance demand information increases from downstream to upstream, and so the need to keep a base stock of FG inventory at each stage decreases from downstream to upstream. The results in this chapter indicate that it is optimal to reduce the optimal base stock levels at all stages until they drop to zero, one after the other, starting from the last stage and moving upstream the system.

Finally, the optimal number of kanbans determines the optimal production capacity of the system and appears to be independent of the amount of ADI.

Chapter 3 An Analytical Method for the Performance Evaluation of Echelon Kanban Control Systems

3.1 Introduction

In this chapter, we develop an analytical approximation method for the performance evaluation of an echelon kanban control system and test it with several numerical examples. The term echelon kanban control system or policy was introduced in [53] to describe a kanban-type mechanism for the production control of multi-stage production-inventory systems. When a multi-stage production/inventory system is controlled by an echelon kanban control system, each stage in the system has associated with it a number of tags called echelon kanbans that are used to request and authorize the release of parts into the stage. Specifically, if a part to be released into a stage exists and an echelon kanban of that stage is available, the echelon kanban is attached onto the part and together they are released into the stage. The echelon kanban then follows the part all the way to the end of the system and is detached from the part when the part leaves the system. Once an echelon kanban is detached from a part, it becomes available to once again request and authorize the release of a new part into the stage. An echelon kanban control system can be alternatively viewed as a policy where whenever a part leaves the system (i.e. the last stage), a new part is requested and authorized to be released into every stage simultaneously. It is worth noting that the integral control system described in [16] is equivalent to an echelon kanban control system.

The difference between a conventional kanban system, which is also referred to as installation kanban control system or policy in [53], and an echelon kanban control system is that in the former system, a so-called installation kanban of a stage follows a part through the stage and is detached from it when the part exits the stage, whereas in an echelon kanban control system, an echelon kanban of a stage follows a part through the stage and all its downstream stages and is detached from it when the part exits the system. This implies that the decision to request and authorize the production of a new part at a stage is based on local stage information, in the case of a conventional installation kanban system, and global information from all downstream stages, in the case of an echelon kanban control system. This constitutes a potential advantage of the echelon kanban control system over the conventional installation kanban system. Moreover, the echelon kanban control system, just like the installation kanban system, depends on only one parameter per stage, the number of echelon kanbans, and is therefore simpler to optimize and implement than more complicated kanban-type systems that depend of two parameters per stage, such as the generalized kanban control system [15] and the extended kanban control system [23]. These two apparent advantages of the echelon kanban control system motivated our effort to develop an approximation method for its performance evaluation.

The approximation method for the performance evaluation of the echelon kanban control system that we develop is based on modeling the system as a queuing network with synchronization mechanisms. By exchanging the roles of the jobs and the resources (echelon kanbans) we then obtain an equivalent multi-class, nested, closed queuing network in which the population of each class is equal to the capacity or number of echelon kanbans of each echelon, i.e. a particular stage and all the downstream stages. Next, we decompose the original echelon kanban control system into a set of nested subsystems, each subsystem being

associated with a particular echelon. This means that we have as many subsystems as the number of the stages. Each subsystem is analyzed in isolation using Marie's method [54]. Each subsystem interacts with its neighboring subsystems in that it includes its downstream subsystem in the form of a single-station server with load-dependent exponential service rates, and it receives external arrivals from its upstream subsystem. A fixed-point iterative procedure is then used to determine the unknown parameters of each subsystem by taking into account the interactions between neighboring subsystems.

The rest of this chapter is organized as follows. In Section 3.2, we describe the exact operation of an echelon kanban control system by means of a simple example. In Section 3.3 we present a queuing network model of an echelon kanban control system and the performance measures of the network that we are interested in evaluation. We also introduce the decomposition-based approximation method that we develop for the performance evaluation of the echelon kanban control system. In Section 3.4, we present the decomposition scheme that we use for the performance evaluation. In Section 3.5, we present the analysis in isolation of each subsystem of the decomposition, and in Section 3.6 we present the analysis of the entire system. In Section 3.7, we present numerical results on the influence and optimization of the parameters, and in Section 3.8 we draw conclusions.

3.2 Description of an Echelon Kanban Control System

In this section, we give a precise description of the operation of an echelon kanban control system by means of a simple example. In this example, we consider a production system that consists of $M = 9$ machines in series labeled M1 to M9, produces a single part type, and does not involve batching, reworking or scrapping of parts. On each machine, a

certain amount of time, which may be constant or variable, is required to process a part. All parts visit successively machines M1 to M9 (see Figure 3-1).



Figure 3-1: Example of a serial production system.

The production system is decomposed into $N = 3$ stages. Each stage is a production-inventory system consisting of a manufacturing process and an output buffer. The output buffer of each stage stores the finished parts of the stage. The manufacturing process at each stage consists of a subset of machines of the original manufacturing system and contains parts that are in service or waiting for service on the machines. These parts represent the WIP of the stage and are used to supply the output buffer. In the example, each stage consists of three machines. In particular, the sets of machines $\{M1, M2, M3\}$, $\{M4, M5, M6\}$ and $\{M7, M8, M9\}$ belong to stages 1, 2 and 3, respectively. The decomposition of the production system into three stages is illustrated in Figure 3-2.

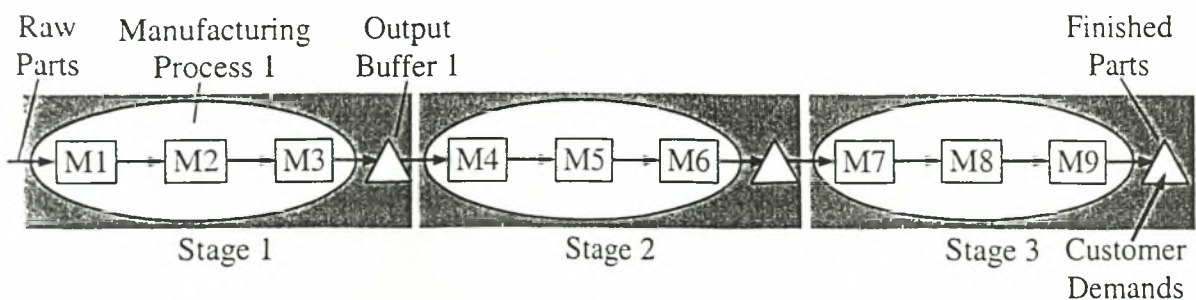


Figure 3-2: Production system of Figure 3-1 decomposed into three stages in series.

Each stage has associated with it a number of echelon kanbans that request and authorize the release of parts into the stage and trace a closed path through the stage and all the downstream stages. The number of echelon kanbans of stage i is fixed and equal to K_i .

There must be at least one kanban of stage i available in order to release a new part into that stage. In this case, the kanban is attached onto the part and follows it through the system until the last output buffer.

Parts that are in the output buffer of stage 3 are the finished parts of the production system. These parts are used to satisfy the customer demands. When a customer demand arrives to the system, a request for the release of a finished part from the output buffer to the customer is placed. If there is at least one finished part, the customer demand is immediately satisfied, otherwise the demand cannot be immediately satisfied and so it is backordered until a finished part is available. If a part is available, it is released to the customer after releasing the kanbans of all the stages (1, 2, 3) that were attached to it. These kanbans are then transferred upstream, each kanban to its corresponding stage. The kanban of stage i carries with it a demand for the production of a new stage- i finished part and authorizes the release of a finished part from stage $i-1$ (waiting in the output buffer) to stage i . When a finished part of stage $i-1$ is transferred to stage i , the stage- i kanban is attached to it on top of the kanbans of stages 1 to $i-1$, which have already been attached to the part at previous stages. This implies that

$$K_i \geq K_{i+1}, i = 1, \dots, N-1. \quad (3.1)$$

In summary, in an echelon kanban control system, a customer demand that arrives to the system is immediately transferred from the last stage to all upstream stages as soon as the demand is satisfied. Moreover, since an echelon kanban of stage i is attached to every part in stages i to N , the number of parts in stages i to N is limited by K_i . This is an advantageous consequence of the operation of the echelon kanban control system. A potential disadvantage is that the system may not immediately respond to demand, since a customer demand may not

be immediately transferred to all stages upon its arrival to the system, if a finished part is not available.

The example presented above concerns a basic echelon kanban control system with the following assumptions: a) There is an infinite number of raw parts at the input of the production system; b) when a kanban of stage i is detached from a part at the output buffer of stage N , it is immediately transferred to stage i carrying along with it a demand and an authorization for the production of a new stage- i finished part; c) if there are no stage- N finished parts at the time of a customer demand arrival, the customer demand is backordered; d) the N stages are in series; e) there is a single class of parts; f) each part has attached on it a specific number of kanbans, as we described above, depending on the stage in which his processing occurs. Although in the rest of the chapter the approximate method for performance evaluation will concern this type of productions systems, the method can be extended to more general classes of echelon kanban control systems.

3.3 Queuing Network Model of an Echelon Kanban Control System

In this section, we present a queuing network model of an echelon kanban control system, the performance measures of the network that we are interested in evaluating, and the methodology that we use in order to develop an approximation method for the performance evaluation of the system.

3.3.1 Modeling

In order to develop the approximation method for the performance evaluation of the echelon kanban control system we first model the system as a queuing network with synchronization mechanisms. Figure 3-3 shows the queuing network model of the echelon

kanban control system with three stages in series considered in Section 3.2. The manufacturing process of each stage is modeled as a subnetwork in which the machines of the manufacturing process are represented by single-server stations. The subnetwork associated with the manufacturing process of stage i is denoted by I_i , and the single-server stations representing machines $M1, \dots, M9$ are denoted S_1, \dots, S_9 , respectively. The number of stations of subnetwork I_i is denoted by m_i . In the example, $m_i = 3$, $i = 1, 2, 3$. The echelon kanban control mechanism is modeled via three synchronization stations, denoted J_i , at the output of each stage i , $i = 1, 2, 3$.

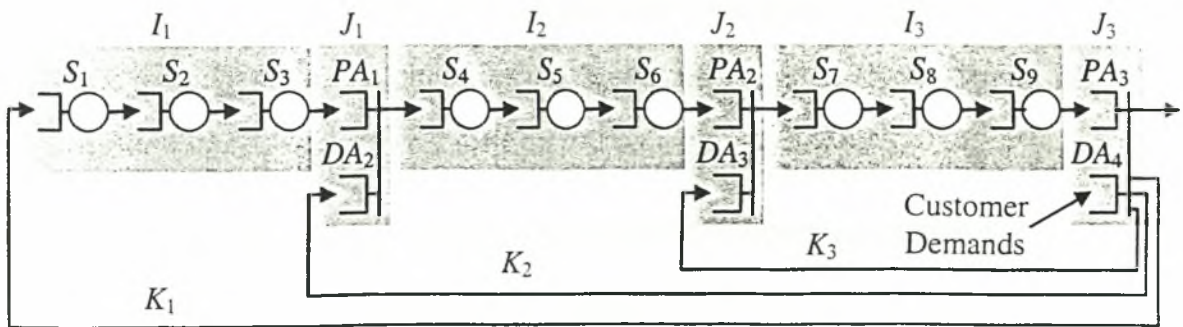


Figure 3-3: Queuing network model of the echelon kanban control system of Figure 3-2.

A synchronization station is a tool often used to model assembly operations in queuing networks. It consists of a server with instant service times. The server is fed by two or more queues (in our case by two). When there is at least one customer in each of the queues that feed the server, these customers move instantly through and out of the server. This implies that at any time at least one of the queues that feed the server is empty. Customers that enter the server immediately exit the server after possibly having been separated into more or joined into fewer customers. In our case, the queues in a synchronization station contain parts and demands which are combined with kanbans.

To illustrate this, let us first focus on any synchronization station J_i but that of the last stage. This synchronization station represents the synchronization of a finished part of stage i and a free kanban of stage $i+1$. Let PA_i and DA_{i+1} denote the two upstream queues of the synchronization station J_i . Queue PA_i represents the output buffer of stage i and contains stage- i finished parts, each of which has attached to it kanbans from stages 1 to i . Queue DA_{i+1} contains customer demands for the production of new stage- $(i+1)$ parts, each of which has attached to it a kanban of stage $i+1$. The synchronization station operates as follows. As soon as there is one entity in each one of the queues PA_i and DA_{i+1} , the finished part of stage i engages the stage- $(i+1)$ kanban without releasing the kanbans from stages 1 to i , and joins the first station of stage $i+1$ (i.e. a part of stage i is consumed by stage $i+1$). Note that at stage 1, as soon as a stage-1 kanban is available, a new part is immediately released into stage 1 since there are always raw parts at the input of the system.

Let us now consider the last synchronization station J_N (J_3 in the example). This synchronization station synchronizes queues PA_N , and D_{N+1} . Queue PA_N represents the output buffer of stage N and contains stage- N finished parts each of which has attached to it kanbans from stages 1 to N . Queue D_{N+1} contains customer demands. When a customer demand arrives to the system, it joins D_{N+1} , thereby requesting the release of a finished part from PA_N to the customer. If there is a finished part in queue PA_N , it is released to the customer and the demand is satisfied. In this case, the finished part in PA_N releases the kanbans that were attached to it and these kanbans are transferred upstream to queues DA_i ($i = 1, 2, \dots, N$), each kanban carrying along with it a demand for the production of a new stage- i ($i = 1, 2, \dots, N$) finished part and authorizing the release of a finished part from queue PA_{i-1} into stage i . If there is not an entity in queue PA_N , the customer demand is put on hold as a backordered demand.

An important special case of the echelon kanban control system is that in which there are always customer demands for finished parts. This case is known as the *saturated* echelon kanban control system. Its importance lies in the fact that its throughput determines the maximum capacity of the original system. In the saturated system, when there are finished parts at stage N , they are immediately consumed and equal number of parts enters the system. As far as the queuing network corresponding to this model is concerned, the synchronization station J_N can be removed since queue D_{N+1} is never empty and as a result it can be ignored. The resulting network is shown in Figure 3-4. In the saturated echelon kanban control system, when the processing of a part is completed at stage N , this part is immediately consumed after releasing the kanbans of stages $1, 2, \dots, N$, that were attached to it, and sending them back to queues DA_i ($i = 1, 2, \dots, N$).

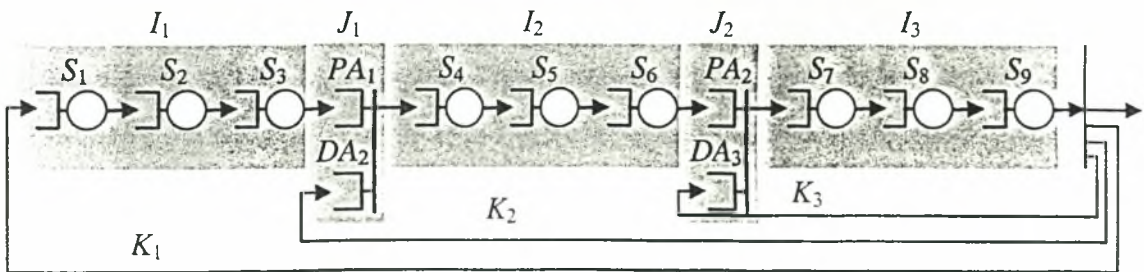


Figure 3-4: Queuing network model of the saturated echelon kanban control system associated with the queuing network of Figure 3-3.

Finally, it is worth noticing that an echelon kanban control system contains the CONWIP system [68] as a special case. In a CONWIP system, as soon as a finished product leaves the production system to be delivered to a customer, a new part enters the system to begin its processing. An echelon kanban control system with $K_1 \leq K_i, i \neq 1$ behaves exactly like a CONWIP system.

3.3.2 Performance Measures

The performance measures of an echelon kanban control system depend on the manufacturing processes, the arrival process of customer external demands, and the number of kanbans of each stage. The performance measures that are of particular interest are the average work in process (WIP) in each stage, the mean number of finished parts at each stage, the mean number of backordered (not immediately satisfied) demands, and the average waiting time and percentage of backordered demands. In the case of the saturated echelon kanban control system, the main performance measure is its production rate, P_r , i.e. the average number of finished products leaving the output buffer of stage N per unit of time. Given that the production rate or production capacity of the saturated system represents the maximum rate at which customer demands can be satisfied, the average arrival rate of external customer demands, say λ_D , must be strictly less than the production rate of the saturated system, in order for the system to meet all the demands in the long run. In other words, the stability condition for the system is

$$\lambda_D < P_r. \quad (3.2)$$

3.3.3 Methodology for the Analysis of an Echelon Kanban Control System

The approximation procedure for the performance evaluation of a multi-stage echelon kanban control system that we develop is based on decomposing the original multi-stage system into many nested single-stage subsystems and analyzing each system in isolation. The subsystems are nested in each other so that each subsystem includes its downstream subsystem in the form of a single-server station and receives external arrivals from its upstream subsystem. The first subsystem mimics the original system. To analyze each subsystem, we view it as a closed queuing network and we approximate each station of the

closed queuing network by an exponential service station with load-dependent service rates. The resulting network is then a product-form network. A fixed-point iterative procedure is then used to determine the unknown parameters of each subsystem by taking into account the interactions between neighboring subsystems. A detailed description of the steps of the approximation procedure is given in the Sections 3.4-3.6.

3.4 Decomposition of an Echelon Kanban Control System

The first step of the approximation procedure for the performance evaluation of a multi-stage echelon kanban control system is to decompose the original multi-stage system into many nested single-stage subsystems. Consider the queuing network model of an echelon kanban control system consisting of N stages in series as described in Section 3.3. (see Figure 3-3 for $N = 3$). Let us denote the queuing network by S . Our goal is to analyze S by decomposing it into a set of N nested subsystems S^i , $i = 1, \dots, N$, as follows.

Subsystem S^N is an open queuing network with restricted capacity consisting of 1) an upstream synchronization station, denoted by I^N , representing J_{N-1} in the original system, 2) the subnetwork of stations in the original system, I_N , and 3) a downstream synchronization station, denoted by O^N , representing J_N in the original system. Each subsystem S^i , $i = 2, 3, \dots, N-1$, is an open queuing network with restricted capacity consisting of 1) an upstream synchronization station, denoted by I^i , representing J_{i-1} in the original system, 2) the subnetwork of stations in the original system, I_i , and 3) a downstream single-server pseudo-station, denoted by \hat{S}_i , representing the part of the system downstream of I_i in the original system. Finally, subsystem S^1 is a closed queuing network consisting of 1) the subnetwork of stations in the original system, I_1 and 2) a downstream single-server pseudo-station, denoted by \hat{S}_1 , representing the part of the system downstream of I_1 in the original system. Notice that

pseudo-station \hat{S}_i in subsystem S^i , $i = 1, \dots, N - 1$, is an aggregate representation of subsystem S^{i+1} , which is nested inside subsystem S^i .

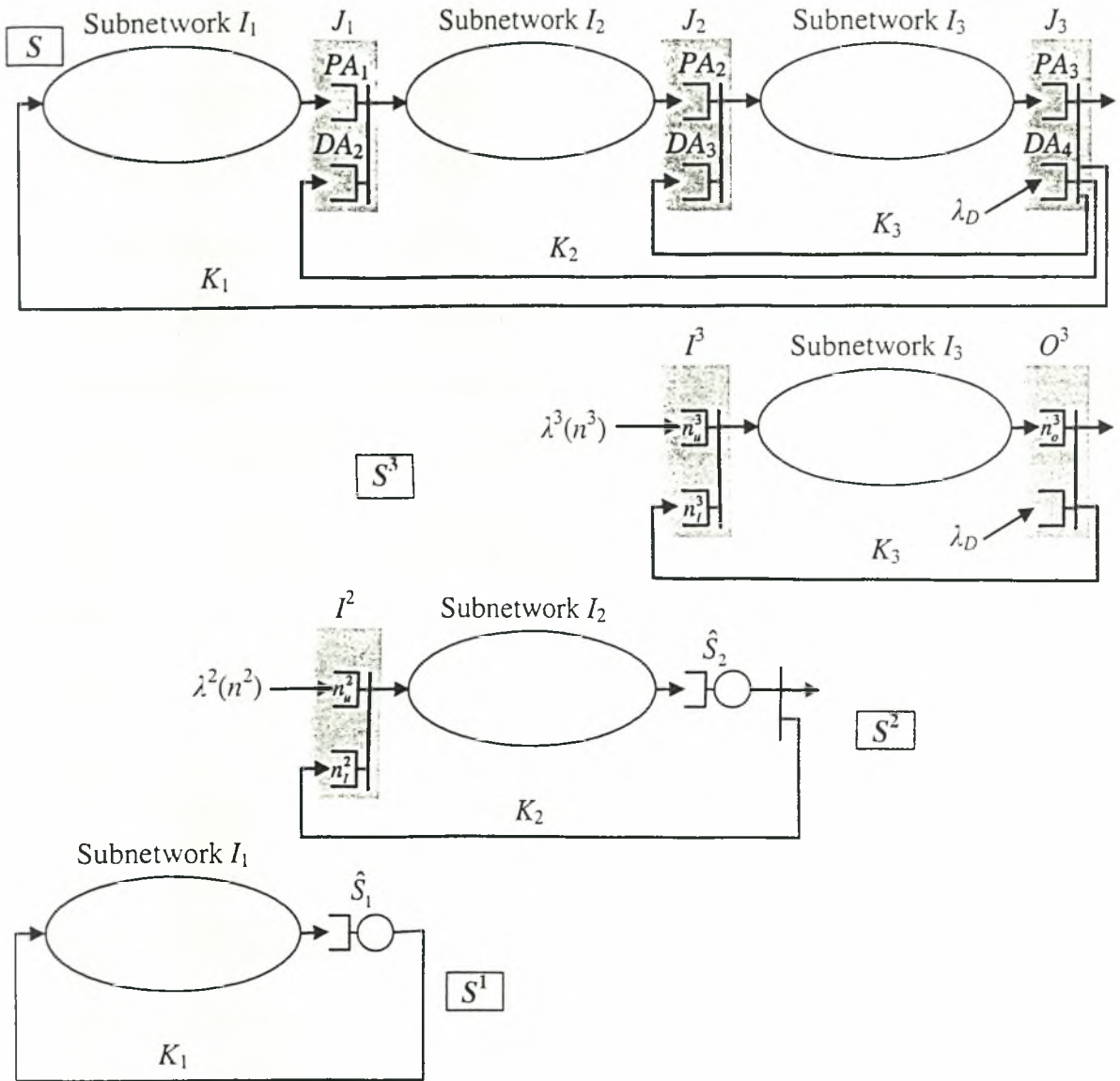


Figure 3-5: Illustration of the decomposition of a 3-stage echelon kanban control system.

The number of kanbans of subsystem S^i is K_i . Subsystem S^N is synchronized with two external arrival processes, one at synchronization station I^N , concerning parts that arrive from subnetwork I_{N-1} , and the other at synchronization station O^N , concerning customer demands. Subsystem S^i , $i = 2, \dots, N - 1$, is synchronized with only one external arrival process at

synchronization station I^i , concerning parts that arrive from subnetwork I_{i-1} . Subsystem S^i is a closed network; therefore it is not synchronized with any external arrival processes. The decomposition described above is illustrated in Figure 3-5 for an echelon kanban control system consisting of $N = 3$ stages. As can be seen, each synchronization station J_i of the original network S , linking stage i to stage $i+1$, is represented only once in the decomposition.

To completely characterize each subsystem S^i , $i = 2, \dots, N - 1$, we assume that each of the external arrival processes to S^i is a state-dependent Markovian process. Let $\lambda^i(n^i)$ denote the state-dependent arrival rate of stage- i raw parts at the upstream synchronization station I^i of subsystem S^i , where n^i is the state of subsystem S^i and is defined as the number of parts in this subsystem. Let Q_u^i and Q_l^i be the two queues of synchronization station I^i , containing n_u^i and n_l^i customers, respectively, where n_u^i is the number of finished parts of stage $i-1$ waiting to enter subnetwork I_i , and n_l^i is the number of free stage- i kanbans waiting to authorize the release of stage $i-1$ finished parts into subnetwork I_i . Then it is clear that the only possible states in the synchronization station are the states $(n_l^i, 0)$, for $n_l^i = 0, \dots, K_i$, and $(0, n_u^i)$, for $n_u^i = 0, \dots, K_{i-1} - K_i$. Then, the state n^i of subnetwork S^i can be simply obtained from n_u^i and n_l^i using the following relation:

$$n^i = \begin{cases} K_i - n_l^i & \text{if } n_l^i \neq 0, \\ K_i + n_u^i & \text{if } n_l^i = 0. \end{cases} \quad (3.3)$$

The above relation implies that $0 \leq n^i \leq K_{i-1}$. Also, since the number of raw parts at the input of stage i cannot be more than the number of kanbans of stage $i-1$, $\lambda^i(K_{i-1}) = 0$. In subsystem S^N , besides the arrival rate of stage- N raw parts at I^N , $\lambda^N(n^N)$, there is also the

arrival rate of customer demands at O^N , λ_D . Subsystem S^1 , as was mentioned above, is a closed network and therefore has no external arrival processes to define.

To obtain the performance of network S , the following two problems must be addressed: 1) How to analyze each subsystem S^i , $i = 1, \dots, N$, assuming that the external arrival rates are known (except in the case of the first subsystem S^1 , where there are no external arrival rates), and 2) how to determine the unknown external arrival rates. These two problems are addressed in Sections 3.5 and 3.6, respectively. Once these two problems have been solved, the performance of each stage of the original system S can be obtained from the performances of subsystems S^i , $i = 1, \dots, N$.

3.5 Analysis of Each Subsystem in Isolation

In this section, we describe how we analyze each subsystem in isolation using Marie's approximate analysis of general closed queuing networks [54]. Throughout this analysis, the state-dependent rates of the external arrival processes, $\lambda^i(n^i)$, $0 \leq n^i \leq K_{i-1}$, $i = 2, \dots, N$, are supposed to be known. To analyze each subsystem using Marie's method, we first view the subsystem as a closed queuing network. For subsystems S^i , $i = 2, \dots, N$, this is done by considering the kanbans of stage i as the customers of the closed network, and the stage- i raw parts and the demands (in the case of the last subsystem S^N) as external resources. Note that the queuing network associated with subsystem S^1 is already being modeled as a closed queuing network in the decomposition. Its customers are the kanbans of stage 1.

The closed queuing network associated with subsystem S^N is partitioned into $m_N + 2$ stations, namely the synchronization stations I^N and O^N and the m_N stations of subnetwork I_N . Similarly, the closed queuing network associated with each subsystem S^i is partitioned into $m_i + 2$ stations, namely the synchronization station I^i , the m_i stations of subnetwork I_i , and station

\hat{S}_i . Finally, the closed queuing network associated with subsystem S^1 is partitioned into $m_1 + 1$ stations, namely the m_1 stations of subnetwork I_1 , and station \hat{S}_1 . Each station is approximated by an exponential service station with load-dependent service rates. The resulting network associated with each subsystem is then a product-form network [35] consisting of K_i customers and $m_i + 2$ stations for subsystem S^i , $i = 2, \dots, N$, or $m_1 + 1$ stations for subsystem S^1 . The stations within each subsystem S^i , $i = 1, \dots, N$, will be denoted by the index $k \in M_i$, where $M_1 = \{1, \dots, m_1, \hat{S}\}$, $M_i = \{I, 1, \dots, m_i, \hat{S}\}$ for $i = 2, \dots, N-1$, and $M_N = \{I, 1, \dots, m_N, O\}$, where I , O and \hat{S} denote the upstream synchronization station, downstream synchronization station and pseudo-station in each subsystem, respectively. Let $\mu_k^i(n_k^i)$ denote the load-dependent service rate of station k in subsystem S^i when n_k^i customers are present at the station. The resulting equivalent network is then a Gordon-Newell network [35]. In the following subsection, we show how to determine $\mu_k^i(n_k^i)$, $n_k^i = 1, \dots, K_i$, for each station $k \in M_i$ within each subsystem S^i , $i = 1, \dots, N$.

3.5.1 Determination of the Load-Dependent Service Rates in Each Subsystem

In this subsection, we show how to determine the unknown load-dependent service rates $\mu_k^i(n_k^i)$, $n_k^i = 1, \dots, K_i$, for each station $k \in M_i$ in any subsystem S^i , $i = 1, \dots, N$. Since, we will be dealing with a single subsystem, for the sake of notational simplicity we will drop the index i to denote variables associated with subsystem i .

Let vector $\mathbf{n} = (n_1, \dots, n_k, \dots, n_M)$ be the state of the network, where n_k denotes the number of customers present at station k . Then, the probability of being in stage \mathbf{n} , $P(\mathbf{n})$, is given by the following product-form solution:

$$P(\mathbf{n}) = \frac{1}{G(K)} \prod_{k \in M} \left[\prod_{n=1}^{n_k} \frac{V_k}{\mu_k(n)} \right], \quad (3.4)$$

where V_k is the average visit ratio of station k in the original system and is given from the routing matrix of the original system [5], and $G(K)$ is the normalization constant.

To determine the unknown parameters $\mu_k(n_k)$ in the product-form solution (3.4), each station is analyzed in isolation as an open system with a state-dependent Poisson arrival process whose rate $\lambda_k(n_k)$ depends on the total number of customers n_k present in the station. Let T_k denote this open system. Assume first that the rates $\lambda_k(n_k)$ are known for $n_k = 1, \dots, K - 1$. The queuing system T_k can then be analyzed using any appropriate technique to obtain the steady-state probabilities of having n_k customers in the isolated system, say $P_k(n_k)$. The issue of analyzing each queuing system T_k will be discussed in Subsection 3.5.2. The conditional throughput of this isolated system with population n_k , $v_k(n_k)$, can then be derived using the relation,

$$v_k(n_k) = \lambda_k(n_k - 1) \frac{P_k(n_k - 1)}{P_k(n_k)}, \text{ for } n_k = 1, \dots, K. \quad (3.5)$$

The load-dependent service rates of the k -th station of the equivalent closed network are then set equal to the conditional throughputs of the corresponding station in isolation, i.e.:

$$\mu_k(n_k) = v_k(n_k), \text{ for } n_k = 1, \dots, K. \quad (3.6)$$

Once the rates $\mu_k(n_k)$ have been obtained, the state-dependent arrival rates $\lambda_k(n_k)$ can be obtained from the generalized product-form solution as [5], [24], [54]:

$$\lambda_k(n_k) = V_k \frac{G_k(K - n_k - 1)}{G_k(K - n_k)}, \text{ for } n_k = 1, \dots, K - 1, \quad (3.7)$$

where $G_k(n)$ is the normalization constant of the equivalent network with station k removed (complementary network) and population n . $G_k(n)$ is a function of the parameters $\mu_k(n_k)$ for all $k' \neq k$ and $n_{k'} = 1, \dots, K$, and can be efficiently computed using any computational algorithm for product-form networks [18], [14]. An iterative procedure can then be used to determine these unknown quantities. This procedure is described by the following algorithm.

Algorithm 1: Analysis of a Subsystem in Isolation

Step 0: (Initialization) Set $\mu_k(n_k)$ to some initial value, for $k \in M$ and $n_k = 1, \dots, K$.

Step 1: For $k \in M$:

Calculate the state-dependent arrival rates $\lambda_k(n_k)$, for $n_k = 0, \dots, K - 1$, using (3.7).

Step 2: For $k \in M$:

- a. Analyze the open queuing system T_k .
- b. Derive the steady state probabilities $P_k(n_k)$ of having n_k customers, for $n_k = 1, \dots, K$.
- c. Calculate the conditional throughputs $v_k(n_k)$ using (3.5).

Step 3: For $k \in M$:

Set the load-dependent service rates of station k in the equivalent product-form network to $\mu_k(n_k) = v_k(n_k)$ for $n_k = 1, \dots, K$.

Step 4: Go to Step 1 until convergence of the parameters $\mu_k(n_k)$.

3.5.2 Analysis of the Open Queuing Systems

In this section we reintroduce index i denoting the subsystem. Step 2a of Algorithm 1 above requires the analysis of the open queuing systems T_k^i for $k \in M_i$ and $i = 1, \dots, N$. There are four different types of queuing systems: 1) the synchronization station O^N in subsystem S^N , 2) the synchronization stations I^i in subsystems S^i , $i = 2, \dots, N$, 3) the m_i stations in each subnetwork I_i , $i = 1, \dots, N$, and 4) the pseudo-stations \hat{S}_i in subsystems S^i , $i = 1, \dots, N - 1$.

Consider first the analysis of synchronization station O^N in subsystem S^N . O^N is a synchronization station fed by a Markovian arrival process with state-dependent rates, $\lambda_o^N(n_o^N)$, $0 \leq n_o^N \leq K_N$, and an external Poisson process with fixed rate λ_D . An exact solution of this system is easy to obtain by solving the underlying Markov chain. The steady-state probabilities $P_o^N(n_o^N)$ of having n_o^N customers in subsystem O^N can then be derived, and therefore the conditional throughput $v_o^N(n_o^N)$ can be estimated using (3.5) (see [21] and Appendix 3A).

The synchronization station I^i in each subsystem S^i , $i = 2, \dots, N$, is a synchronization station fed by two Markovian arrival processes with state-dependent rates, $\lambda_i^i(n_i^i)$, $0 \leq n_i^i \leq K_i$, and $\lambda^{i-1}(n^{i-1})$, $0 \leq n^{i-1} \leq K_{i-1}$. An exact solution of this system is also easy to obtain by solving the underlying Markov chain. (see [6] and Appendix 3B).

The analysis in isolation of any station $k \in \{1, \dots, m_i\}$ in each subnetwork I_i , $i = 1, \dots, N$, reduces to the analysis of a $\lambda_k^i(n_k^i)/G_i/1/N$ queue. Thus, classical methods can be used to analyze this queue to obtain the steady-state probabilities $P_k^i(n_k^i)$. For instance, if the service time distribution is Coxian [20], the algorithms given in [55] may be used. For multiple-server stations we can use the numerical technique presented in [69]. The conditional throughput

$v_k^i(n_k^i)$ can then be derived from the state probabilities using (3.5). In the special case where the service time is exponentially distributed, the conditional throughput $v_k^i(n_k^i)$ is simply equal to the load-dependent service rate $\mu_k^i(n_k^i)$ [24].

Finally, as was mentioned earlier, the pseudo-station \hat{S}_i in subsystem S^i , $i = 1, \dots, N - 1$, is an aggregate representation of subsystem S^{i+1} , which is nested inside subsystem S^i . Therefore, the conditional throughput of pseudo-station \hat{S}_i , $v_s^i(n_s^i)$, is set equal to the conditional throughput of subsystem S^{i+1} , $v^{i+1}(n^{i+1})$ [6]. The conditional throughput $v^i(n^i)$ of any subsystem S^i , $i = 2, \dots, N$, can be estimated by the following simple expression [6]:

$$v^i(n^i) = \begin{cases} \lambda_1(K_i - n^i) & \text{for } 1 \leq n^i \leq K_i, \\ \lambda_1(0) & \text{for } K_i \leq n^i \leq K_{i-1}. \end{cases} \quad (3.8)$$

3.6 Analysis of the Entire Echelon Kanban Control System

In Section 3.5 we analyzed each subsystem of the decomposition in isolation, given that the arrival rates of the external arrival processes were known. In this section, we address the problem of determining these arrival rates.

3.6.1 Equations of the Decomposition

Consider again network S in Figure 3-5, which has been decomposed into N subsystems. In each subsystem S^i , $i = 2, 3, \dots, N$, the unknown parameters involved in the decomposition are the arrival rates of raw parts at each upstream synchronization station I^i , $\lambda^i(n^i)$, $0 \leq n^i \leq K_{i-1}$. Recall that pseudo-station \hat{S}_{i-1} in subsystem S^{i-1} represents subsystem S^i , $i = 2, 3, \dots, N$. Therefore, the external arrival process of raw parts at synchronization station I^i in subsystem S^i should be identical to the arrival process of parts at pseudo-station \hat{S}_{i-1} in

subsystem S^{i-1} . This latter process is involved in the analysis of subsystem S^{i-1} in isolation and has been characterized by a state-dependent Poisson arrival process with rate $\lambda_s^{i-1}(n_s^{i-1})$, $0 \leq n_s^{i-1} \leq K_{i-1}$. As a result, the following set of equations holds:

$$\lambda_u^i(n) = \lambda_s^{i-1}(n) \text{ for } 0 \leq n \leq K_{i-1} \text{ and } i = 2, 3, \dots, N. \quad (3.9)$$

Therefore, the unknown parameters $\lambda^i(n^i)$ are the solution of a fixed-point problem. In Subsection 3.6.2 we propose an iterative procedure to determine these quantities.

3.6.2 Iterative Procedure for Determining the Unknown Parameters

The iterative procedure for determining the unknown parameters $\lambda^i(n^i)$ is described in Algorithm 2 given below. Algorithm 2 consists of several forward and backward steps. A forward step from subsystem S^{i-1} to S^i uses new estimates of the arrival rates to the upstream synchronization station I^i of subsystem S^i , to resolve S^i using Algorithm 1. A backward step from S^i to S^{i-1} solves S^{i-1} using Algorithm 1, given that the arrival rates to the upstream synchronization station I^j of each subsystem S^j , $j = i, \dots, N$, have converged. The procedure starts with subsystem S^N and moves backwards until it reaches subsystem S^1 . Subsystem S^N is first analyzed using Algorithm 1 and current estimates of $\lambda^N(n^N)$. This yields the conditional throughput of S^N , $v^N(n^N)$, which is needed to analyze subsystem S^{N-1} , as it determines the load-dependent exponential service rates of pseudo-station \hat{S}_{N-1} . Subsystem S^{N-1} is then analyzed using Algorithm 1 and current estimates of $\lambda^{N-1}(n^{N-1})$. This yields the conditional throughput of S^{N-1} , $v^{N-1}(n^{N-1})$, and the arrival rates to the pseudo-station \hat{S}_{N-1} , $\lambda_s^{N-1}(n_s^{N-1})$. If these arrival rates are not equal to the current estimates of the arrival rates $\lambda^N(n^N)$, then the latter rates have not converged. In this case, the current estimates of $\lambda^N(n^N)$ are updated to $\lambda_s^{N-1}(n_s^{N-1})$ and subsystem S^N is analyzed again using Algorithm 1 with the new estimates. Otherwise, the

arrival rates $\lambda^N(n^N)$ have converged and the procedure moves on to the analysis of subsystem S^{N-2} using Algorithm 1, where the load-dependent exponential service rates of pseudo-station \hat{S}_{N-2} are set equal to $\nu^{N-1}(n^{N-1})$. This procedure is repeated for subsystems S^{N-2}, S^{N-3}, \dots , until the first subsystem, S^1 , is reached and all the arrival rates $\lambda^i(n^i), i = 2, 3, \dots, N$, have converged. All the performance parameters of interest can then be derived.

Algorithm 2: Analysis of a Multistage Echelon Kanban Control System

Step 0: (Initialization) Set the unknown arrival rates of each subsystem S^i to some initial values, e.g. $\lambda^i(n^i) = \lambda_D, 0 \leq n^i \leq K_{i-1}$, and $i = 2, \dots, N$.

Step 1: Computation and convergence of the arrival rates, $\lambda^i(n^i), i = 2, \dots, N$.

Set $i = N$

While $i \geq 1$

 If $i = N$

 Solve subsystem S^N using Algorithm 1 and calculate the throughput $\nu^N(n^N), n^N = 1, \dots, K_{N-1}$, from (3.8).

 Set $i = i - 1$.

 Else

 Solve subsystem S^i using Algorithm 1 and calculate the arrival rate $\lambda_s^i(n_s^i), n_s^i = 0, \dots, K_i$, and the throughput $\nu^i(n^i), n^i = 1, \dots, K_{i-1}$, from (3.8).

 If $\lambda^{i+1}(n) = \lambda_s^i(n), n = 0, \dots, K_i$,

 Set $i = i - 1$

Else

Set $\lambda^{i+1}(n) = \lambda_s^i(n)$, $n = 0, \dots, K_i$ and set $i = i + 1$

Endif

Endif

Endwhile

In the case of the saturated echelon kanban control system, we can use the same algorithm. The only difference is in the analysis of subsystem S^N in Algorithm 1, where there is no downstream synchronization station O^N . As far as the convergence property of the algorithm is concerned, in all of the numerical examples that we examined (see Section 3.7), Algorithm 2 converged. The convergence criterion was that for every unknown parameter, the relative difference between its values at two consecutive iterations should be less than 10^{-4} .

Once Algorithm 2 has converged, all the performance parameters of the system can be calculated. Indeed, from the analysis of each subsystem S^i using Algorithm 1, it is possible to derive the performance parameters of stage i in the original network S , especially the throughput and the average length of each queue, including the queues of the synchronization stations. Thus, in the case of the saturated echelon kanban system we can derive the throughput, the average WIP, the average number of finished parts, and the average number of free echelon kanbans for each stage. In the case of an echelon kanban control system with external demands, some other important performance measures can be derived from the analysis of subsystem S^N , namely, the proportion of backordered demands, p_B , the average number of backordered demands, Q_D , and the mean waiting time of a backordered demand, W_B . These performance measures can be derived as follows [21], [28]:

$$P_B = P_O^N(0), \quad Q_D = P_O^N(0) \frac{1}{\frac{\lambda_O^N(0)}{\lambda_D} - 1}, \quad W_B = \frac{Q_D}{P_B \lambda_D},$$

where $\lambda_O^N(0)$ is the arrival rate of finished parts at synchronization station O^N when there are no finished parts at that station and $P_O^N(0)$ is the steady-state probability of having no finished part at synchronization station O^N .

3.7 Numerical Results

In this section, we test the approximation method for the performance evaluation of echelon kanban control systems developed in the previous sections on several numerical examples. Specifically, in Subsection 3.7.1, we study the accuracy and rapidity of the method as well as the influence of some key parameters of the echelon kanban control system on system performance. In Subsection 3.7.2, we use the method to optimize the design parameters (echelon kanbans) of the echelon kanban control system.

3.7.1 Influence of Parameters

In this subsection, we test the accuracy and rapidity of the approximation method for the performance evaluation of echelon kanban control systems developed in this chapter with two numerical examples in which we vary the number of stages, the number of kanbans in each stage, and the service-time distributions of the manufacturing process of each stage. In each example, we compare the performance of the system obtained by our approximation method to that obtained by simulation. We also compare the performance of the echelon kanban control system obtained by our approximation method to the performance of the conventional or installation kanban control system obtained by a similar approximation

method developed in [28] and by simulation. For each example, we consider first the case of the saturated system and then the case of the system with external demands.

Example 1

In example 1 we consider an echelon kanban system composed of N identical stages, where each stage contains a single machine with exponentially distributed service time with mean equal to 1. In order to compare the echelon kanban control system to the conventional kanban control system, we first set the number of installation kanbans of each stage i in the conventional kanban system, K_i^c , equal to some constant K , i.e. $K_i^c = K$. Then, we set the number of echelon kanbans of each stage i in the echelon kanban system, K_i^e , equal to the sum of the installation kanbans of stages i, \dots, N , in the conventional kanban system, i.e.

$$K_i^e = \sum_{j=i}^N K_j^c = (N + 1 - i)K.$$

For the case of the saturated system, the main performance parameter of interest is the throughput of the system, which determines the production capacity of the system. Table 3-1 shows the throughput of the saturated echelon kanban control system obtained by the approximation method and by simulation, for different values of N and K . The same table also shows the 95% confidence interval for the simulation results, the percentage of relative error of the approximation method with respect to simulation, and the number of iterations of Algorithm 2 that are needed to reach convergence. Table 3-2 shows the same results for the conventional kanban control system obtained in [28].

From the results in Table 3-1, we note that the number of iterations of Algorithm 2 of the approximation method increases with the number of stages, as is expected. Specifically, for $N = 3, 5$ and 10 , we have 7, 16, and 56 iterations of Algorithm 2, respectively. As far as

the convergence of Algorithm 1 is concerned, subsystem S^N requires 2 iterations of Algorithm 1, subsystem S^1 requires one iteration, whereas all other subsystems require 3 iterations irrespectively of the number of stages N . The simulation time is extremely long (between 2-7 hours for a total number of 68.000.000 parts produced) when compared to the time required for the approximation method, which is approximately 1-10 seconds.

Configuration	Simulation		Approximation		
	Production Capacity	Confidence Interval	Production Capacity	Relative Error	Iterations
1.1: $N = 3; K = 1$	0.581	$\pm 0.1\%$	0.571	- 1.8%	7
1.2: $N = 3; K = 3$	0.809	$\pm 0.1\%$	0.804	- 0.6%	7
1.3: $N = 3; K = 5$	0,877	$\pm 0.2\%$	0.873	- 0.5%	7
1.4: $N = 3; K = 10$	0.934	$\pm 0.5\%$	0.933	- 0.1%	7
1.5: $N = 3; K = 15$	0.955	$\pm 0.6\%$	0.954	- 0.1%	7
1.6: $N = 5; K = 1$	0.522	$\pm 0.0009\%$	0.502	- 4%	16
1.7: $N = 5; K = 3$	0,772	$\pm 0.1\%$	0.761	- 1.4%	16
1.8: $N = 5; K = 5$	0.85	$\pm 0.1\%$	0.843	- 0.8%	16
1.9: $N = 5; K = 10$	0.919	$\pm 0.2\%$	0.916	- 0.3%	16
1.10: $N = 5; K = 15$	0.945	$\pm 0.0009\%$	0.942	- 0.3%	16
1.11: $N = 10; K = 1$	0.485	$\pm 0.0007\%$	0.456	- 6.4%	56
1.12: $N = 10; K = 3$	0.745	$\pm 0.5\%$	0.730	- 2.1%	56
1.13: $N = 10; K = 5$	0,831	$\pm 0.7\%$	0.820	- 1.3%	56
1.14: $N = 10; K = 10$	0.908	$\pm 0.1\%$	0.902	- 0.7%	56
1.15: $N = 10; K = 15$	0.937	$\pm 0.1\%$	0.933	- 0.4%	56

Table 3-1: Production capacity of the saturated echelon kanban control system (Example 1).

From Table 3-1 we note that as the number of echelon kanbans increases, for a given number of stages N , the throughput also increases and asymptotically tends to the production rate of each machine in isolation. Moreover, the throughput seems to be decreasing in the number of stages. The results obtained by the approximation method are fairly accurate when compared to the simulation results. The relative error is very small, and only for the cases where K is very low, especially $K = 1$, we observe significant errors. This happens because when the number of echelon kanbans is small, there are strong dependence phenomena among

stations and these phenomena are not well captured by the state-depended Markovian arrival processes assumed in the decomposition method. Comparing the results between Table 3-1 and Table 3-2, we note that the production capacity of the echelon kanban control system is always higher than that of the conventional kanban controlled system, given that the two systems have the same value of K .

Configuration	Simulation		Approximation		
	Production Capacity	Confidence Interval	Production Capacity	Relative Error	Iterations
1.1: $N = 3; K = 1$	0.562	$\pm 0.5\%$	0.547	- 2.7%	2
1.2: $N = 3; K = 3$	0.800	$\pm 0.7\%$	0.792	- 1.0%	2
1.3: $N = 3; K = 5$	0.869	$\pm 1.3\%$	0.865	- 0.5%	2
1.4: $N = 3; K = 10$	0.926	$\pm 0.8\%$	0.928	+ 0.2%	2
1.5: $N = 3; K = 15$	0.952	$\pm 1.2\%$	0.951	- 0.1%	2
1.6: $N = 5; K = 1$	0.484	$\pm 0.6\%$	0.449	- 7.0%	4
1.7: $N = 5; K = 3$	0.746	$\pm 0.8\%$	0.731	- 2.0%	4
1.8: $N = 5; K = 5$	0.833	$\pm 0.8\%$	0.822	- 1.3%	4
1.9: $N = 5; K = 10$	0.901	$\pm 1.2\%$	0.904	+ 0.3%	4
1.10: $N = 5; K = 15$	0.943	$\pm 1.1\%$	0.934	- 0.9%	4
1.11: $N = 10; K = 1$	0.429	$\pm 0.5\%$	0.379	- 11.6%	7
1.12: $N = 10; K = 3$	0.704	$\pm 0.7\%$	0.680	- 3.4%	6
1.13: $N = 10; K = 5$	0.806	$\pm 0.9\%$	0.786	- 2.6%	5
1.14: $N = 10; K = 10$	0.855	$\pm 0.5\%$	0.883	- 3.2%	5
1.15: $N = 10; K = 15$	0.917	$\pm 1.3\%$	0.919	+ 0.2%	5

Table 3-2: Production capacity of the saturated conventional kanban system (Example 1).

For the system with backordered demands, the main performance parameters of interest are the proportion of backordered demands, p_B , the average number of backordered demands, Q_D , and the mean waiting time of a backordered demand, W_B , as defined at the end of Subsection 3.6.2. Table 3-3 shows these performance parameters obtained by the approximation method and by simulation, for the configurations of parameters 1.3, 1.8 and 1.13 of Table 3-1 and different values of the customer demand rate, λ_D . The same table also

shows the 95% confidence interval for the simulation results and the number of iterations of Algorithm 2 that are needed to reach convergence.

Table 3-4 shows the same results for the conventional kanban control system obtained in [28].

Configuration	Q_D	W_B	P_B (%)	Iterations
1.16: $N = 3; K = 5; \lambda_D = 0.1$				
Approximation	0.0	0.0	0.0	6
Simulation	0.0	0.0	0.0	
1.17: $N = 3; K = 5; \lambda_D = 0.5$				
Approximation	0.035	4.069	1.729	7
Simulation	0.034 ($\pm 0.9\%$)	2.066 ($\pm 1.2\%$)	3.337	
1.18: $N = 3; K = 5; \lambda_D = 0.625$				
Approximation	0.221	4.594	7.687	7
Simulation	0.213 ($\pm 0.1\%$)	3.014 ($\pm 14.2\%$)	11.32	
1.19: $N = 3; K = 5; \lambda_D = 0.8$				
Approximation	4.176	10.791	48.38	8
Simulation	4.095 ($\pm 3.6\%$)	9.755 ($\pm 7\%$)	52.47	
1.20: $N = 5; K = 5; \lambda_D = 0.1$				
Approximation	0.0	0.0	0.0	16
Simulation	0.0	0.0	0.0	
1.21: $N = 5; K = 5; \lambda_D = 0.5$				
Approximation	0.035	4.070	1.71	16
Simulation	0.032 ($\pm 0.007\%$)	3.189 ($\pm 0.003\%$)	2.03	
1.22: $N = 5; K = 5; \lambda_D = 0.8$				
Approximation	6.774	14.440	58.69	22
Simulation	6.5686 ($\pm 0.08\%$)	12.895 ($\pm 0.02\%$)	63.67	
1.23: $N = 10; K = 5; \lambda_D = 0.1$				
Approximation	0.0	0.0	0.0	20
Simulation	0.0	0.0	0.0	
1.24: $N = 10; K = 5; \lambda_D = 0.5$				
Approximation	0.035	4.070	1.72	39
Simulation	0.023 ($\pm 0.005\%$)	3.512 ($\pm 0.002\%$)	1.28	
1.25: $N = 10; K = 5; \lambda_D = 0.77$				
Approximation	3.817	10.709	46.3	61
Simulation	3.131 ($\pm 0.003\%$)	9.064 ($\pm 0.001\%$)	49.3	

Table 3-3: Average number of backordered demands, mean waiting time of a backordered demand and proportion of backordered demands for the echelon kanban control system (Example 1).

Configuration	Q_D	W_B	P_B (%)	Iterations
1.16: $N = 3; K = 5; \lambda_D = 0.1$				
Approximation	0.0	0.0	0.0	1
Simulation	0.0	0.0	0.0	
1.17: $N = 3; K = 5; \lambda_D = 0.5$				
Approximation	0.035	2.06	3.4	2
Simulation	0.033 ($\pm 30\%$)	2.16 ($\pm 17\%$)	3.1	
1.18: $N = 3; K = 5; \lambda_D = 0.625$				
Approximation	0.222	3.00	11.82	3
Simulation	0.230 ($\pm 17\%$)	3.26 ($\pm 15\%$)	11.78	
1.19: $N = 3; K = 5; \lambda_D = 0.8$				
Approximation	4.56	10.1	56.3	4
Simulation	4.26 ($\pm 19\%$)	10.3 ($\pm 13\%$)	52.1	
1.20: $N = 5; K = 5; \lambda_D = 0.1$				
Approximation	0.0	0.0	0.0	1
Simulation	0.0	0.0	0.0	
1.21: $N = 5; K = 5; \lambda_D = 0.5$				
Approximation	0.0353	2.07	3.40	2
Simulation	0.038 ($\pm 30\%$)	2.16 ($\pm 9\%$)	3.58	
1.22: $N = 5; K = 5; \lambda_D = 0.8$				
Approximation	11.26	19.3	73.0	7
Simulation	8.93 ($\pm 22\%$)	17.2 ($\pm 15\%$)	65.2	
1.23: $N = 10; K = 5; \lambda_D = 0.1$				
Approximation	0.0	0.0	0.0	1
Simulation	0.0	0.0	0.0	
1.24: $N = 10; K = 5; \lambda_D = 0.5$				
Approximation	0.0353	2.07	3.40	2
Simulation	0.0368 ($\pm 30\%$)	2.18 ($\pm 17\%$)	3.38	
1.25: $N = 10; K = 5; \lambda_D = 0.77$				
Approximation	6.89	13.9	64.2	11
Simulation	5.95 ($\pm 22\%$)	13.7 ($\pm 14\%$)	56.9	

Table 3-4: Average number of backordered demands, mean waiting time of a backordered demand and proportion of backordered demands for the conventional kanban control system (Example 1).

From the results in Table 3-3 we note that as the customer demand arrival rate increases, the number of iterations of Algorithm 2 also increases, although not dramatically. As far as the average number of backordered demands, Q_D , is concerned, we note that the analytical method is fairly accurate. This is not true for the mean waiting time of a backordered demand, W_B , where in some cases the difference between the approximation

method and simulation are significant. Comparing the results between Table 3-3 and Table 3-4, we note that the echelon kanban control system always has a smaller average number of backordered demands, Q_D , than the conventional kanban control system, given that the two systems have the same value of K . This is particularly true when the two systems are highly loaded (i.e. λ_D is close to the production capacity).

Table 3-5 shows the results for the average number of finished parts (FP) and the average work-in-process (WIP) at each stage for the configurations of parameters 1.17 and 1.19 in Table 3-3. Table 3-6 shows the same results for the conventional kanban control system.

Configuration	Stage 1		Stage 2		Stage 3	
	WIP	FP	WIP	FP	WIP	FP
1.17: $N = 3; K = 5;$ $\lambda_D = 0.5$						
Simulation	0.988 ($\pm 0.1\%$)	4.039 ($\pm 0.09\%$)	0.978 ($\pm 0.1\%$)	4.022 ($\pm 0.1\%$)	0.961 ($\pm 0.1\%$)	4.011 ($\pm 0.1\%$)
Approximation	0.999	4.031	0.995	4.005	0.969	4.000
Error	+ 1.1%	- 0.2%	+ 1.7%	- 0.4%	+ 0.8%	- 0.3%
1.19: $N = 3; K = 5;$ $\lambda_D = 0.8$						
Simulation	3.363 ($\pm 0.5\%$)	2.392 ($\pm 0.3\%$)	3.068 ($\pm 0.3\%$)	2.018 ($\pm 0.3\%$)	2.589 ($\pm 0.3\%$)	1.569 ($\pm 0.5\%$)
Approximation	3.479	2.349	3.159	1.902	2.655	1.455
Error	+ 3.3%	- 1.8%	+ 2.9%	- 6.1%	+ 2.5%	- 7.8%

Table 3-5: Average work in process (WIP) and average number of finished parts (FP) in each stage for the echelon kanban control system (Example 1).

Comparing the results between Table 3-5 and Table 3-6, we note that the echelon kanban control system has slightly higher average WIP and lower FG inventory than the conventional kanban control system, when the two systems are highly loaded (i.e. λ_D is close to the production capacity), and given that the two systems have the same value of K . When the two systems are not highly loaded, the difference in average WIP and FG inventory

between them is very small. Finally, it appears that the difference in average WIP and FG inventory between the echelon kanban control system and the conventional kanban system is higher in upstream stages than in downstream stages.

Configuration	Stage 1		Stage 2		Stage 3	
	WIP	FP	WIP	FP	WIP	FP
1.17: $N = 3; K = 5;$ $\lambda_D = 0.5$						
Simulation	0.94 ($\pm 3.2\%$)	4.06 ($\pm 0.7\%$)	0.95 ($\pm 3.1\%$)	4.02 ($\pm 0.7\%$)	0.94 ($\pm 3.2\%$)	4.04 ($\pm 0.8\%$)
Approximation	0.97	4.03	0.97	4.01	0.97	4.00
Error	+ 3%	- 0.7%	+ 2%	- 0.2%	+ 3%	- 1%
1.19: $N = 3; K = 5;$ $\lambda_D = 0.8$						
Simulation	2.54 ($\pm 3.0\%$)	2.47 ($\pm 4.0\%$)	2.52 ($\pm 3.2\%$)	1.98 ($\pm 5.0\%$)	2.55 ($\pm 3.1\%$)	1.58 ($\pm 6.3\%$)
Approximation	2.61	2.38	2.58	1.85	2.66	1.40
Error	+ 2.7%	- 3.6%	+ 2.4%	- 6.5%	+ 4%	- 11%

Table 3-6: Average work in process (WIP) and average number of finished products (FP) in each stage for the conventional kanban system (Example 1).

Example 2

In Example 2 we consider an echelon kanban control system consisting of $N = 3$ identical stages, where each stage contains a single machine with identical service time distribution with mean equal to 1. The number of echelon kanbans at each stage is $K_1 = 15$, $K_2 = 10$ and $K_3 = 5$. Our goal is to investigate the influence of the variability of the service time on the performance of the above system. To this end, we consider three different distributions: a Coxian-2 distribution with squared coefficient of variation $cv^2 = 2.0$, an Erlang-2 distribution with $cv^2 = 0.5$, and an exponential distribution ($cv^2 = 1.0$). Table 3-7 shows the production capacity for the saturated echelon kanban control system obtained by the approximation method and by simulation, for the three different distributions. Table 3-8 shows the same results for the conventional kanban control system obtained in [28].

From the results in Table 3-7, we note that when the variability of the service time distribution increases, the production capacity decreases, as is expected. The results obtained by the approximation method are fairly accurate when compared to the simulation results. Comparing the results between Table 3-7 and Table 3-8, we note that for all the service-time distributions, the production capacity of the echelon kanban control system is higher than that of the conventional kanban control system. The results for the analytical solution and simulation for the case of the echelon kanban system with backordered demands is shown in Figure 3-6. In particular, we present the proportion of backordered demands p_B as a function of the arrival rate of demands λ_D for the three different service time distributions. It appears that as the cv^2 of the service time distribution increases, the difference between simulation and analytical results tends to increase.

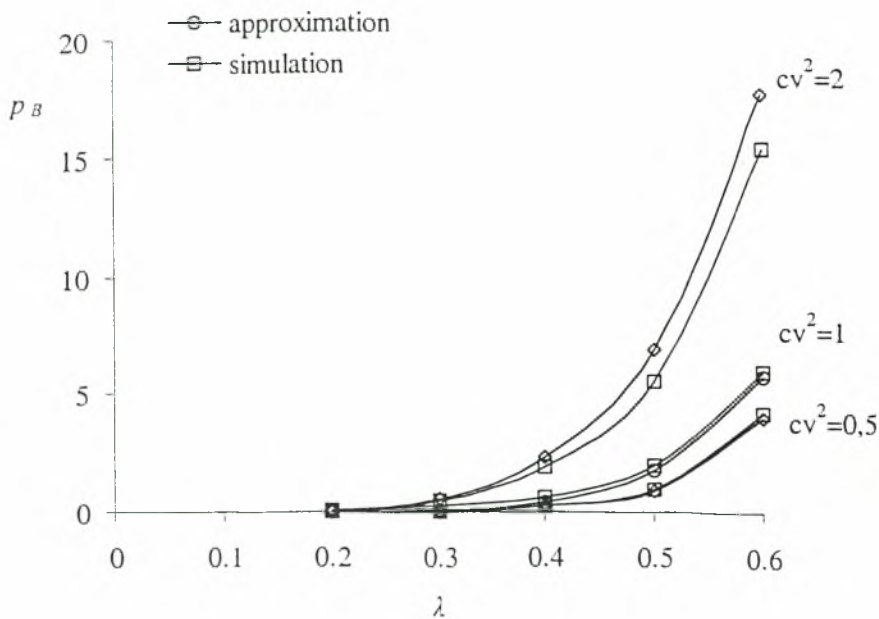


Figure 3-6: Proportion of backordered demands versus the average arrival rate of demands for different values of the squared coefficient of variation (Example 2).

Configuration	Simulation		Approximation		
	Production Capacity	Confidence Interval	Production Capacity	Relative Error	Iterations
2.1: $N = 3; K = 5; cv^2 = 0.5$	0.929	$\pm 0.1\%$	0.934	+ 0.5%	11
2.2: $N = 3; K = 5; cv^2 = 1$	0.876	$\pm 0.2\%$	0.873	- 0.3%	7
2.3: $N = 3; K = 5; cv^2 = 2$	0.813	$\pm 0.3\%$	0.808	- 0.6%	13

Table 3-7: Production capacity of the echelon kanban control system (Example 2).

Configuration	Simulation		Approximation		
	Production Capacity	Confidence Interval	Production Capacity	Relative Error	Iterations
2.1: $N = 3; K = 5; cv^2 = 0.5$	0.926	$\pm 0.2\%$	0.932	+ 0.6%	2
2.2: $N = 3; K = 5; cv^2 = 1$	0.870	$\pm 0.1\%$	0.865	- 0.6%	2
2.3: $N = 3; K = 5; cv^2 = 2$	0.787	$\pm 0.5\%$	0.786	- 0.2%	2

Table 3-8: Production capacity of the conventional kanban control system (Example 2).

3.7.2 Optimization of Parameters

The main reason for developing an approximation method for the performance evaluation of an echelon kanban control system is to use it to optimize the design parameters of the system. The design parameters of the echelon kanban control system are the echelon kanbans of each stage. In order to optimize these parameters, we must define a performance measure of the system. Typical performance measures are those which include the cost of not being able to satisfy the demands on time (i.e. quality of service) and the cost of producing parts ahead of time and therefore building up inventory (inventory holding cost). In this chapter, we consider an optimization problem where the objective is to meet a certain quality of service constraint with minimum inventory holding cost.

We examine two quality-of-service measures as in [26], [29]. The first measure is the probability that when a customer demand arrives, it is backordered, and the second measure is the probability that when a customer demand arrives, it sees more than n waiting demands, excluding itself. The first measure is denoted by P_{rupt} and concerns the situation where the

demands must be immediately satisfied. The second measure is denoted by $P(Q > n)$ and concerns the situation where we have the prerogative to introduce a delay in filling orders, which is equivalent to authorizing demands to wait. Specifically, P_{rupt} is the stationary probability of having no finished parts in the last synchronization station, and can be computed as the marginal distribution of having no finished parts in that station, which is given by (3.18) in Appendix 3A. Similarly, $P(Q > n)$ is the stationary probability of having more than n customers waiting and can be computed from the following expression:

$$P(Q > n) = \sum_{x=n+1}^{\infty} P(Q = x) = 1 - \sum_{y=0}^n P(Q = y) \quad (3.10)$$

where $P(Q = n)$ is given by (see Appendix 3A):

$$P(Q = n) = p_o^N(0, n) = p_o^N(0, 0) \left(\frac{\lambda_D}{\lambda_o^N(0)} \right)^n \quad (3.11)$$

The stationary distribution $p_o^N(0, 0)$ that is needed to evaluate both P_{rupt} and $P(Q > n)$ is given by the following expression:

$$p_o^N(0, 0) = \frac{1}{\frac{1}{1 - \frac{\lambda_D}{\lambda_o^N(0)}} + \sum_{x=1}^{K_N} \left(\frac{1}{\lambda_D^x} \prod_{i=0}^{x-1} \lambda_o^N(i) \right)} \quad (3.12)$$

The cost function that we want to minimize is the long run expected average cost of holding inventory,

$$C_{\text{total}} = \sum_{i=1}^N h_i E[WIP_i + FP_i] \quad (3.13)$$

where h_i is the unit cost of holding $WIP_i + FP_i$ inventory per unit time in stage i .

In this subsection, we optimize the echelon kanbans of an echelon kanban control system made up of $N = 5$ stages, where each stage contains a single machine with exponentially distributed service time with mean equal to 1, for different combinations of inventory holding cost rates $h_i, i = 1, \dots, 5$, and arrival rate $\lambda_D = 0.5$. In all cases we assume that there is value added to the parts at every stage so that the inventory holding cost increases as the stage increases i.e. $h_1 < h_2 < \dots < h_5$. If this were not the case, i.e. if $h_1 = h_2 = \dots = h_5$, then clearly it would make no sense to block the passage of parts from one stage to another via the use of echelon kanbans, because this would not lower the inventory holding cost but would worsen the quality of service. This implies that if $h_1 = h_2 = \dots = h_5$, the optimal echelon kanbans satisfy $K_1 \leq K_i, i = 2, \dots, 5$, in which case the echelon kanban control system is equivalent to a CONWIP system [68] with a WIP-cap on the total number of parts in the system equal to K_1 .

Table 3-9 shows the optimal design parameters (K_1, \dots, K_5) and associated minimum long run expected average cost of holding inventory for different quality of service constraints (design criteria) and inventory holding cost rates h_1, \dots, h_5 , where $h_1 < h_2 < \dots < h_5$. The quality of service constraints that we use are $P_{\text{rupt}} \leq 0.02$ and $P(Q > n) \leq 0.02$, for $n = 2, 5$ and 10. From the results, we note that the higher the number of backordered demands, n , the lower the optimal number of echelon kanbans, and hence the inventory holding cost. As the difference between the holding cost rates $h_i, i = 1, \dots, 5$, increases the difference between the optimal values of $K_i, i = 1, \dots, 5$ increases since the behavior of the echelon kanban control system diverts from that of the CONWIP system. When the difference between the holding cost rates $h_i, i = 1, \dots, 5$, is low, the behavior of the echelon kanban control system tends to that of the CONWIP system.

Design criterion	K_1	K_2	K_3	K_4	K_5	Cost
$h_1 = 1, h_2 = 2, h_3 = 3, h_4 = 4, h_5 = 5$						
$P_{\text{rupt}} \leq 0.02$	15	13	12	10	8	55.8854
$P(Q > 2) \leq 0.02$	13	11	10	8	7	46.5547
$P(Q > 5) \leq 0.02$	10	8	7	6	2	31.1202
$P(Q > 10) \leq 0.02$	7	6	5	3	1	20.2531
$h_1 = 3, h_2 = 8, h_3 = 9, h_4 = 10, h_5 = 12$						
$P_{\text{rupt}} \leq 0.02$	15	13	12	10	8	144.3137
$P(Q > 2) \leq 0.02$	13	11	10	9	6	121.1610
$P(Q > 5) \leq 0.02$	10	8	7	6	2	84.0744
$P(Q > 10) \leq 0.02$	7	6	5	3	1	57.3596
$h_1 = 1, h_2 = 2, h_3 = 4, h_4 = 11, h_5 = 12$						
$P_{\text{rupt}} \leq 0.02$	15	14	13	9	8	121.2880
$P(Q > 2) \leq 0.02$	14	13	10	7	6	98.8898
$P(Q > 5) \leq 0.02$	10	9	8	5	2	67.3833
$P(Q > 10) \leq 0.02$	8	6	4	3	1	39.4826
$h_1 = 1, h_2 = 6, h_3 = 11, h_4 = 16, h_5 = 21$						
$P_{\text{rupt}} \leq 0.02$	17	13	11	10	8	218.7023
$P(Q > 2) \leq 0.02$	15	11	10	8	5	178.1619
$P(Q > 5) \leq 0.02$	10	8	7	6	2	115.6012
$P(Q > 10) \leq 0.02$	8	6	5	3	1	76.5234
$h_1 = 1, h_2 = 11, h_3 = 21, h_4 = 31, h_5 = 41$						
$P_{\text{rupt}} \leq 0.02$	17	13	11	10	8	420.4045
$P(Q > 2) \leq 0.02$	15	11	10	8	5	341.3238
$P(Q > 5) \leq 0.02$	10	8	7	6	2	221.2025
$P(Q > 10) \leq 0.02$	8	6	5	3	1	145.0467
$h_1 = 1, h_2 = 2, h_3 = 4, h_4 = 8, h_5 = 16$						
$P_{\text{rupt}} \leq 0.02$	17	15	12	9	7	143.8791
$P(Q > 2) \leq 0.02$	14	13	11	7	5	112.4422
$P(Q > 5) \leq 0.02$	10	8	7	6	2	65.8427
$P(Q > 10) \leq 0.02$	8	6	5	3	1	39.9336
$h_1 = 1, h_2 = 3, h_3 = 9, h_4 = 27, h_5 = 81$						
$P_{\text{rupt}} \leq 0.02$	19	17	14	10	6	633.1783
$P(Q > 2) \leq 0.02$	17	15	12	8	4	471.8673
$P(Q > 5) \leq 0.02$	12	10	8	6	1	231.4458
$P(Q > 10) \leq 0.02$	8	6	5	3	1	139.0666

Table 3-9: Optimal configuration and associated costs for different values of h_1, \dots, h_5 and λ_D

= 0.5 for the echelon kanban control system.

Table 3-10 shows the optimal design parameter K_1 and associated minimum inventory holding cost for different quality of service constraints and inventory holding cost rates h_1, \dots, h_5 , and $\lambda_D = 0.5$, for the CONWIP system. Comparing the results between Table 3-9 and Table 3-10, we note that the CONWIP system performs quite worse than the echelon kanban system.

Design criterion	K_1	Cost
$h_1 = 1, h_2 = 6, h_3 = 11, h_4 = 16, h_5 = 21$		
$P_{\text{rupt}} \leq 0.02$	14	244.1633
$P(Q > 2) \leq 0.02$	12	202.4152
$P(Q > 5) \leq 0.02$	10	161.0062
$P(Q > 10) \leq 0.02$	8	120.3066
$h_1 = 1, h_2 = 11, h_3 = 21, h_4 = 31, h_5 = 41$		
$P_{\text{rupt}} \leq 0.02$	14	474.3265
$P(Q > 2) \leq 0.02$	12	392.8304
$P(Q > 5) \leq 0.02$	10	312.0123
$P(Q > 10) \leq 0.02$	8	232.6132
$h_1 = 1, h_2 = 2, h_3 = 4, h_4 = 8, h_5 = 16$		
$P_{\text{rupt}} \leq 0.02$	14	175.1600
$P(Q > 2) \leq 0.02$	12	143.4069
$P(Q > 5) \leq 0.02$	10	111.9860
$P(Q > 10) \leq 0.02$	8	81.2605
$h_1 = 1, h_2 = 3, h_3 = 9, h_4 = 27, h_5 = 81$		
$P_{\text{rupt}} \leq 0.02$	14	850.9274
$P(Q > 2) \leq 0.02$	12	690.3584
$P(Q > 5) \leq 0.02$	10	531.7149
$P(Q > 10) \leq 0.02$	8	377.1016

Table 3-10: Optimal configuration and associated costs for different values of h_1, \dots, h_5 and $\lambda_D = 0.5$ for the CONWIP system.

3.8 Conclusions

We developed an analytical decomposition-based approximation method for the performance evaluation of an echelon kanban control system and tested it with several numerical examples. The numerical examples showed that the method is quite accurate in

most cases. They also showed that the echelon kanban control system has some advantages over the conventional kanban control system. Specifically, when the two systems have the same value of K , the production capacity of the echelon kanban control system has higher production capacity, lower average number of backordered demands, but only slightly higher average WIP and either slightly higher or slightly lower FG inventory than the conventional kanban control system. The numerical results also showed that as the variability of the service time distribution increases, the production capacity of the echelon kanban control system and the accuracy of the approximation method decrease. Finally, we know that the optimized echelon kanban control system always performs at least as well as the optimized CONWIP system since the latter system is a special case of the first system. The numerical results showed that in fact the superiority in performance of the echelon kanban control system over that of the CONWIP system can be quite significant, particularly when the increase in inventory holding costs from one stage to its downstream stage becomes larger.

Appendix 3A Analysis of synchronization station O^N

O^N is a synchronization station fed by a Markovian arrival process with state-dependent arrival rate $\lambda_o^N(n_o^N)$, $0 \leq n_o^N < K_N$, and an external Poisson process with rate λ_D . The underlying Continuous Time Markov Chain is shown in Figure 3-7. The state of this Markov chain is (n_o^N, n_D) , where n_o^N is the number of engaged kanbans and $n_D, n_D \geq 0$, is the number of external resources (customer demands) currently present in subsystem O^N . Let $p_o^N(n_o^N, n_D)$ be the steady-state probabilities of the above state. These probabilities are solution of the following balance equations:

$$p_o^N(n_o^N, 0)\lambda_D = p_o^N(n_o^N - 1, 0)\lambda_o^N(n_o^N - 1) \quad \text{for } n_o^N = 1, 2, \dots, K_N, \quad (3.14)$$

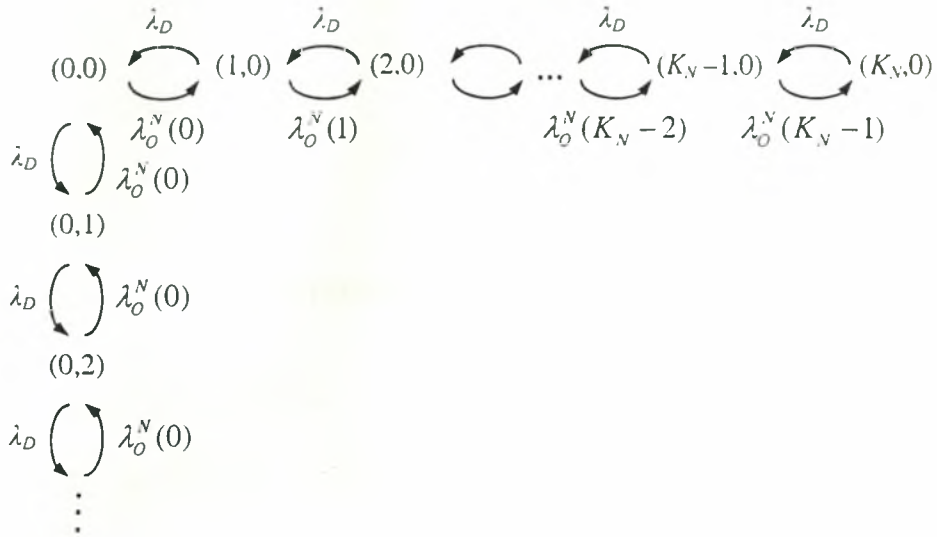


Figure 3-7: Markov chain describing the state (n_0^N, n_D) of synchronization station O^N .

$$p_O^N(0, n_D) \lambda_O^N(0) = p_O^N(0, n_D - 1) \lambda_D \quad \text{for } n_D > 0. \quad (3.15)$$

The marginal probabilities $P_O^N(n_0^N)$ are then simply given by

$$P_O^N(n_0^N) = p_O^N(n_0^N, 0) \quad \text{for } n_0^N = 1, 2, \dots, K_N, \quad (3.16)$$

$$P_O^N(0) = \sum_{n_D=0}^{\infty} p_O^N(0, n_D). \quad (3.17)$$

From (3.15) and (3.17) we get

$$P_O^N(0) = \sum_{n_D=0}^{\infty} p_O^N(0, 0) \left(\frac{\lambda_D}{\lambda_O^N(0)} \right)^{n_D} = p_O^N(0, 0) \frac{1}{1 - \frac{\lambda_D}{\lambda_O^N(0)}}. \quad (3.18)$$

The conditional throughputs of subsystem O^N are then obtained from (3.5), (3.14) and (3.16), as follows:

$$v_O^N(n_0^N) = \lambda_D \quad \text{for } n_0^N = 2, 3, \dots, K_N \quad (3.19)$$

From (3.5), (3.14), (3.16) and (3.18), we also get

$$v_0^v(1) = \frac{1}{\frac{1}{\lambda_D} - \frac{1}{\lambda_0^N(0)}}. \quad (3.20)$$

Appendix 3B Analysis of synchronization station I^i

I^i , $i = 2, \dots, N$, is a synchronization station fed by two Markovian arrival process with state-dependent arrival rates: $\lambda_i^i(n_i^i)$, $0 \leq n_i^i \leq K_i$, and $\lambda^i(n^i)$, $0 \leq n^i \leq K_{i-1}$. The underlying continuous-time Markov chain is shown in Figure 3-8. The state of this Markov chain is (n_i^i, n_u^i) , where n_i^i is the number of free kanbans and n_u^i is the number of external resources (finished parts of stage $i-1$) currently present in subsystem I^i . Recall that n^i can be obtained from n_u^i and n_i^i using (3.3). The steady-state probabilities $p_i^i(n_i^i, n_u^i)$ can be derived as solutions of the underlying balance equations and are given by:

$$p_i^i(n_i^i, 0) = \left[\prod_{n=1}^{n_i^i} \frac{\lambda_i^i(n-1)}{\lambda^i(K_i - n)} \right] p_i^i(0, 0), \quad (3.21)$$

$$p_i^i(0, n_u^i) = \frac{\prod_{n=1}^{n_u^i} \lambda^i(K_i + n - 1)}{[\lambda_i^i(0)]^{n_u^i}} p_i^i(0, 0). \quad (3.22)$$

The marginal probabilities, $P_i^i(n_i^i)$, can then be derived by summing up probabilities above as follows:

$$P_i^i(n_i^i) = \left[\prod_{n=1}^{n_i^i} \frac{\lambda_i^i(n-1)}{\lambda^i(K_i - n)} \right] p_i^i(0, 0) \text{ for } n_i^i = 1, 2, \dots, K_i, \quad (3.23)$$

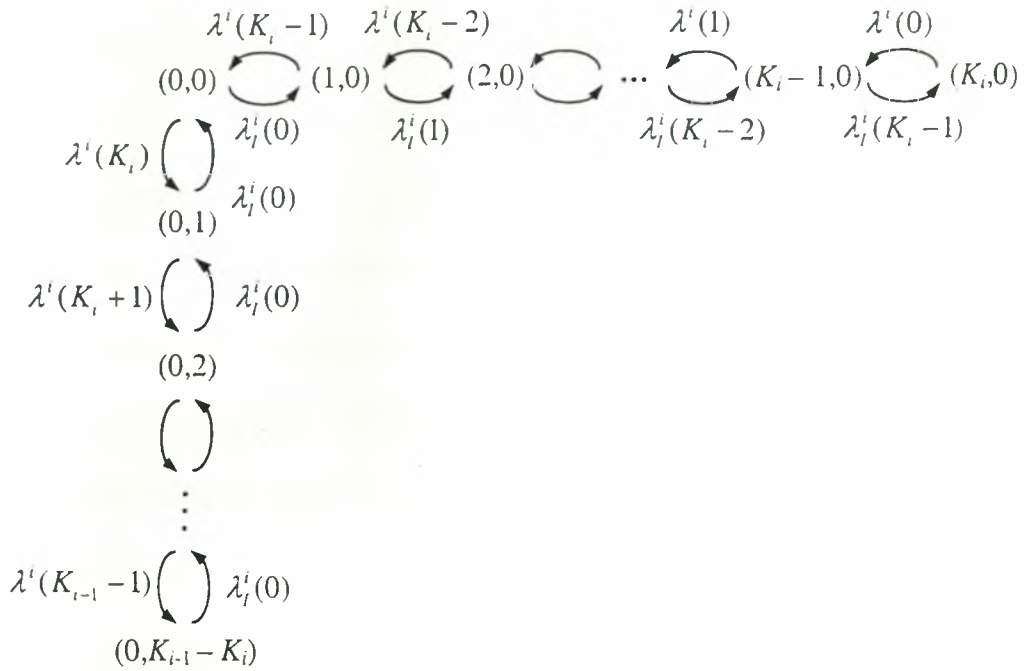


Figure 3-8: Markov chain describing the state (n_l^i, n_u^i) of queuing network l^i .

$$P_l^i(0) = \left[1 + \sum_{n_u^i=1}^{K_{i-1}-K_i} \frac{\prod_{n=1}^{n_u^i} \lambda^i(K_i+n-1)}{[\lambda_l^i(0)]^{n_u^i}} \right] p_l^i(0,0). \quad (3.24)$$

The estimation of the conditional throughputs of subsystem l^i can then be obtained by substituting the above probabilities into (3.5), as follows:

$$v_l^i(n_l^i) = \lambda^i(K_i - n_l^i) \quad \text{for } n_l^i = 2, 3, \dots, K_i, \quad (3.25)$$

$$v_l^i(1) = \lambda^i(K_i - 1) \left[1 + \sum_{n_u^i=1}^{K_{i-1}-K_i} \frac{\prod_{n=1}^{n_u^i} \lambda^i(K_i+n-1)}{[\lambda_l^i(0)]^{n_u^i}} \right]. \quad (3.26)$$

Chapter 4 Thesis Summary

In this thesis we investigated properties of kanban-type policies for the coordination of multi-stage production-inventory systems with and without advance demand information and we developed an analytical method for the performance evaluation of an echelon kanban control system.

In Chapter 1, we provided some background information, which led to the motivation behind this thesis. We also reviewed the literature, which is related to the thesis, and we gave a brief description of the two main parts of the thesis, which occupy Chapters 2 and 3, respectively.

In Chapter 2, we numerically investigated the tradeoffs between optimal base stock levels, numbers of kanbans, and planned supply lead times in single-stage and two-stage production-inventory systems operating under base stock and hybrid base stock/kanban policies with ADI. The results of our investigation lead to the following conjectures.

In multi-stage make-to-stock production-inventory control policies in which a base stock level of FG inventory is set at every stage, that base stock level represents finished goods that have been produced before any demands have arrived to the system to protect the system against uncertainties in production or demand that may cause costly backorders. Holding inventory, however, is itself costly and so the optimal base stock level at every stage should be as low as possible. The lowest possible optimal base stock level is attained when the replenishment policy adopted is such that a replenishment order is issued and released into

the facility of every stage immediately after the arrival of the customer demand that triggered it. This can be achieved by setting the echelon planned supply lead time at the first stage greater than or equal to the demand lead time, and by setting the number of kanbans equal to infinity at every stage, so that no inventory limit is imposed at any stage. Delaying or postponing the issuing of a replenishment production order by means of an offsetting by the echelon planned supply lead time mechanism or a kanban mechanism appears to lower the total cost as long as it does not cause an increase in the optimal base stock level of FG inventory above its lowest possible value. Moreover, for a fixed demand lead time, the more downstream a stage is, the less advance demand information is available and so the higher the need to keep a base stock of FG inventory of that stage. As the demand lead time increases, the amount of advance demand information increases from downstream to upstream, and so the need to keep a base stock of FG inventory at each stage decreases from downstream to upstream. The results in this chapter indicate that it is optimal to reduce the optimal base stock levels at all stages until they drop to zero, one after the other, starting from the last stage and moving upstream the system. Finally, the optimal number of kanbans determines the optimal production capacity of the system and appears to be independent of the amount of ADI.

One attractive feature of the above procedure may be the case where we have more complex manufacturing stages like parallel machines in each stage and/or a combination of parallel machines and machines in series and the case where we have larger systems with more than two stages in series for each one of the control systems that presented in Chapter 2.

In Chapter 3, we developed an analytical decomposition-based approximation method for the performance evaluation of an echelon kanban control system and tested it with several numerical examples. The numerical examples showed that the method is quite accurate in most cases. They also showed that the echelon kanban control system has some advantages

over the conventional kanban control system. Specifically, when the two systems have the same value of K , the production capacity of the echelon kanban control system has higher production capacity, lower average number of backordered demands, but only slightly higher average WIP and either slightly higher or slightly lower FG inventory than the conventional kanban control system. The numerical results also showed that as the variability of the service time distribution increases, the production capacity of the echelon kanban control system and the accuracy of the approximation method decrease. Finally, we know that the optimized echelon kanban control system always performs at least as well as the optimized CONWIP system since the latter system is a special case of the first system. The numerical results showed that in fact the superiority in performance of the echelon kanban control system over that of the CONWIP system can be quite significant, particularly when the increase in inventory holding costs from one stage to its downstream stage becomes larger.

The method presented in Chapter 3 is sufficiently general. Specifically, one feature of the method is that it can be extended to handle different and more general situations than considered in Chapter 3. In particular, we can handle more general arrival processes for customer demands than Poisson (namely PH arrival processes). Also, it is possible to consider the case where instead of having backordered demands, the demands that cannot be satisfied immediately may be lost. In both cases, the only modification of our method pertains to the analysis of synchronization station J_N of stage. Another extension deals with handling more complex manufacturing stages, like the case, where in each stage there are a number of parallel machines and the case where the intermediate buffers between machines have finite capacity [5]. Finally, other possible extensions include dealing with extended kanban control systems, multiple producers and/or customers and assembly systems.

Bibliography

- [1] Altiok, T. (1996) *Performance Analysis of Manufacturing Systems*, Springer Series in Operations Research, Springer, Berlin.
- [2] Avi-Itzhak, B. and D.P. Heyman (1973) "Approximate Queuing Models for Multiprogramming Computer Systems," *Operations Research*, 21, 1212-1230.
- [3] Balsamo, S., V. de Nitto Personé, and R. Onvural (2001) *Analysis of Queueing Networks with Blocking*, International Series in Operations Research and Management Science, Kluwer Academic Publishers.
- [4] Baskett, F., K.M. Chandy, R.R. Muntz and F. Palacios-Gomez (1975) "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," *Journal of ACM*, 22, 248-260.
- [5] Baynat, B. and Y. Dallery (1993) "A Unified View of Product-Form Approximation Techniques for General Closed Queuing Networks," *Performance Evaluation*, 18 (3), 205-224.
- [6] Baynat, B. and Y. Dallery (1993) "Approximate Techniques for General Closed Queuing Networks with Subnetworks having Population Constraints," *European Journal of Operational Research*, 69, 250-264.
- [7] Baynat, B. and Y. Dallery (1995) "Approximate Analysis of Multi-Class Synchronized Closed Queuing Networks," in *Proceedings of the International Workshop on Modeling*,

Analysis and Simulation of Computer and Telecommunications Systems, Durham, North Carolina.

- [8] Baynat, B. and Y. Dallery (1996) "A Product-Form Approximation Method for General Closed Queuing Networks with Several Classes of Customers," *Performance Evaluation*, 24, 165-188.
- [9] Benjaafar S. and J.-S. Kim (2001) "When Does Higher Demand Variability Lead to Lower Safety Stocks?" Working Paper, Department of Mechanical Engineering, University of Minnesota.
- [10] Berkley, B.J. (1992) "A Review of the Kanban Production Control Research Literature," *Production and Operations Management*, 1 (4), 393-411.
- [11] Bonvik, A.M., C.E. Couch and S.B. Gershwin, (1997) "A Comparison of Production-Line Control Mechanisms," *International Journal of Production Research*, 35 (3), 789-804.
- [12] Bourland, K.E., S.G. Powell and D.F. Pyke (1996) "Exploiting Timely Demand Information to Reduce Inventories," *European Journal of Operational Research*, 92, 239-253.
- [13] Brandwajn, A.. (1985) "Equivalence and Decomposition in Queuing Systems: A Unified Approach," *Performance Evaluation*, 5, 175-186.
- [14] Bruell, S.C. and G. Balbo (1980) *Computational Algorithms for Closed Queuing Networks*, Elsevier North-Holland, Amsterdam.
- [15] Buzacott, J.A. (1989) "Queueing Models of Kanban and MRP Controlled Production Systems," *Engineering Costs and Production Economics*, 17, 3-20.

- [16] Buzacott, J.A. and J.G. Shanthikumar (1993) *Stochastic Models of Manufacturing Systems*, Prentice-Hall, Englewood Cliffs, NJ.
- [17] Buzacott, J.A. and J.G. Shanthikumar (1994) "Safety Stock versus Safety Time in MRP Controlled Production Systems," *Management Science*, 40 (12), 1678-1689.
- [18] Buzen, J.P. (1973) "Computational Algorithms for Closed Queuing Networks with Exponential Servers," *Comm. ACM*, 16 (9), 527-531.
- [19] Chen, F. (2001) "Market Segmentation, Advanced Demand Information, and Supply Chain Performance," *Manufacturing & Service Operations Management*, 3, 53-67.
- [20] Cox, D.R. (1955) "A Use of Complex Probabilities in the Theory of Stochastic Processes". *Proc. Camb. Philos. Soc.*, 51, 313-319.
- [21] Dallery, Y. (1990) "Approximate Analysis of General Open Queuing Networks with Restricted Capacity," *Performance Evaluation*, 11 (3), 209-222.
- [22] Dallery, Y. and S.B. Gershwin (1992) "Manufacturing flow line systems: A review of models and analytical results," *Queueing Systems*, 12, 3-94.
- [23] Dallery, Y. and G. Liberopoulos (2000) "Extended Kanban Control System: Combining Kanban and Base Stock," *IIE Transactions*, 32 (4), 369-386.
- [24] Dallery, Y. and X. Cao (1992) "Operational Analysis of Stochastic Closed Queuing Networks." *Performance Evaluation*, 14 (1), 43-61.
- [25] Daryanto, A., J.C.W van Ommeren and W.H.M. Zijm (2003) "A Closed-Loop Two-Indenture Repairable Item System," in *Proceedings of the Fourth Aegean International Conference on Analysis of Manufacturing Systems*, Samos Island, Greece, July 1-4, Ziti Publishing. Thessaloniki, Greece, 191-201.

- [26] De Araujo, S.L., (1994) *Sur l'Analyse et le Dimensionnement des Systèmes de Production Gères en Kanban*, Ph.D. Dissertation, INPG, Laboratoire d'Automatique de Grenoble, France.
- [27] DeCroix, G.A. and V.S. Mookerjee (1997) "Purchasing Demand Information in a Stochastic-Demand Inventory System," *European Journal of Operations Research*, 102 (1), 36-57.
- [28] Di Mascolo, M., Y. Frein and Y. Dallery (1996) "An Analytical Method for Performance Evaluation of Kanban Controlled Production Systems," *Operations Research*, 44 (1), 50-64.
- [29] Duri, C., Y. Frein and M. Di Mascolo (2000) "Comparison among Three Pull Control Policies: Kanban, Base Stock and Generalized Kanban," *Annals of Operations Research*, 93, 41-69.
- [30] Frein, Y., M. Di Mascolo and Y. Dallery (1995) "On the Design of Generalized Kanban Control Systems," *International Journal of Operations and Production Management*, 15 (9), 158-184.
- [31] Gallego, G. and A.Ö. Özer (2001) "Integrating Replenishment Decisions with Advance Demand Information," *Management Science*, 47 (10), 1344-1360.
- [32] Gavirneni, S. and S. Tayur (1999) "Managing a Customer Following a Target Reverting Policy," *Manufacturing & Service Operations Management*, 1 (2), 157-173.
- [33] Gershwin, S.B. (1994) *Manufacturing Systems Engineering*, Prentice Hall, Prentice-Hall, Englewood Cliffs, NJ.
- [34] Gilbert, S.M. and R.H. Ballou (1999) "Supply Chain Benefits from Advanced Customer Commitments," *Journal of Operations Management*, 18 (1), 61-73.

- [35] Gordon, W.J. and G.F. Newell (1967) "Closed Queuing Networks with Exponential Servers," *Operations Research*, 15, 252-267.
- [36] Graves, S.C., D.B. Kletter and W.B. Hetzel (1998) "A Dynamic Model for Requirements Planning With Application to Supply Chain Optimization," *Operations Research*, 46 (3), S25-49.
- [37] Graves, S.C., H.C. Meal, S. Dasu and Y. Qiu (1986) "Two-Stage Production Planning in a Dynamic Environment," in *Multi-Stage Production Planning and Inventory Control*, S. Axsater, C. Schneeweiss, and E. Silver, (eds.), *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin, 266, 9-43.
- [38] Güllü, R. (1996) "On the Value of Information in Dynamic Production/Inventory Problems under Forecast Evolution," *Naval Research Logistics*, 43 (2), 289-303.
- [39] Haji, R. and G. Newell (1971) "A Relation between Stationary Queue Lengths and Waiting Time Distributions," *Journal of Applied Probability*, 8, 617-620.
- [40] Hariharan, R. and P. Zipkin (1995) "Customer-Order Information, Leadtimes, and Inventories," *Management Science*, 41 (10), 1599-1607.
- [41] Heath, D.C. and P.L. Jackson (1994) "Modeling the Evolution of Demand Forecasts With Application to Safety-Stock Analysis in Production/Distribution Systems," *IIE Transactions*, 26 (3) 17-30.
- [42] Jackson, J.R. (1963) "Jobshop-Like Queuing Systems," *Management Science*, 10 (1), 131-142.
- [43] Karaesmen, F. and Y. Dallery (2000) "A Performance Comparison of Pull Control Mechanisms for Multi-Stage Manufacturing Systems," *International Journal of Production Economics*, 68, 59-71.

- [44] Karaesmen, F., J.A. Buzacott and Y. Dallery (2002) "Integrating Advance Order Information in Make-to-Stock Production," *IIE Transactions*, 34 (8), 649-662.
- [45] Karaesmen, F., Liberopoulos, G. and Dallery, Y. (2003) "The Value of Advance Demand Information in Production/Inventory Systems," *Annals of Operations Research* (to appear).
- [46] Karaesmen, F., Liberopoulos, G. and Dallery, Y., (2003) "Production/Inventory Control with Advance Demand Information," in *Stochastic Modelling and Optimization of Manufacturing Systems and Supply Chains*, J.G. Shanthikumar, D.D. Yao and W.H.M. Zijm (eds.), International Series in Operations Research and Management Science, Kluwer Academic Publishers.
- [47] Kuehn, P.J. (1979) "Approximate Analysis of General Queuing Networks by Decomposition," *IEEE Transactions on Communications*, 27 (1), 113-126.
- [48] Labetoulle, J. and G. Pujolle (1980) "Isolation Method in a Network of Queues," *IEEE Transactions on Software Engineering*, 6 (4), 373-381.
- [49] Lazowska, E.D., J. Zahorjan, G.S. Graham and K.C. Sevcik (1984) *Quantitative System Performance: Computer System Analysis Using Queuing Network Models*, Prentice-Hall, Englewood Cliffs, NJ.
- [50] Liberopoulos, G. and I. Tsikis (2003) "Unified Modeling Framework of Multi-Stage Production-Inventory Control Policies with Lot-Sizing and Advance Demand Information," in *Stochastic Modelling and Optimization of Manufacturing Systems and Supply Chains*, J.G. Shanthikumar, D.D. Yao and W.H.M. Zijm (eds.), *International Series in Operations Research and Management Science*, Vol. 63, Kluwer Academic Publishers, Boston, MA, 271-297.

- [51] Liberopoulos, G. and Y. Dallery (2000) "A Unified Framework for Pull Control Mechanisms in Multi-Stage Manufacturing Systems," *Annals of Operations Research*, 93, 325-355.
- [52] Liberopoulos, G. and Y. Dallery (2002) "Base Stock versus WIP Cap in Single-Stage Make-to-Stock Production-Inventory Systems," *IIE Transactions*, 34 (7), 613-622.
- [53] Liberopoulos, G. and Y. Dallery (2002) "Comparative Modeling of Multi-Stage Production-Inventory Control Policies with Lot Sizing," *International Journal of Production Research*, 41 (6), 1273-1298.
- [54] Marie, R. (1979) "An Approximate Analytical Method for General Queuing Networks," *IEEE Transactions on Software Engineering*, 5 (5), 530-538.
- [55] Marie, R. (1980) "Calculating Equilibrium Probabilities for $\lambda(n)/C_k/1/N$ Queues," *Performance Evaluation Review*, 9, 117-125.
- [56] Marie, R., P. Snyder and W. Stewart (1982) "Extensions and Computational Aspects of an Iterative Method," *ACM Sigmetrics*, Sept. Washington.
- [57] Marklund, J. (1999) "Controlling Inventories in Divergent Supply Chains with Advance-Order Information," Working Paper, Department of Industrial Management and Logistics, Lund University, Sweden.
- [58] Milgrom, P. and J. Roberts (1988) "Communication and Inventory as Substitutes in Organizing Production," *Scandinavian Journal of Economics*, 90, 275-289.
- [59] Papadopoulos, H.T. and C. Heavey (1996) "Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines," *European Journal of Operational Research*, 92, 1-27.

- [60] Perros, H.G. (1994) *Queueing Networks with Blocking: Exact and Approximate Solutions*, Oxford University Press.
- [61] Perros, H.G. and T. Altiok, (Eds.) (1989) *First International Workshop on Queueing Networks With Blocking*, North Holland, Amsterdam.
- [62] Reiser, M. and S.S. Lavenberg (1980) "Mean Value Analysis of Closed Multichain Queueing Networks," *Journal of ACM*, 27 (2), 313-322.
- [63] Rubio R. and L.W. Wein (1996) "Setting Base Stock Levels Using Product-Form Queueing Networks," *Management Science*, 42 (2), 259-268.
- [64] Sauer, C.H. and K.M. Chandy (1981) *Computer Systems Performance Modelling*, Prentice-Hall, Englewood Cliffs, NJ.
- [65] Schweitzer, P.J. "A Survey of Mean Value Analysis, its Generalizations, and Applications for Networks of Queues," Working Paper.
- [66] Song, J.S. and D.D. Yao (2002) "Performance Analysis and Optimization of Assemble-to-Order Systems with Random Lead Times," *Operations Research*, 50 (5), 889-903.
- [67] Spearman, M.L. (1992) "Customer Service in Pull Production Systems," *Operations Research*, 40 (5), 948-958.
- [68] Spearman, M.L., D.L. Woodruff and W.J. Hopp (1990) "CONWIP: A Pull Alternative to Kanban," *International Journal of Production Research*, 28, 879-894.
- [69] Stewart, W.J. and R. Marie, (1980) "A Numerical Solution for the $\lambda(n)/C_k/r/N$ Queue," *European Journal of Operational Research*, 5, 56-68.
- [70] Stewart, W.J., (1978) "A Comparison of Numerical Techniques in Markov Modelling," *Communications of the ACM*, 21 (2), 144-152.

- [71] Toktay, L.B. and L.M. Wein (2000) "Analysis of a Forecasting-Production-Inventory System with Stationary Demand," *Management Science*, 47 (9), 1268-1281.
- [72] Van Donselaar, K., L.R. Kopczak and M. Wouters (2001) "The Use of Advance Demand Information in a Project-Based Supply Chain," *European Journal of Operational Research*, 130, 519-538.
- [73] Veach, M.H. and L.M. Wein (1994) "Optimal Control of a Two-Station Tandem Production-Inventory System," *Operations Research*, 42 (2), 337-350.
- [74] Whitt, W. (1983) "The Queuing Network Analyser," *Bell Systems Technology Journal*, 62 (9), 2779-2815.
- [75] Zipkin, P. (1989) "A Kanban-Like Production Control System: Analysis of Simple Models," Research Working Paper No. 89-1, Graduate School of Business, Columbia University, New York.
- [76] Zipkin, P. (2000) *Foundations of Inventory Management*, McGraw Hill: Management & Organization Series, Boston, MA.