A COMPARISON OF PULL CONTROL POLICIES IN HYBRID PRODUCTION SYSTEMS

by

Özge Çadırcı

B.S., Industrial Engineering, Yıldız Technical University, 2005

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2008

ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis supervisor Assist. Prof. Aybek Korugan for his guidance and help during my study. He is not only a motivating supervisor, but also a very friendly person. He behaves his students as if they were his friends. I am also grateful to Assoc. Prof. Necati Aras and Assoc. Prof. Aslı Erdem for their important contributions and comments.

I would like to express my special thanks to my mother, my father and my brother for supporting and encouraging me throughout my life.

Furthermore, special thanks to my friends Kemal and Ferhat because of their support; and also TÜBİTAK which supported me financially during my master education.

ABSTRACT

A COMPARISON OF PULL CONTROL POLICIES IN HYBRID PRODUCTION SYSTEMS

Environmental concerns trigger companies for utilizing reuse activities as much as possible, gradually. One of these activities is remanufacturing process. When manufacturing process is combined with remanufacturing process, the system is referred to as a hybrid production system.

This study concentrates on a hybrid production system that performs both remanufacturing and manufacturing activities under pull type production control. The incoming demand is satisfied by the output of either process, where remanufactured products are assumed to be restored into “as good as new” condition. In the study, four of the most common pull control systems, viz. Kanban Control System (KCS), Base Stock Control System (BSCS), Generalized Kanban Control System (GKCS) and Extended Kanban Control System (EKCS), are compared on the single-stage hybrid production system considered.

A stochastic model is developed for each of the four control policies. Then these models are analyzed using methods based on the previous studies available in the literature in order to obtain performance measures of interest. After comparing the outcomes of analytical models with the simulation results to test accuracy, a cost function is defined using the performance measures. Then this cost function is minimized with respect to the control parameters of each control mechanism. Finally, results derived from numerical experimentation are obtained, and conclusions are drawn.

ÖZET

HİBRİT ÜRETİM SİSTEMLERİNDE ÇEKME TİPİ KONTROL MEKANİZMALARININ KARŞILAŞTIRILMASI

Çevre bilincinin giderek artması şirketlerin yeniden kullanım aktivitelerine önem vermelerini gerektirmektedir. Bu aktivitelerden biri de yeniden imalat prosesidir. İmalat prosesinin yeniden imalat prosesi ile birlikte kullanıldığı sistemlere ise hibrit sistemler denilmektedir.

Bu çalışmada, yeniden imalat ve imalat aktivitelerini çekme tipi kontrol altında yürüten hibrit bir üretim sistemi incelenmektedir. Yeniden imal edilen ürünlerin "yenisi kadar iyi" olduğu kabul edildiğinden, dışarıdan gelen talepler, söz konusu proseslerin herhangi birinin ürettiği ürünlerden karşılanabilmektedir. Çalışmada en yaygın dört tane çekme tipi kontrol sistemi göz önüne alınmıştır. Bu sistemler Kanban Kontrol Sistemi, Seviye Esaslı Envanter Sistemi, Genelleştirilmiş Kanban Kontrol Sistemi ve Genişletilmiş Kanban Kontrol Sistemi'dir.

Öncelikle, dört kontrol yönteminin her biri için stokastik bir model kurulmuş ve bu modeller performans ölçütlerini elde edebilmek amacıyla literatürde mevcut bulunan çalışmalara dayandırılarak analiz edilmiştir. Analitik modellerin sonuçları, sonuçların geçerliliğini ölçmek amacıyla simülasyon sonuçları ile karşılaştırıldıktan sonra, bu performans ölçütleri kullanılarak bir maliyet fonksiyonu tanımlanmıştır. Daha sonra bu maliyet fonksiyonu her bir kontrol mekanizmasının kontrol parametrelerine göre minimize edilmiştir. Son olarak sayısal deneylerden sonuçlara varılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

A_2	Backorder queue
A_M	Free kanban queue for raw parts
A_R	Free kanban queue for cores
b	Unit backorder cost per month
B	The buffer for cores
B_1	Free kanban queue
c_D	Cost for disposing one item
c_M	Marginal cost for manufacturing one item
c_R	Marginal cost for remanufacturing one item
D_1	Demand queue for cores
D_2	Backorder queue for KCS, BSCS and EKCS (Demand queue for GKCS)
D_R	Demand queue for cores belonging to EKCS
F_R	Free remanufacturing kanbans
F_M	Free manufacturing kanbans
FB	Fraction of backorders
G	Normalization constant associated with the single-stage network
h	Unit out-of-pocket holding cost rate per month
i	Stations of the original network
K	Kanban size
K_M	Remanufacturing kanban size
K_R	Manufacturing kanban size
n	State vector of the network
n_A	The current number in the backorder queue for GKCS
n_B	The current number in the free kanban queue for GKCS
n_{BM}	The current number in manufacturing kanban queue for EKCS
n_{BR}	The current number in remanufacturing kanban queue for EKCS
n_D	The current number in the demand queue for GKCS and EKCS
n_{DM}	The current number in the demand queue for manufacturing for EKCS

n_{DR}	The current number in the demand queue for remanufacturing for EKCS
n_{FP}	A variable representing n_P when it has positive value, and n_D when it has negative value
n_i	Current number of customers present at station *i*
n_P	The current number in the finished product queue for GKCS and EKCS
n_M	A variable representing n_{BM} when it has positive value, and n_{DM} when it has negative value
n_R	A variable representing n_{BR} when it has positive value, and n_{DR} when it has negative value
P_0	The buffer for cores
P_1	Output buffer for finished products
$P(n)$	Steady-state probabilities for the network
$\widetilde{P}_i(n_i)$	Marginal probabilities of station *i* in isolation
QA	The average queue length of backorders for GKCS
QB	The average queue length of free kanbans for GKCS
QBL	The average queue length of backorders for EKCS and BSCS
QD	The average queue length of backorders for KCS
QFG	The average queue length of backorders
QFP	The average queue length of finished products
QR	The average queue length of cores
r	Static routing probability
S	Base stock level
S_M	Finished products obtained via manufacturing
S_R	Finished products obtained via remanufacturing
$S(1)$	Set of stations visited by single-class customers
T_0	The first synchronization station
T_1	The last synchronization station
T_M	The synchronization station related to manufacturing
T_R	The synchronization station related to remanufacturing
TH_M	Expected throughput of remanufacturing process

TH_R	Expected throughput of manufacturing process
$v_i(n_i)$	Conditional throughputs of station *i* in the original network
V_i	Visit ratio of station *i*
WIPR	Average work-in-process level of remanufacturing
WIPM	Average work-in-process level of manufacturing
α	Inventory carrying charge per month
λ_D	Demand arrival rate
$\lambda_i(n_i)$	Load-dependent arrival rate of station *i*
μ_1	Service rate for remanufacturing machine
μ_2	Service rate for manufacturing machine
$\mu_i(n_i)$	Load-dependent service rates of station *i*
γ	Return arrival rate
CONWIP	Constant work in process
CTMC	Continuous-time Markov chain
EKCS	Extended kanban control system
GKCS	Generalized kanban control system
JIT	Just-in-time
KCS	Kanban control system

1. INTRODUCTION

Environmental issues have been forcing companies to reuse old products as much as possible. Reuse activities can be classified as recycling, remanufacturing and direct reuse. Remanufacturing is the collection of activities whose aim is to bring a used product back to its useful state (Korugan and Gupta, 2001). Remanufacturing can be basically defined as the transformation of used units, consisting of components and parts, into units that satisfy exactly the same standards as new units (Guide *et al.,* 1999). The remanufacturing process comprises disassembly, sorting, refurbishing and assembly operations in order to bring the product to a desired level of quality while preserving the product's identity. Remanufacturing finds most frequently applications in automotive industry, electronics industry and tire manufacturing (Gungor and Gupta, 1999).

Remanufacturing systems are faced with a greater degree of uncertainty and complexity than conventional manufacturing systems (Guide *et al.,* 1999). This uncertainty is induced by factors such as the probabilistic recovery rates of parts from the cores, unknown conditions of the recovered parts until inspected, imperfect correlation between supply of cores and demand for remanufactured units or the quantity of returned products. To handle the uncertainties flexibility is needed where remanufacturing environments play their roles since they are characterized by their highly flexible structures (Gungor and Gupta, 1999).

When this profitable and environmentally friendly process, i.e. remanufacturing, is combined with a manufacturing process, this whole system is called a hybrid production system. In this thesis we concentrate on companies that are involved in both remanufacturing and manufacturing activities simultaneously, viz. a hybrid production system is investigated.

An effective production control is key to the competitiveness of the system. The question of when and how much to produce becomes a crucial issue in order to achieve customer satisfaction while keeping low in-process inventories. One way of handling this problem is restricting the search for a production control policy to a class of simple, sub-optimal policies that are easy to implement and try to find out the optimal policy within this class. These simple production control policies have often emerged from actual industrial practice and depend on a small number of parameters. In many of these policies production is triggered by actual demands for end-products (Liberopoulos and Dallery, 2000). Material flow control systems in which production is triggered by actual demands are often referred to as pull control systems.

Much work has been carried out on control policies such as kanban, base stock, generalized kanban and extended kanban, but relatively few comparison studies have been conducted (Duri *et al.*, 2000b). Moreover, these studies are based on a manufacturing system, namely a hybrid remanufacturing / manufacturing system was not considered at all. Besides, an infinite number of raw parts were always supplied for the production system, however, any return flow according to a specific distribution has not been considered and integrated with the production system before.

We focus on material flow control systems in which production is triggered by actual demands that are often referred to as pull control systems. The aim of this paper is to compare performances of kanban, base stock, generalized kanban and extended kanban control systems considered in a hybrid remanufacturing / manufacturing system environment using continuous-time Markov chains (CTMCs). After modeling these control systems as queuing networks with synchronization mechanisms, we obtain steady-state probabilities for each state by expressing the behavior of the queuing network as CTMCs which will then be used for calculating performance measures of interest. Using these performance measures we obtain cost functions for each system and compare them while changing the values of parameters.

The rest of the thesis is organized as follows. Chapter 2 provides an overview of the pull type control systems and remanufacturing literature. Chapter 3 represents the main objectives of the thesis. In chapter 4, after the problem definition, the performance analysis takes place. Validation for the proposed method is stated in this chapter, as well, and then optimization of the parameters is given. Numerical analysis is illustrated in Chapter 5. In the last chapter, conclusions are drawn.

2. LITERATURE REVIEW

The literature review considers studies on both kanban control and remanufacturing areas as our goal is to compare different pull type control policies in a remanufacturing / manufacturing environment. There are two sub-sections, one is related to pull type control policies, and the other is related to remanufacturing literature. There are a number of studies both on pull type control policies and remanufacturing areas, but we only take the studies into consider that can be helpful to our thesis.

2.1. Literature for Pull Type Control Policies

In kanban control, the manufacturing system is decomposed into stages, where each stage is controlled by a kanban mechanism. The number of kanbans for each stage are the parameters of the control policy. After the system has been decomposed into stages, the design of a Kanban Control System (KCS) reduces to setting the kanban sizes for each stage. The efficiency of the KCS depends on these parameters. Analytical methods have been proposed in the literature for performance evaluation of kanban control systems.

Mitra and Mitrani (1990) study a multi-stage kanban system, each stage containing a single station with exponential service time and being saturated with demands. They use an approximation technique based on decomposition of the kanban system into subsystems. Each subsystem corresponds to a particular stage of the original kanban system, and is analyzed exactly. To determine the unknown parameters of each subsystem, an iterative procedure is used.

Di Mascolo *et al.* (1996) present a general purpose analytical method for performance evaluation of a large class of kanban systems. A multi-stage kanban production system where each stage consists of a subsystem of the original production system is considered. With each stage is associated a given number of kanbans. The kanban system is modeled as a queuing network with synchronization mechanisms. They proposed an approximation method where the original kanban system is decomposed into a set of subsystems, each subsystem being associated with a particular stage. Each subsystem is analyzed in isolation using an approximation technique. Afterwards, an iterative procedure is used to determine the unknown parameters of each subsystem. The method is based on a general-purpose technique known as product-form approximation proposed by Baynat and Dallery (1993). The general technique proposed in Baynat and Dallery (1996) is applicable to the queuing network of one-stage kanban control system in which the kanban loop is represented by a class of customers where the queuing network is analyzed using product-form approximation methods.

Baynat *et al.* (2001) present a new way of deriving the analytical method presented in Di Mascolo *et al*. The queuing network of the kanban control system is seen as a multi-class queuing network in which each kanban loop is represented by a class of customers. Thus, they can utilize the general technique proposed in (Baynat and Dallery, 1996) to analyze the multi-class queuing network using product-form approximation methods. The new method is equivalent to that presented in (Di Mascolo *et al.*, 1996). However, the new algorithm is more efficient in terms of computational complexity. Moreover, it is easier to implement, and it is applicable to general kanban systems.

Koukoumialos and Liberopoulos (2005) develop a general purpose analytical approximation method for the performance evaluation of a multi-stage, serial, echelon kanban control system. The original system is decomposed into a set of nested subsystems, each being associated with a particular echelon of stages. Making use of a product-form approximation technique, each subsystem is analyzed in isolation. To determine the unknown parameters of each one, an iterative procedure is used.

Spearman *et al.* (1990), describe a pull control system called CONWIP (Constant Work in Process). Simulation studies are included to give insight into the system's performance. Its practical advantages over push control systems are given. Liberopoulos and Dallery (2000) present a unified framework for pull type control mechanisms in multi-stage manufacturing systems. How these mechanisms are related to each other is reported. Four basic pull control mechanisms, viz. base stock, kanban, generalized kanban and extended kanban are presented. It is argued that on top of any of these mechanisms a local mechanism to control the work-in-process in each stage may be superimposed. Several cases of basic stage coordination mechanisms with stage work-in-process control are presented. Some production control systems appearing in the literature are shown to be equivalent to some of these cases. Dallery and Liberopoulos (2000) introduce a new mechanism for the coordination of multi-stage manufacturing systems, called the Extended Kanban Control System (EKCS). The dynamics of the EKCS are described, in relation to the dynamics of the Generalized Kanban Control System (GKCS). Its advantages over GKCS are discussed. Karaesmen and Dallery (2000) study several pull type control mechanisms, and quantify how good these mechanisms are. Moreover, the structural properties that make them desirable are presented by making use of a two stage model and an optimal control framework. The tradeoffs between single versus multiple control points and service level constraints on the backorders are also analyzed.

Duri *et al.* (2000a) show the equivalence between two methods of the literature which allow the analysis of base stock systems composed of stages in series, and where each stage contains one exponential station. Since the two are equal, they concentrate on only one of them. They propose some extensions in order to study more complex systems. Several examples of base stock systems are designed, as well. Sbiti *et al.* (2003), on the other hand, are interested in base stock controlled assembly systems. Production systems where an assembly operation between components takes place, are considered. They propose an analytical technique for performance evaluation of base stock controlled assembly systems, based on a decomposition method. Steady-state performance measures are provided by this analytical method.

Frein *et al.* (1995) investigate the influence of the design parameters on the efficiency of generalized kanban control policies. They provide insight, as well as general rules, that leads to a better understanding of GKCS, and to help design these parameters. The study relies on a general analytical technique that exists in the literature. Duri *et al.* (2000b) provide qualitative and quantitative comparisons of KCS, Base Stock Control System (BSCS) and GKCS. It is helpful to choose the policy to be implemented in order to control a production system. They study single-product, multi-stage systems with each stage consisting of one or more stations in series. Processing times are exponentially distributed and demand is Poisson. They use approximate analytical methods to rapidly evaluate performance measures. For parameter optimization they utilize enumeration technique.

2.2. Remanufacturing Literature

The inventory control models of hybrid production systems can be classified into two groups, viz. uncapacitated models and capacitated models. These models can be either deterministic or stochastic. Stochastic uncapacitated models can be investigated under periodic review or continuous review policies.

Van der Laan *et al.* (1999) presented one of the first stochastic uncapacitated models for production planning and inventory control in hybrid systems. Here, both the outputs of the manufacturing and remanufacturing process are used to fulfill customer demands. A simple hybrid system which is related to a single component durable product is considered. They compare traditional systems without remanufacturing to push and to pull controlled systems with remanufacturing. Moreover, they showed that the pull control strategy was more cost effective than the push control strategy for inventory systems with return flows.

Teunter *et al.* (2000) proposed a method for setting the holding cost rates in average cost inventory models with reverse logistics. An average cost inventory model where the opportunity cost rate is obtained by multiplying the interest rate by the marginal cost for producing an item, is easy to use for single source inventory systems with only forward logistics. However, the method is no longer straightforward for inventory systems with reverse logistics. In their study, they compare different methods for calculating the opportunity cost rates of returned non-serviceable, remanufactured and manufactured items.

Korugan and Gupta (2001a) study the impact of the differentiation of both the recovery and the sales activities for the remanufactured products from that of the new products. They address the optimal switching problem for end-products capable of replacing each other to satisfy the demand. Then, they define several control policies and compare them with respect to the expected total cost function of the system.

Korugan and Gupta (2001b) consider single-stage pull production control mechanisms. They assume that they have an infinite supply of raw and used materials for both production lines. This system is modeled as a queuing network with a general manufacturing process that supplies a synchronization station. They determine the type of the routing mechanism that decides which sub-process will be informed after a release of a finished product. Then, they perform the Markov chain analysis for each control mechanism where they usually deal with a two-dimensional state space. They propose optimization algorithms for the control parameters with respect to an expected total cost function.

Korugan and Gupta (2001c) they consider a modified single stage pull control mechanism using an adaptive kanban control procedure to control material flow through a hybrid production system with two discrete processes that deal with manufacturing and remanufacturing activities, respectively. This system is modeled as a queuing network. They use a static routing mechanism, to direct the production triggering information after a release of a finished product to the specific sub-process. Then, they perform the Markov chain analysis for the control mechanism in a three-dimensional state space in order to find the performance measures of the system. After defining the expected total cost function for this process with the performance measures calculated, they present an optimal and a near optimal algorithm to calculate the values of the control variables.

The study of Teunter and Vlachos (2002) is based on a single item hybrid production system. They assume that remanufacturing is profitable, and there are more demands than returns. Using simulation, they look at the cost reduction associated with having a disposal option for cores, for a variety of cases.

Teunter *et al.* (2002) propose a new class of inventory strategies, for hybrid systems with slow manufacturing and fast remanufacturing. Numerical experiments show that the optimal strategy in the new class almost always performs better than the optimal strategies in all other classes, namely those based on equal lead times.

Bayındır *et al.* (2003) investigate the conditions on different system parameters such as lifetime of the product, lead time etc. that lead the remanufacturing option to provide cost benefits. To do that, the return ratio is considered as a decision variable. A queuing network is constructed to represent the production environment and the life cycle of parts. Their objective is to determine the system wide order-up-to level and the return ratio such that the total variable manufacturing and remanufacturing costs and inventory costs are minimized.

Aras *et al.* (2004) investigate potential cost savings associated with the effective management of inventories under uncertainties such as variabilities in remanufacturing costs and leadtimes. Their goal is to assess the cost effectiveness of quality-based categorization of returned products. They develop a CTMC model of a make-to-stock production system with remanufacturing. They assume that returned products can be of high, or low quality. Numerical studies point out that categorizing returned products together with appropriate remanufacturing and disposal strategies can lead to significant cost savings.

Nakashima *et al.* (2004) concentrate on product recovery in a remanufacturing system under stochastic variability. They cope with an optimal control problem of such a system which is formulated into a Markov Decision Process. They consider two types of inventories, viz. the actual product inventory in a factory, and the virtual inventory used by customer. The state of the remanufacturing system is defined by both of the inventory levels. They obtain the optimal production policy that minimizes the expected average cost per period. Utilizing design of experiments, some scenarios are considered under various conditions.

Bayındır *et al.* (2007) consider a segmented market for manufactured and remanufactured products. The aim of their study is to investigate the system conditions under which utilization of remanufacturing option provides profit improvement. Their contributions are investigating a model under a capacity constraint with two customer classes, deriving analytical conditions for profitability of remanufacturing, obtaining characterization for behavior of optimal solution with respect to system parameters, investigating risk reducing benefits of substitution policy, and computational analysis to investigate optimal utilization ratio of remanufacturing.

In the literature of remanufacturing, while basic kanban control is considered, the remaining pull type control systems such as BSCS, GKCS and EKCS are not considered. Moreover, a comparison study encompassing all these pull type control systems in a hybrid production environment is not conducted. In this study we consider these issues. In order to model the problem, we use a queuing network approach and pull type control, as pull control is shown to perform better in these systems than push control (Gupta *et al.*, 2004). We adapt Baynat's algorithm to analyze our hybrid production system consisting of a remanufacturing and manufacturing process where the remanufacturing process is fed by an external return flow. None of the studies mentioned above, explains a hybrid production system whose remanufacturing process is fed by an external return flow, namely, in previous studies cores were supposed to be supplied always when necessary. Furthermore, a demand triggering mechanism is provided by means of static routing. External return flow with a static routing mechanism is the main contribution of our study. Besides, only a few number of comparison studies are available in the literature what forces us to conduct cost comparison experiments. We based our cost structure on the study of Teunter *et al*. The adaptation of the enumeration technique proposed by (Duri *et al.*, 2000b) to our system implies the cost comparison and thereby the optimization part of our study.

3. OBJECTIVES

Literature survey points out, that much work has been carried out on control policies such as kanban, base stock, generalized kanban and extended kanban, but relatively few comparison studies have been conducted. This is partly due to the fact that different systems have been described within different frameworks. Moreover, these studies are based on a manufacturing system, namely a hybrid (remanufacturing / manufacturing) system was not considered at all. Besides, an infinite number of raw parts were always supplied for the production system. Hence, a return flow according to a specific distribution has not been considered.

We consider a hybrid production system with random returns. In this setting we compare several pull type control methods with respect to their performance measures. Here, a demand triggering mechanism will be used to direct demands to manufacturing and remanufacturing processes respectively. In this dissertation, our goal is to make a cost comparison among four pull type control policies, viz. KCS, BKCS, EKCS and GKCS. Comparisons will be made with respect to a total cost function that considers the average queue lengths of backorders, finished products, work-in-process levels and cores. The performance measures will be obtained analytically. To this end, an approximation method proposed in the literature will be adapted. The cost function will be optimized with respect to the control parameters before comparison.

4. MODEL

4.1. Problem Definition

For an effective production control the systems cannot deny the meaning of producing the right parts, at the right time, at a competitive cost. In order to meet these objectives, some manufacturers, especially Japanese firms, developed “pull based” production planning and control systems which are often referred to as just-in-time (JIT), kanban or zero inventory (Spearman *et al.*, 1990). Such production control and coordination mechanisms, called also pull type control mechanisms, react to actual occurrences of demand rather than future demand forecasts (Karaesmen and Dallery, 2000). Thus, the release of parts into each stage of the system is coordinated with the arrival of customer demands for final products which generates the problem we are concerned with. In most manufacturing systems production activities are grouped into well identifiable production stages which operate independently of one another and what couples them is the release of items from one stage to the next. Having fewer points to control makes the production control problem simpler and the implementation of a production control policy becomes easier. That is why firms want to aggregate their production activities into stages and control the material flow between these. A stage is a production/inventory system basically in which there exists an input buffer, a manufacturing facility and an output buffer (Liberopoulos and Dallery, 2000).

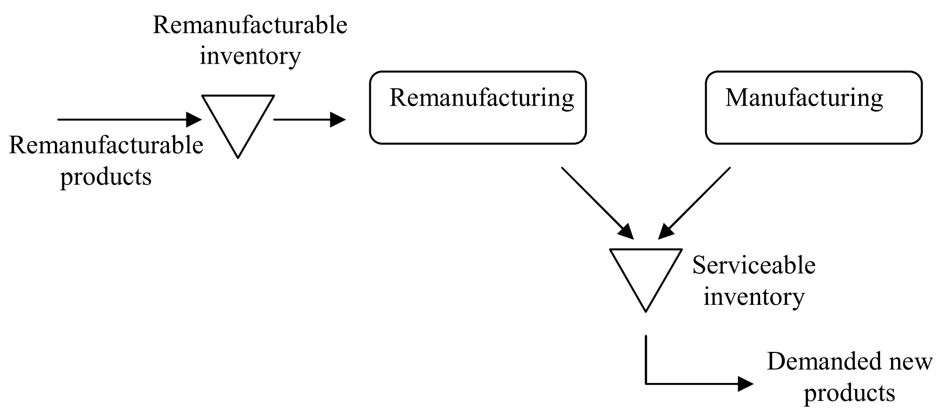

Figure 4.1. A general hybrid production system (Van der Laan *et al.*, 1999)

In a general production control system, there is only one source to feed the output buffer and satisfy the demand. However, a hybrid production system has two input sources to serve customers, one of them is the manufacturing process and the other one is the remanufacturing process which receives cores to process. The cores arrive randomly while the manufacturing process is supplied by raw parts whenever needed. Furthermore, a demand triggering mechanism, viz. the static routing mechanism, has to balance the two production sources.

A hybrid production consisting of remanufacturing and manufacturing processes and stocking points is depicted in Figure 4.1. Remanufacturable products enter the remanufacturable inventory if the current queue length of core buffer is convenient, otherwise they are disposed of. The items available in remanufacturable inventory are to be remanufactured and to be put into the serviceable inventory, however, the output of the remanufacturing process may be insufficient to fulfill all the incoming demands. Therefore, a manufacturing process exists to produce new modules (Van der Laan *et al.*, 1999). In the remainder of the thesis “cores” will be used instead of “returned items” as a matter of convenience.

4.1.1. Kanban Control System (KCS)

In this thesis a hybrid production system consisting of both a remanufacturing process and a manufacturing process is being handled. Either external flow, namely both the return flow and demand arrival, eventuate according to Poisson process. First control policy to be considered is the Kanban Control System (KCS).

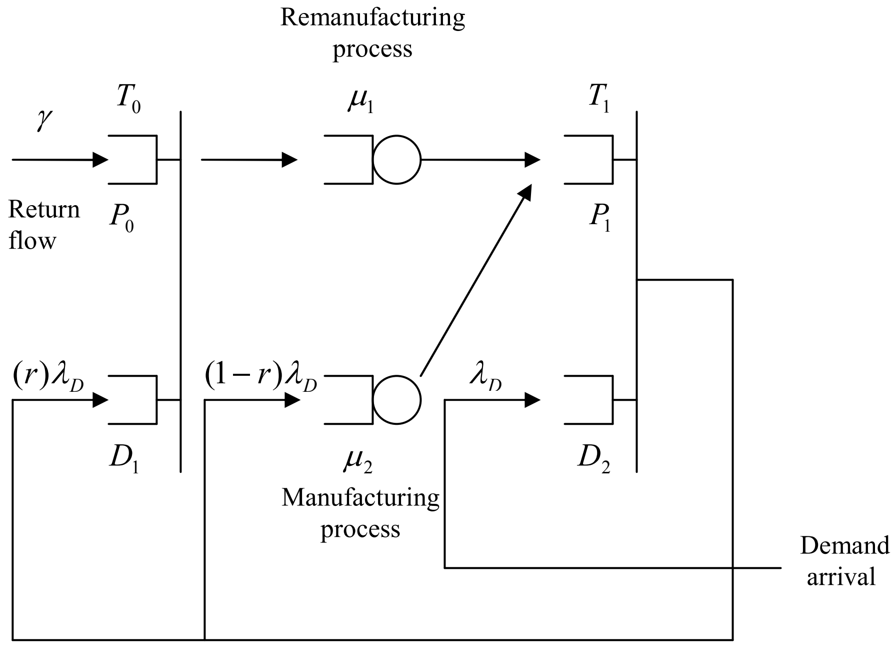

Figure 4.2. Kanban controlled hybrid production system

We have a hybrid production system which consists of a remanufacturing process and a manufacturing process. Upon the arrival of a customer demand, this demand joins the backorder queue (D_2), thereby requesting the release of a finished product from the output buffer (P_1) down to the customer. If a finished product is available in the output buffer, it is released to the customer after liberating the kanban that was attached to it. This kanban is transferred upstream either to the manufacturing process with probability (1-*r*), thereby requesting a finished product via manufacturing process or to the demand queue for cores (D_1) with probability (*r*) carrying along with it a demand for obtaining a finished part via remanufacturing and authorizing the release of a core into the remanufacturing process. As it is obvious, the demand queue of cores contains pairs of demands and remanufacturing authorizations.

Manufacturing process does not need any demand queue in front of it since sufficient raw parts are always supplied whenever needed, however, cores arrive at the system according to Poisson process, i.e. return flow is Poisson distributed with return rate γ. The triggering procedure where the predefined probability mentioned above, viz. routing probability (*r*), remains constant is called "static routing mechanism" (Korugan and Gupta, 2001b). Either part, namely the core or a new raw part, moves downstream the system whereas the demand flow moves upstream the system beginning from the customer up either to the demand queue of cores or to the manufacturing process. The remanufacturing process (node) contains a buffer where the admitted cores await until the remanufacturing server becomes available and a server where they receive service. The manufacturing process (node) contains a buffer where the admitted raw parts await until the manufacturing server becomes available and a server where they receive service. The service-time distribution of each server is represented by an exponential distribution with service-rates of μ_1 and μ_2 for remanufacturing machine and for manufacturing machine, respectively. The output buffer contains finished products, either obtained from remanufacturing or manufacturing process, waiting for demand. The backorder queue (backlog or bulletin board) is where requests for new products are posted (Mitra and Mitrani, 1990). The arrival process of external demands is Poisson with a demand rate of λ_D. The return flow is Poisson distributed with a return rate of γ. The buffer for cores (P_0) has a finite capacity of size *B*. This means that if a core arrives at the system as the current queue length of the cores is *B*, then this core is disposed of. In other words, in order to be able to enter the buffer of cores, the current queue length of the cores must be less than *B*. Backorder queue is assumed to be infinite, thus, it can grow up to infinity. The hybrid production system has a fixed number of kanbans represented as *K*.

The kanban system described above can be modeled as a queuing network with synchronization mechanisms. The behavior of this kanban system can be represented by the queuing network shown in Figure 4.2. The remanufacturing process and manufacturing process can be modeled by a sub network in which each machine concerned is represented by a station of the sub network. At the input and output of the hybrid production system the kanban control policy is modeled by a synchronization station.

The first synchronization station T_0 represents the synchronization between a core and a free kanban in the hybrid production system whereas the second synchronization station T_1 represents the synchronization between a finished product and an external demand. Let P_0 and D_1 denote the two upstream queues of the synchronization station T_0. The content of queue P_0 is the number of cores. The content of queue D_1 is the number of free kanbans of the hybrid production system. The behavior of the synchronization station is as follows. As soon as there is a core in queue P_0 and a kanban in queue D_1, the kanban is attached to the core and is sent to join the buffer of the remanufacturing process to receive service.

Station T_1 represents the synchronization between the finished products of the system and the external demands. Let P_1 and D_2 denote the two upstream queues of the synchronization station T_1. The content of queue P_1 is the number of finished products currently available at the output of the hybrid production system. The content of queue D_2 is the number of demands currently backordered. As soon as there is an entity in queue P_1 (representing a finished product of the system on which a kanban is attached) and an entity in queue D_2 (representing an external demand), a demand is satisfied and the liberated kanban joins either queue D_1 with a routing probability (*r*) or immediately enters the manufacturing process with a routing probability (1-*r*) (Di Mascolo *et al.*, 1996).

4.1.2. Base Stock Control System (BSCS)

One of the best known pull controlled production systems encountered in the literature is the Base Stock Control System (BSCS) which functions with a logic of make to stock production. Such a system uses the big advantage of its great reactivity, i.e. when an external demand arrives, it is instantaneously transmitted to each stage. BSCS has one adjustment parameter per stage that represents the initial level of finished products for the considered stage (Sbiti *et al.*, 2003). In a BSCS applied to a hybrid production system which has a static routing triggering mechanism, as in our case, information on final demand is made available at either the remanufacturing process by triggering demand queue of cores (D_1) with probability (*r*) or at the manufacturing process with probability (1-*r*). BSCS applied to a hybrid production system depends on only one parameter per stage that determines the target in terms of the number of products to be produced and stored at the output of the whole system. We limit our study to a hybrid production system composed of a remanufacturing process and a manufacturing process and producing only one type of product. We assume that there occurs a return flow according to Poisson process with a return rate of γ. However, there are always raw parts in front of the manufacturing process so that no demand queue is necessary for manufacturing process. We also assume that the external demands are backordered when no finished product is present in the output buffer. External demands are Poisson distributed with rate λ_D. Service times are exponentially distributed with rates μ_1 and μ_2, for remanufacturing machine and manufacturing machine, respectively.

A queuing network model with synchronization stations is shown in Figure 4.3. Queue P_1 represents the output buffer for finished products, no matter which process they are received from. Queue D_2 contains demands for the production of new finished parts. Queue P_0 is where cores await removal until a demand arrives at queue D_1 requesting the release of a core into the remanufacturing process. When the system is in its initial state, that is, before any demands have arrived to the system, P_0 contains an initial number of cores (*B*) and P_1 contains *S* finished parts, all other queues in the system are empty. *S* is the only control parameter of the system and is referred to as the base stock of the system (Liberopoulos and Dallery, 2000).

The first synchronization station named T_0 is composed of two queues, one containing the cores (P_0) and the other (D_1) containing demands for cores. The second (last) synchronization station represented as T_1 is composed of two queues, one containing the finished products received from either process (P_1) and the other (D_2) containing demands for these end-products. The purpose of BSCS is to satisfy demands and to lead P_1 to its maximum level *S*. Thus, if the demand arrival process stops and if we wait long enough, the final state of the system will be: *S* products in queue P_1, *B* cores in P_0 and the other queues are empty (Duri *et al.*, 2000a).

BSCS operates as follows. Upon the arrival of a customer demand, this demand joins the backorder queue (D_2), thereby requesting the release of a finished product from the output buffer (P_1) down to the customer. If a finished product is available in the output buffer, it is directly released to the customer. The customer demand instantly also generates a demand in either D_1 that authorizes the release of a core from P_0 to the remanufacturing process with probability (*r*) or this demand can also be generated for the manufacturing process with probability (1-*r*). As it is obvious, the demand queue of cores (D_1) contains only demands for remanufacturable items and no authorizations. Demand generation behavior of the BSCS leads to an immediate start either for working on a core which it pulls from the buffer of cores or for working on a new raw part to be manufactured, no matter whether it is necessary or not. Therefore, it becomes urgent that WIP levels grow up very rapidly without any limit (Liberopoulos and Dallery, 2000).

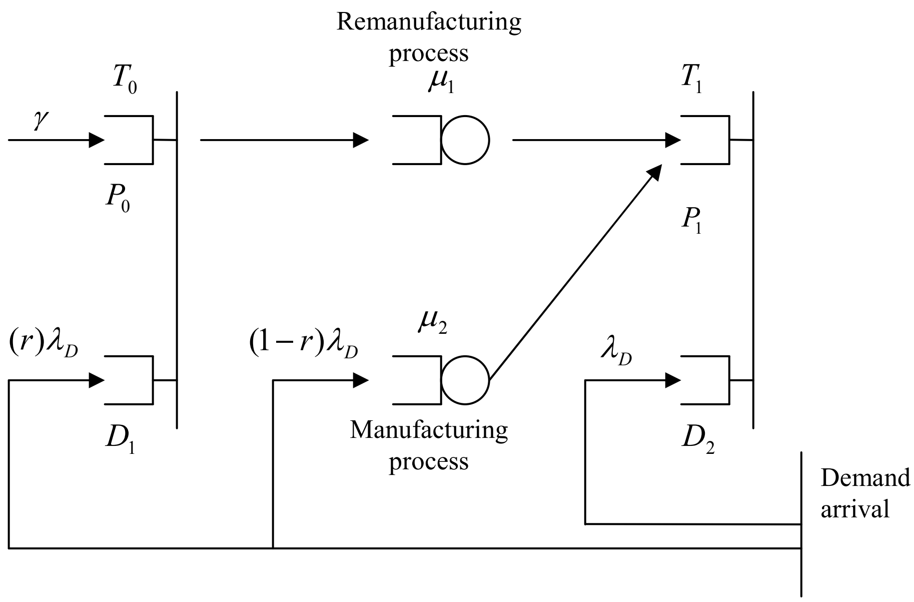

Figure 4.3. Base stock controlled hybrid production system

4.1.3. Generalized Kanban Control System (GKCS)

Generalized Kanban Control System (GKCS) is basically designed to combine the advantages of KCS and BSCS. Rapid reaction to demand captured in BSCS and better coordination and controlled work in process inventories achieved in KCS are features that are combined to develop a GKCS. This leads GKCS to an urgency for borrowing the idea of safety stocks from the BSCS and production authorization cards from the KCS, so that GKCS is defined by two parameters per stage, one defining the safety stocks (*S*) and the other the number of production authorization cards (*K*) (Karaesmen and Dallery, 2000).

A GKCS applied to a hybrid production system can be modeled as a queuing network with synchronization mechanisms as illustrated in Figure 4.4. The first synchronization station (T_0) consists of two upstream queues, i.e. P_0 and D_1. Entities in queue P_0 represent the buffer of cores while entities in queue D_1 imply demands for remanufacturing of cores. The second synchronization station (T_1) is composed of two synchronization stations, each having two upstream queues: P_1 and A_2 make up the first part while D_2 and B_1 form the second one. Entities in queue P_1 represent the inventory of finished products of the system. Entities in queue A_2 represent the number of backordered demands. Entities in queue D_2 represent demands for production of new parts by the hybrid production system, and queue B_1 includes free kanbans and represents authorizations to transfer demands either to queue D_1 or to the manufacturing process.

GKCS can be regarded as an extension of KCS by noting that queue P_1 in KCS is split into two queues in GKCS, namely P_1 and B_1, and queue D_2 in KCS is split into two queues in GKCS, namely D_2 and A_2. This matter leads to an advantage of GKCS, namely the transfer of products throughout the system and the transfer of demands on the way upstream can be done independently of one another (Frein *et al.*, 1995).

The behavior of GKCS is as follows. As soon as there is one core in queue P_0 and one kanban in queue D_1, a core / kanban pair is transferred to the remanufacturing process where it will be treated. Again- since after a kanban is freed, it returns to either to queue D_1 or to the manufacturing process due to static routing mechanism, and since sufficient raw parts are always available for being manufactured- as soon as there is one kanban available for manufacturing, a raw part / kanban pair is transferred to the manufacturing process where it will be treated. When either process is complete, the kanban is separated from the product to join queue B_1 and the product enters into queue P_1 (Duri *et al.*, 2000b).

Upon the arrival of an external demand, this demand is split into two entities. One of these entities joins queue A_2 and enables the consumption of a finished product from queue

P_1. The other one joins queue D_2 in order to be transferred upstream. If there is at least one finished product in queue P_1, the demand is immediately satisfied. Otherwise, the demand cannot be satisfied immediately and is backordered in queue A_2 until a finished product becomes available. Either immediately or after being backordered, when a demand is satisfied, a finished product leaves the system and a backordered demand is removed from queue A_2. On the other hand, if there is at least one free kanban in queue B_1 on the demand's arrival, the demand is immediately transferred either to queue D_1 with probability (*r*) or to the manufacturing process with probability (1-*r*). If there is no free kanban in queue B_1, the demand cannot be immediately transferred and waits in queue D_2 until a kanban is available. When available, the demand / free kanban pair is transferred either to queue D_1 with probability (*r*) or to the manufacturing process with probability (1-*r*) (Frein *et al.*, 1995). If we stop the arrival process of external demands and wait long enough, queue P_1 will contain parts while queue B_1 will contain kanbans, and all other queues will be empty**.** Bringing queue P_1 back to its maximum level while satisfying the demands and controlling the WIP is the function of GKCS (Duri *et al.*, 2000b).

Here, we have the following assumptions. We have a return flow according to the Poisson process where items to be remanufactured return with a return arrival rate of γ to the system. External demands arrive at the hybrid production system according to Poisson process with arrival rate λ_D. The service-time distributions of both remanufacturing machine and manufacturing machine are exponential, and their corresponding service-rates are μ_1 and μ_2, respectively. Sufficient parts are always supplied for manufacturing process so that no demand queue is necessary for manufacturing process. We also assume that the external demands are backordered when no finished product is present in the output buffer.

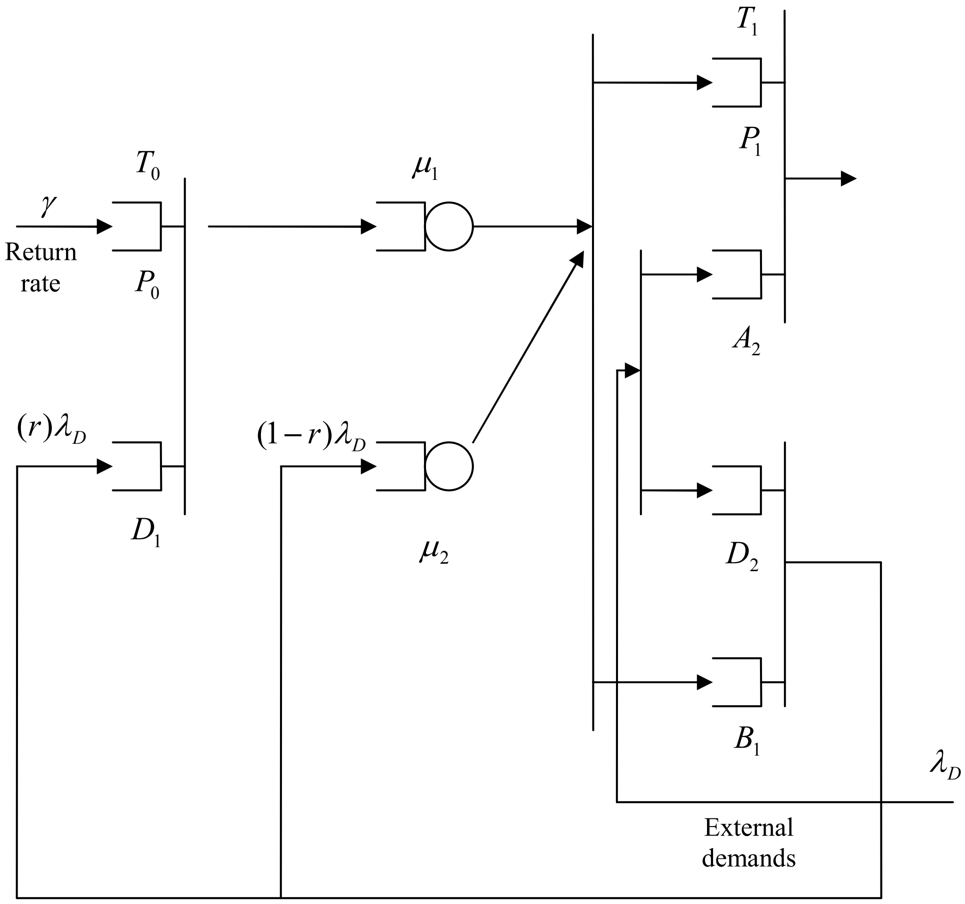

Figure 4.4. Generalized kanban controlled hybrid production system

4.1.4. Extended Kanban Control System (EKCS)

Extended Kanban Control System (EKCS) can be regarded as an integration of conventional KCS and BSCS in the sense that it is defined by two parameters per stage, namely the number of kanbans, *K*, and the base stock of parts in inventory, *S*. The idea behind EKCS is to capture the advantages of both systems (Ebisuya *et al.*, 2004). EKCS seems to be similar to GKCS in terms of its multi-parameter structure. However, it is an extension of GKCS. EKCS is a control system easier to apply in comparison to GKCS. Furthermore, parts in an EKCS are easier to trace as a result of having kanbans attached to them, whereas in the GKCS finished parts are separated from their kanbans.

To control our hybrid production system with an EKCS we have kanbans dedicated for both remanufacturing and manufacturing processes that flow throughout the system.

However, the demand triggering procedure is static routing, i.e. demands return with a predefined probability either to remanufacturing or manufacturing process. This system can be modeled as a queuing network with synchronization stations whose queues concerned will be noted next. The reason why we built a model of a system with dedicated kanbans and static routing mechanism is, that without using dedicated kanbans, our system did not function, namely depending on the case, we observed that kanbans were collected either before remanufacturing process or manufacturing process, following the animation movements of our simulation model. Thus, the convergence of average queues did not occur, and we could not obtain system parameters. So, we decided on a system where we used dedicated kanbans instead.

As depicted below (Figure 4.5), queue A_R contains free kanbans for remanufacturing process. Queue P_0 represents the buffer for cores. Queue D_R contains demands for obtaining finished products via remanufacturing. These three upstream queues make up the first synchronization station named T_R.

Queue A_M contains free kanbans for manufacturing process. Queue D_M contains demands for obtaining finished products via manufacturing. These two upstream queues make up the second synchronization station named T_M.

Queue P_1 represents the output buffer for finished products whose visitors are pairs of finished products and production authorizations. Queue D_2 nestles external (customer) demands. These two upstream queues form the last synchronization station named T_1.

In total, there are $K_R + K_M$ kanbans in the system where K_R of these are dedicated for remanufacturing process and K_M of these are dedicated for manufacturing process. When the system is in its initial state, that is, before any demands have arrived to the system, P_0 contains an initial number of cores (*B*), P_1 contains *S* finished products which can be obtained as the summation of S_R and S_M, each product having either a remanufacturing kanban or a manufacturing kanban attached to it. S_R represents finished products obtained via remanufacturing, whereas S_M represents finished products obtained via manufacturing. A_R contains K_R - S_R free kanbans, and A_M contains K_M - S_M free kanbans, all other queues in the system are empty.

The behavior of the system is as follows. Upon the arrival of an external demand, a finished product from queue P_1 is consumed if there is any. Otherwise, this demand waits in queue D_2, in other words, it is backordered until a finished product becomes available. The arrival of an external demand also generates a demand in D_R with probability (*r*) or in D_M with probability (1-*r*), as is the case in the BSCS. A core is not immediately released from P_0 into the remanufacturing process unless there is a free kanban in A_R. The same logic is valid for manufacturing process, too, that is, if an external demand generates a demand in D_M with probability (1-*r*), a free kanban must be available in A_M in order a raw part to enter the manufacturing process. In short, we can say, that to enter any process three entities are needed, a part to be processed, a free kanban to be attached and a demand for the part. After being processed with a service rate of either μ_1 or μ_2 by the remanufacturing machine and the manufacturing machine, respectively, the finished products are stored in the output buffer P_1 where they await removal to be consumed by the customers.

In the limiting case, when $K = \infty$, the EKCS is equivalent to the BSCS with the same base stock parameter *S* as that of the EKCS (Dallery and Liberopoulos, 2000). This property will be helpful to solve the BSCS analytically.

The assumptions for this system are as follows. The external demand arrival and the return flow occur according to Poisson process with respective rates λ_D and γ. Service rates of remanufacturing machine and manufacturing machine are exponentially distributed with service rates μ_1 and μ_2, respectively. Demand triggering mechanism implies static routing, that is, the demand generated arrives at D_R with probability (*r*) or in D_M with probability (1-*r*). We use kanbans that are dedicated for both remanufacturing and manufacturing processes.

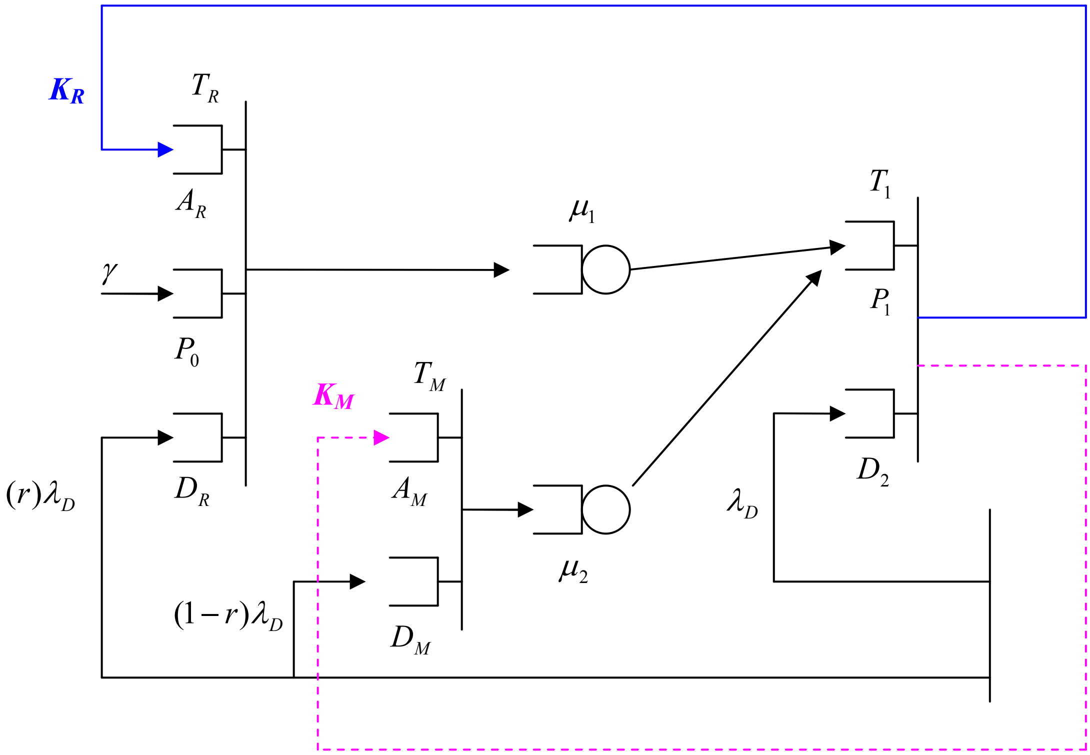

Figure 4.5. Extended kanban controlled hybrid production system

4.2. Performance Evaluation

For performance evaluation of pull type control systems analytical methods have been proposed. One of these methods which is fairly accurate is presented by Di Mascolo *et al.* (1996). In their work, the system is modeled as a queuing network with synchronization stations that again uses "decomposition" idea and product-form approximation to obtain performance measures as previously proposed in (Baynat and Dallery, 1993). However, that the computational algorithm involves two levels of iterations becomes its major drawback since Baynat and colleagues managed to avoid these two levels, so that the new algorithm became much more efficient. This latter method is again based on product-form approximation and it is equivalent to that proposed in (Di Mascolo *et al.*, 1996) in terms of equations. The general technique proposed in (Baynat and Dallery, 1996) is applicable to the queuing network of one-stage kanban control system in which the kanban loop is represented by a class of customers where the queuing network is analyzed using product-form approximation methods. As mentioned above, besides the reduction of the computational complexity, the new method is simpler to describe and implement. That it is applicable to more general kanban systems, is another advantage of this method over that proposed in (Di Mascolo *et al.*, 1996).

4.2.1. Kanban Control System (KCS)

We consider our single-stage hybrid production system as a queuing network with synchronization stations. This system consists of four stations totally, i.e. two synchronization stations at the input and output, one station for remanufacturing and one station for manufacturing. Let *K* be the total number of customers (kanbans) of our one-stage hybrid production system. The set of indexes of the stations that are visited by class-*r* customers where $r = 1$, is represented by $S(1) = \{1,2,3,4\}$. The idea of this method is to associate with the original network, *R* single-class product form networks which represent the number of classes (stages), where *R* becomes 1 in our case. For each station *i* of the original network visited by its customers, we associate a load-dependent exponential service station, as depicted in Figure 4.6. Let $\mu_i(n_i)$, $n_i = 1,...,K$, $i = 1,2,3,4$ denote the load-dependent service rates of station *i* for the single-class product-form network. The visit ratios are simply calculated and are as follows: $V_1 = r$, $V_2 = r$, $V_3 = 1 - r$ and $V_4 = 1$. The single-class network is known as Gordon-Newell network (Baynat *et al.*, 2001), so that we have the opportunity of utilizing product-form solution to get the steady-state probabilities $P(n)$ for the network. The product-form solution is obtained by means of the below formula (Formula 4.1). G is the normalization constant associated with the one-stage network. The normalization constant approach can be found in Bruell and Balbo's study. In a closed queuing network like KCS, a fixed number of customers (*K*) circulate through the network at all times. Here, the customers have four stations to be visited. So, the state of such a network can be described by a vector $n = (n_1, n_2, n_3, n_4)$ where n_i represents the current number of customers present at the i th facility ($\sum_{i=1}^{4} n_i = K$).

$$P(n_1, n_2, n_3, n_4) = \frac{1}{G} \prod_{i \in S(1)} \left[\prod_{n=1}^{n_i} \frac{V_i}{\mu_i(n)} \right] \tag{4.1}$$

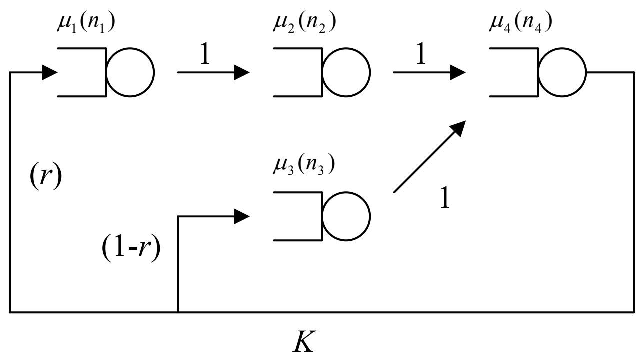

Figure 4.6. One-class product-form network

The first level approximation of the method happens where the performance of the single class of customers in the original network is approximated by the performance of the single class product-form network. It remains to specify the load-dependent service rates $\mu_i(n_i)$, $n_i = 1,...,K$, $i = 1,2,3,4$. These service rates of the flow equivalent service centers of the product-form network should be equal to the conditional throughputs $v_i(n_i)$ of class-1 customers (due to one-staged system) at the corresponding station in the original network. What we mean by conditional throughput $v_i(n_i)$ in our hybrid production system, is the average flow of the system's customers out of station *i*, given that n_i customers are present at the station. Since exact values of the conditional throughputs require an exact solution of the original system, a second level of approximation is needed where conditional throughputs are approximated. In order to approximate the conditional throughputs of our one-class at every station in the original system, the stations are analyzed in isolation.

Each station *i* is analyzed as a queue fed by external arrival processes, as depicted in Figure 4.7. The arrival process of customers of our one-class is assumed to be a state-dependent Markovian process with rates $\lambda_i(n_i)$, $n_i = 0,...,K-1$, where n_i is the current number of customers present at station *i*.

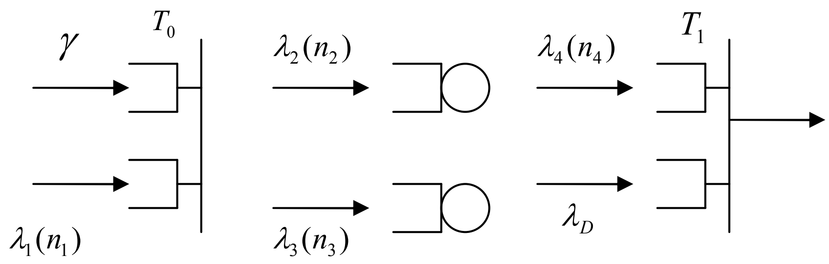

Figure 4.7. Decomposition of the network to the isolated stations with load dependent arrival rates

We first suppose that $\lambda_i(n_i)$ are given, such that $i \in S(1)$ and $n_i = 0,...,K-1$. Station *i* can be analyzed in isolation using continuous-time Markov chains (CTMCs). The underlying Markov chain depicted below (Figure 4.8) represents the behavior of the first synchronization station T_0. This synchronization station is fed by two Markovian arrival processes with state-dependent arrival rates on one hand, and with external resources on the other. Since the arrival process is Poisson, the associated Markov chain has a linear structure. The state of this CTMC is (n_1, n_r), where n_1 is the current number of customers (kanbans) and n_r is the current number of external resources (cores to be remanufactured). Since the buffer size of the cores is *B*, n_r cannot exceed this limit. The steady-state probabilities $p(n_1, n_r)$ can be obtained by means of the following balance equations:

$$p(n_1,0)\,\gamma = p(n_1-1,0)\,\lambda_1(n_1-1) \quad \text{for } n_1 = 1,...,K \tag{4.2}$$

$$p(0,n_r)\,\lambda_1(0) = p(0,n_r-1)\,\gamma \quad \text{for } n_r = 1,...,B \tag{4.3}$$

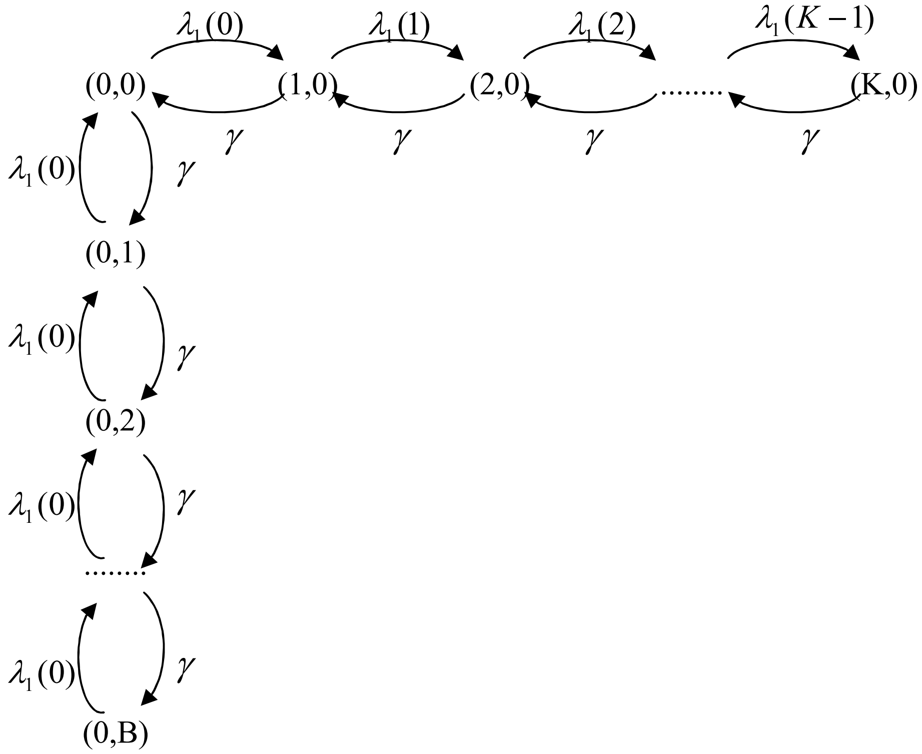

Figure 4.8. Markov chain representing the behavior of the first synchronization station T_0

Since we are actually interested in marginal probabilities, their calculations will be mentioned next. The marginal probabilities can be expressed as:

$$\widetilde{P}_1(n_1) = p(n_1,0) \qquad \text{for } n_1 = 1,...,K \tag{4.4}$$

$$\widetilde{P}_1(0) = \sum_{n_r=0}^{B} p(0,n_r) \tag{4.5}$$

The expressions above imply that:

$$\widetilde{P}_1(n_1) = \frac{\prod_{i=0}^{n_1-1} \lambda_1(i)}{\gamma^{n_1}} p(0,0) \qquad \text{for } n_1 = 1,...,K \tag{4.6}$$

$$\widetilde{P}_1(0) = \frac{\left(\dfrac{\gamma}{\lambda_1(0)}\right)^{B+1} - 1}{\dfrac{\gamma}{\lambda_1(0)} - 1}\, p(0,0) \tag{4.7}$$

In other words, the marginal probabilities $\widetilde{P}_1(n_1)$ that n_1 customers are present in the isolated station for $n_1 = 0,...,K$ have been derived. The iterative method uses these marginal probabilities to estimate the conditional throughputs $\widetilde{v}_i(n_i)$ such that:

$$\widetilde{v}_i(n_i) = \lambda_i(n_i - 1)\frac{\widetilde{P}_i(n_i - 1)}{\widetilde{P}_i(n_i)} \quad \text{for } n_i = 1,...,K \tag{4.8}$$

Here, we have a hybrid production system that consists of only one stage. So, to simplify the formula or the figures, these subscripts are omitted. Since *i* stands for station number, *i* can take values from one to four.

The load-dependent service rates of the associated stations in the single-class product-form network are set equal to these estimated conditional throughputs, i.e.:

$$\mu_i(n_i) = \widetilde{v}_i(n_i) \quad \text{such that } i \in S(1) \text{ and } n_i = 1,...,K \tag{4.9}$$

The state-dependent arrival rates $\lambda_i(n_i)$ of one-class customers at the different stations are required in order to use this approach. These quantities can be obtained from the product-form solutions of the single-class product-form network given by Formula 4.1.

$$\lambda_i(n_i) = \mu_i(n_i + 1)\frac{P_i(n_i + 1)}{P_i(n_i)} \quad \text{for } n_i = 0,...,K-1 \tag{4.10}$$

The last synchronization station T_1 can be analyzed using the same logic where the state of the CTMC is (n_4, n_D) where n_4 represents the current number of customers (kanbans) and n_D is the current number of external resources (external demands). Since the backorder queue has no limit, this queue can grow up to infinity. The underlying

Markov chain and its analysis are given below (Figure 4.9.). The steady-state probabilities $p(n_4, n_D)$ can be obtained by means of the following balance equations:

$$p(n_4,0)\,\lambda_D = p(n_4-1,0)\,\lambda_4(n_4-1) \qquad \text{for } n_4 = 1,\dots,K \tag{4.11}$$

$$p(0,n_D)\,\lambda_4(0) = p(0,n_D-1)\,\lambda_D \qquad \text{for } n_D = 1,\dots,\infty \tag{4.12}$$

The marginal probabilities can be expressed as:

$$\widetilde{P}_4(n_4) = p(n_4,0) \qquad \text{for } n_4 = 1,\dots,K \tag{4.13}$$

$$\widetilde{P}_4(0) = \sum_{n_D=0}^{\infty} p(0,n_D) \tag{4.14}$$

The expressions above imply that:

$$\widetilde{P}_4(n_4) = \frac{\prod_{i=0}^{n_4-1}\lambda_4(i)}{(\lambda_D)^{n_4}}\,p(0,0) \qquad \text{for } n_4 = 1,\dots,K \tag{4.15}$$

$$\widetilde{P}_4(0) = \frac{1}{1-\dfrac{\lambda_D}{\lambda_4(0)}}\,p(0,0) \tag{4.16}$$

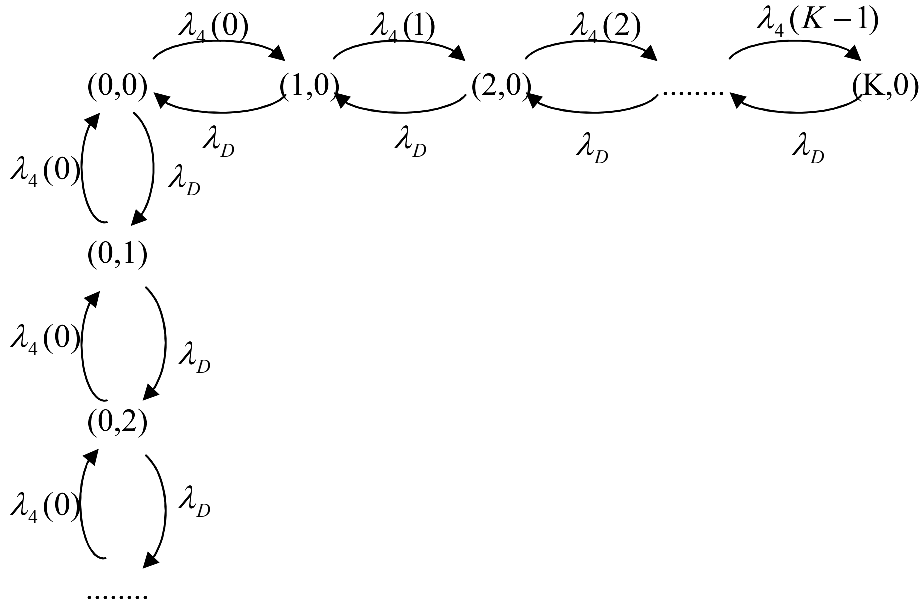

Figure 4.9. Markov chain representing the behavior of the last synchronization station T_1

For the analysis of the second and third stations (remanufacturing and manufacturing process respectively) we do not need to solve Markov chains since the conditional throughputs are simply equal to the load-dependent service rates in the special case where the service times are exponentially distributed (Koukoumialos and Liberopoulos, 2005).

To sum up the whole procedure, marginal probabilities obtained by product-form solution and normalization constant approach are used to obtain state-dependent arrival rates, which are then utilized to estimate conditional throughputs after deriving the marginal probabilities from the solution of CTMCs. The load-dependent service rates are set equal to these estimated conditional throughputs next. The whole story is repeated until a required level of tolerance is captured in convergence of the load-dependent service rates. The algorithm in Table 4.1 presents this process concisely.

After running the algorithm and obtaining convergence, the necessary performance parameters such as the average queue length of cores, the average queue length of finished products, the average queue length of backorders, average work in process can be calculated. The calculations are made by means of expected values which are given below.

Table 4.1. The algorithm

Algorithm:	
Step 0	Set $\mu_i(n_i)$, for $i \in S(1)$ and $n_i = 1,...,K$, to some initial values.
Step 1	a. Using product-form solution technique, calculate the marginal probabilities $P_i(n_i)$, for $i \in S(1)$ and $n_i = 0,...,K$, in the single class network. b. Derive the arrival rates $\lambda_i(n_i)$ for $i \in S(1)$ and $n_i = 0,...,K-1$, from relation (4.10).
Step 2	For $i = 1,2,3,4$: a. Analyze station *i* in isolation by solving corresponding CTMCs. b. Calculate the marginal probabilities $\widetilde{P}_i(n_i)$ such that $i \in S(1)$ and $n_i = 0,...,K$. c. Get the conditional throughputs $\widetilde{v}_i(n_i)$ such that $i \in S(1)$ and $n_i = 1,...,K$, using relation (4.8).
Step 3	Set the load-dependent service rates of station *i* in the single-class network equal to the conditional throughputs, for $i \in S(1)$: $\mu_i(n_i) = \widetilde{v}_i(n_i)$
Step 4	Repeat these steps beginning with Step 1 until convergence of parameters $\mu_i(n_i)$ is achieved for a specified level of tolerance.

The average queue length of cores (QR) is calculated as shown below:

$$QR = \sum_{n_1=1}^{B} n_1 \, \widetilde{R}_1(n_1) \ \text{where} \tag{4.17}$$

$$\widetilde{R}_1(n_1) = \left(\frac{\gamma}{\lambda_1(0)} \right)^{n_1} p(0,0) \ \text{for} \ n_1 = 1,\dots,B \tag{4.18}$$

The average queue length of finished products (QFP) is calculated as below:

$$QFP = \sum_{n_4}^{K} n_4 \, \widetilde{P}_4(n_4) \tag{4.19}$$

The average queue length of backorders (QD) is given as (Di Mascolo *et al.*, 1996):

$$QD = \frac{1}{\frac{\lambda_4(0)}{\lambda_D} - 1} \widetilde{P}_4(0) \tag{4.20}$$

Average work-in-process levels are calculated again by using the expected value approach:

$$WIPR = \sum_{n=1}^{K} n \, \widetilde{P}_2(n_2) \tag{4.21}$$

$$WIPM = \sum_{n=1}^{K} n \, \widetilde{P}_3(n_3) \tag{4.22}$$

4.2.2. Generalized Kanban Control System (GKCS)

Our single-stage hybrid production system controlled by means of GKCS is modeled as a queuing network with synchronization stations as mentioned before. This system consists of four stations totally, i.e. two synchronization stations at the input and output, one station for remanufacturing and one station for manufacturing. *K* denotes the total number of customers (kanbans) associated with the one-stage hybrid production system whereas *S* denotes the maximum number of finished products in this system. The idea used to obtain the performance parameters of the GKCS that is applied to single-stage hybrid production system is the same with that of the conventional KCS. Since the single-class network is known as Gordon-Newell network, we have again the opportunity of utilizing product-form solution to get the steady-state probabilities as is the case in the KCS. One-class product-form network is not different from that of KCS.

In order to approximate the conditional throughputs of our one-class at every station in the original system, the stations are analyzed in isolation. Each station *i* is analyzed as a queue fed by external arrival processes, as is the case in KCS. Only the last synchronization station has a different structure, all the other stations including the first synchronization station and both processes behave the same as in the KCS. Therefore, only the last synchronization station is depicted below in the context of isolated analysis. After the convergence of the analytical method we are informed about the load-dependent arrival rates of kanbans in queue B_1 denoted by $\lambda(n_B)$. The last synchronization station is made up of two synchronization stations which contain two queues, namely, P_1 and A_2 on the one hand, D_2 and B_1 on the other. The contents of these queues have already been explained in the previous chapter.

In order to get the performance measures, we must evaluate $P(n_B = n)$ which corresponds to the probability of having *n* free kanbans. To model the behavior of the station represented in Figure 4.11 the corresponding Markov chain is constructed. The state vector is chosen as (n_B, n_D, n_A, n_P), whose dimensions represent the current number in the free kanban queue, in the demand queue, in the backorder queue and finished product queue, respectively. Considering the values of *K* and *S*, two cases must be distinguished for the construction of the Markov chain (Duri *et al.*, 2000b).

For the first case where $K \geq S$ shown in Figure 4.10, the first quadrant of the coordinated system (quadrant at the top and on the right) includes the state set $(K - S - n_A, 0, n_A, 0)$ where $0 \leq n_A \leq K - S$, the second quadrant of the coordinated system (quadrant at the top and on the left) includes the state set $(0, n_A - K + S, 0, n_A, 0)$ where $n_A \geq K - S$, and the fourth quadrant of the coordinated system (quadrant at the bottom and on the right) includes the state set $(n_B, 0, 0, n_B - K + S)$ where $K - S \leq n_B \leq K$.

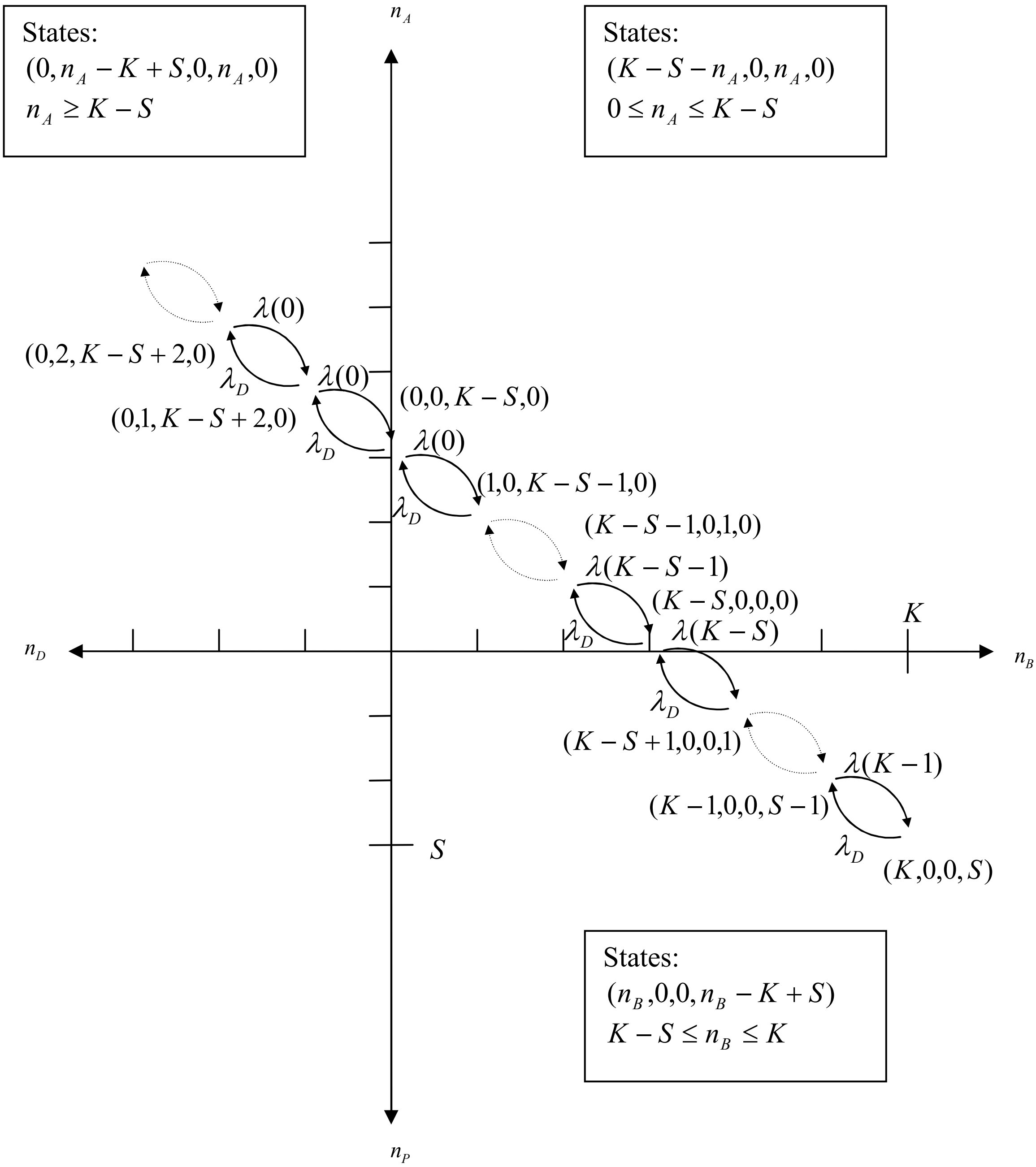

Figure 4.10. Markov chain associated with the last synchronization station in case $K \geq S$ (adapted from Duri *et al.*, 2000b)

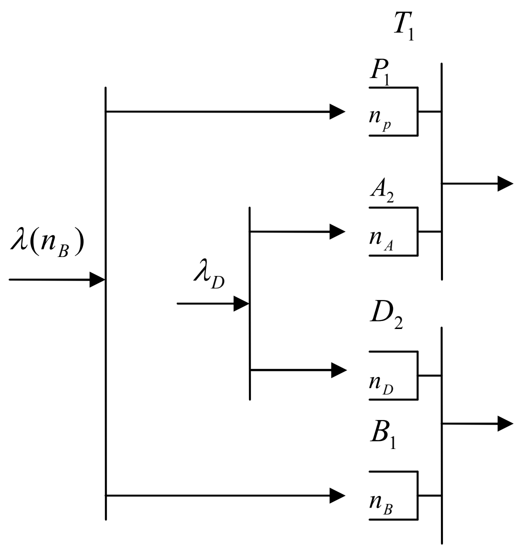

Figure 4.11. Isolated version of the last synchronization station with load dependent arrival rates

Just as for KCS, after writing the balance equations and all the states of the Markov chain as functions of $P(0,0,K-S,0)$ and using the property that the sum of the probabilities of all possible states must be equal to one, we deduce:

$$P(0,0,K-S,0)=\left(\frac{1}{1-\frac{\lambda_D}{\lambda(0)}}+\sum_{n=1}^{K}\frac{\prod_{i=0}^{n-1}\lambda(i)}{(\lambda_D)^n}\right)^{-1} \tag{4.23}$$

$$\widetilde{P}(n_B=n)=\frac{\prod_{i=0}^{n-1}\lambda(i)}{(\lambda_D)^n}P(0,0,K-S,0) \qquad \text{for } n=1,\dots,K \tag{4.24}$$

$$\widetilde{P}(n_B=0)=\frac{1}{1-\frac{\lambda_D}{\lambda(0)}}P(0,0,K-S,0) \tag{4.25}$$

For calculating the average backorder queue length, the probability $P(n_P = 0)$ is needed, and for calculating the average finished product queue the probability $P(n_P = n)$ is needed. Therefore, the following probabilities must be calculated in addition.

$$\widetilde{P}(n_P = n) = \frac{\prod_{i=0}^{K-S+n-1} \lambda(i)}{(\lambda_D)^{K-S+n}} P(0,0,K-S,0) \quad \text{for } n = 1,\ldots,S \tag{4.26}$$

$$\widetilde{P}(n_P = 0) = 1 - \sum_{n=1}^{S} P(n_P = n) \tag{4.27}$$

For the second case where $K < S$ shown in Figure 4.12, the second quadrant of the coordinated system (quadrant at the top and on the left) includes the state set $(0, S-K+n_A, n_A, 0)$ where $n_A \geq 0$, the third quadrant of the coordinated system (quadrant at the bottom and on the left) includes the state set $(0, S-K-n_P, 0, n_P)$ where $0 \leq n_P \leq S-K$, and the fourth quadrant of the coordinated system (quadrant at the bottom and on the right) includes the state set $(n_P - S + K, 0, 0, n_P)$ where $S-K \leq n_P \leq S$. The underlying Markov chain is represented in Figure 4.12 below.

The steady-state probabilities of this linear formed Markov chain are obtained by means of balance equations, and after that the marginal probabilities are calculated since they are needed to get the performance parameters.

$$P(0, S-K, 0, 0) = \frac{\left(\frac{\lambda_D}{\lambda(0)}\right)^{S-K}}{\frac{1}{1-\frac{\lambda_D}{\lambda(0)}} + \sum_{n=1}^{K} \frac{\prod_{i=0}^{n-1} \lambda(i)}{(\lambda_D)^n}} \tag{4.28}$$

$$\widetilde{P}(n_B = n) = \frac{\prod_{i=0}^{n-1} \lambda(i)}{(\lambda_D)^n} \left(\frac{\lambda(0)}{\lambda_D}\right)^{S-K} P(0, S-K, 0, 0) \quad \text{for } n = 1,\ldots,K \tag{4.29}$$

$$\widetilde{P}(n_B = 0) = \left(\sum_{j=1}^{S-K} \left(\frac{\lambda(0)}{\lambda_D} \right)^j + \frac{1}{\frac{\lambda(0)}{\lambda_D} - 1} \right) P(0, S-K, 0, 0) \qquad (4.30)$$

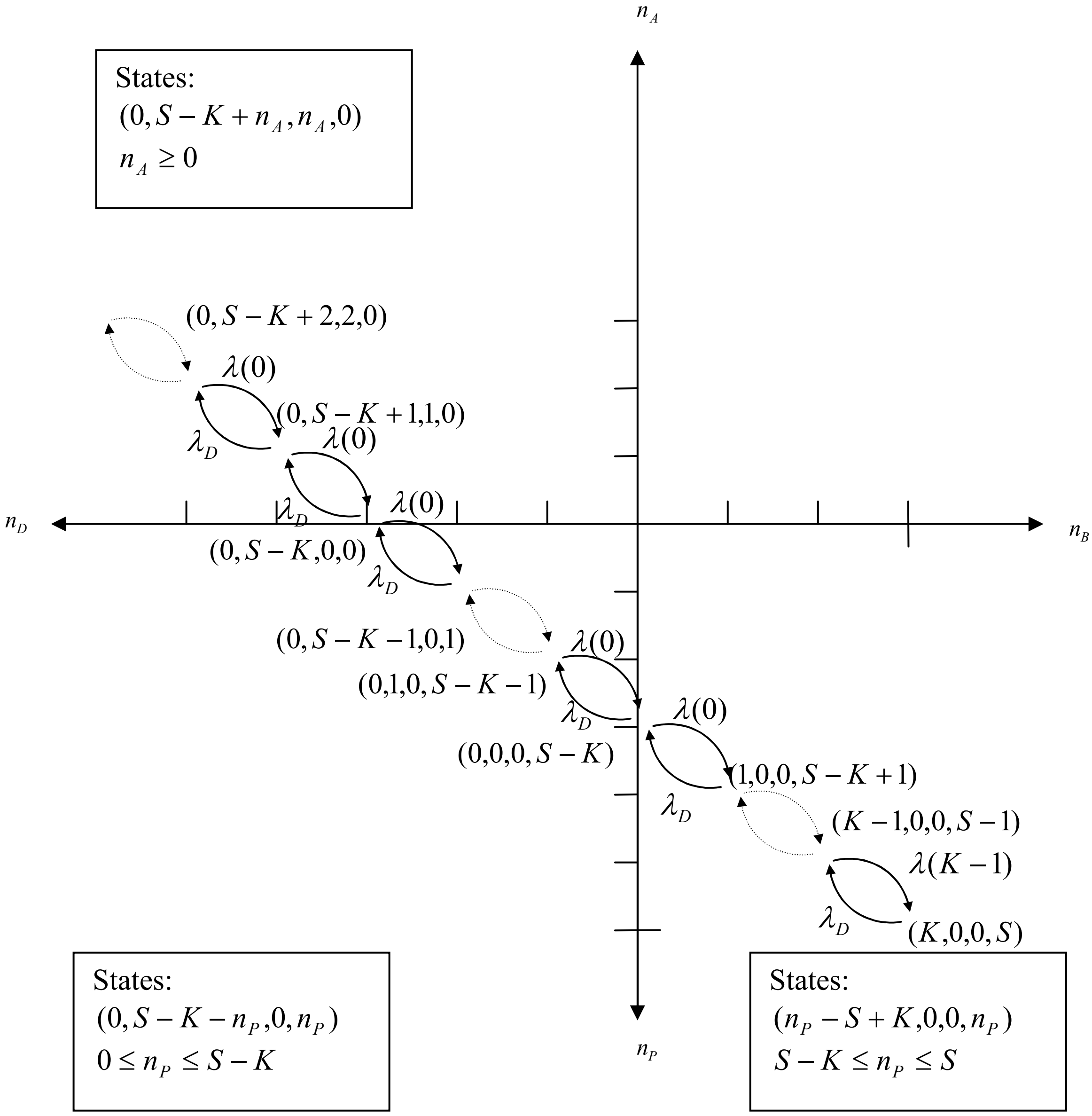

Figure 4.12. Markov chain associated with the last synchronization station in case $K < S$ (adapted from Duri *et al.*, 2000b)

The following probabilities must be calculated, too.

$$\widetilde{P}(n_p = n) = \left(\frac{\lambda(0)}{\lambda_D}\right)^n P(0, S-K, 0, 0) \quad \text{for } n = 1,\ldots, S-K \tag{4.31}$$

$$\widetilde{P}(n_p = n) = \frac{\prod_{i=0}^{n-S+K-1} \lambda(i)}{(\lambda_D)^{n-S+K}} \left(\frac{\lambda(0)}{\lambda_D}\right)^{S-K} P(0, S-K, 0, 0) \quad \text{for } n = S-K+1,\ldots, S \tag{4.32}$$

$$\widetilde{P}(n_P = 0) = 1 - \sum_{n=1}^{S} P(n_P = n) \tag{4.33}$$

4.2.3. Extended Kanban Control System (EKCS)

To be able to find an analytical solution to our problem we convert the EKCS, where we have dedicated kanbans and static routed demands, into a closed-loop network so that a product-form approximation technique can be applied to get the performance parameters. EKCS converted into a closed network is exactly the same with the original network (Figure 4.13). Here, the three synchronization stations on the right-hand-side of the figure are assumed to be one station and build the last synchronization station. The content of queue $S_R + S_M$ is finished products with dedicated kanbans attached to them. The content of queue F_R is free remanufacturing kanbans and queue F_M contains free manufacturing kanbans. The queues below these queues represent the demand queues. If we suppose the kanbans attached to the finished products to be detached and add them to the queues concerned, it turns out, that the maximum number in the queues of free kanbans increase from F_R and F_M to K_R and K_M, respectively. Figure 4.14 represents the last synchronization station in isolation with load-dependent arrival rates. n_{BR} is denoted as the current number in remanufacturing kanban queue, and n_{BM} is denoted as the current number in manufacturing kanban queue.

Let n_R be a variable that represents n_{BR} when it has positive value, and n_{DR} when it has negative value. Also, let n_M be a variable that represents n_{BM} when it has positive value, and n_{DM} when it has negative value. Moreover, let n_{FP} be a variable that represents n_P when it has positive value, and n_D when it has negative value. Then, if we follow the movements that occur in the queues, it can be easily seen that $n_{FP} = S - [(K_R - n_R) + (K_M - n_M)]$. $(K_R - n_R)$ represents the number of arrived remanufacturing demands, whereas $(K_M - n_M)$ represents the number of arrived manufacturing demands. The aim of finding out such an expression is to relate the current number in the finished product queue with the current number in the remanufacturing and manufacturing kanban queues. Thus, we avoid solving multi-dimensional Markov chain of the synchronization station in Figure 4.14 by solving small Markov chains. For example let K_R and K_M be equal to three, respectively. It turns out, that the above formula becomes $n_{FP} = (S-6) + n_R + n_M$. Hence, we can draw a picture of the situation, as depicted below (Figure 4.15).

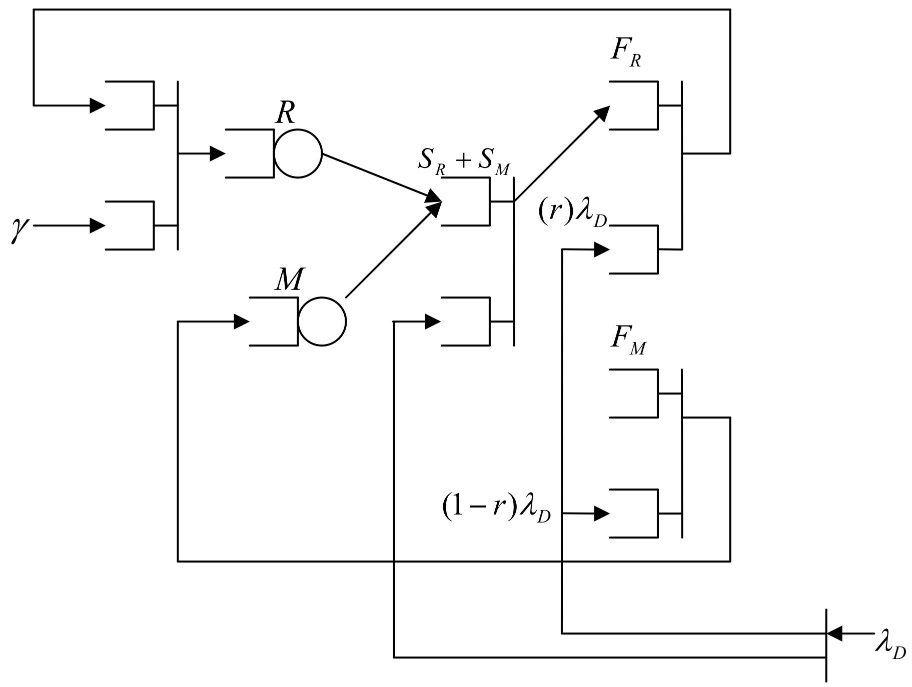

Figure 4.13. EKCS converted to a closed network

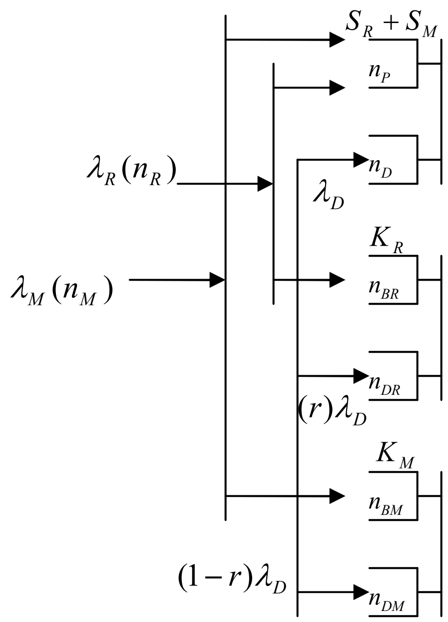

Figure 4.14. Last synchronization station in isolation

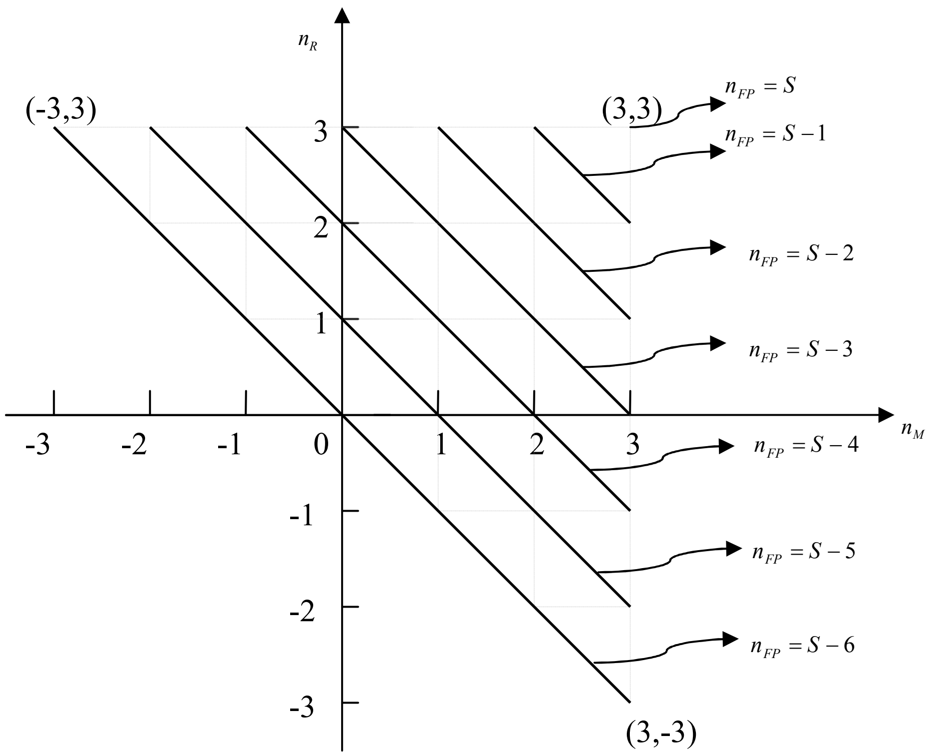

Figure 4.15. An example for $K_R = K_M = 3$

If we write down the probabilities that the current number in the finished product queue is equal to a value, we deduce:

$$P(n_{FP} = S) = P(n_R = K_R, n_M = K_M) \tag{4.34}$$

$$\begin{aligned} P(n_{FP} = S - 1) &= \sum_{j=0}^{1} P(n_R = K_R - 1 + j,\ n_M = K_M - j) = \\ &= P(n_R = K_R - 1, n_M = K_M) + P(n_R = K_R, n_M = K_M - 1) \end{aligned} \tag{4.35}$$

$$\begin{aligned} P(n_{FP} = S - 2) &= \sum_{j=0}^{2} P(n_R = K_R - 2 + j,\ n_M = K_M - j) = \\ &= P(n_R = K_R - 2, n_M = K_M) + P(n_R = K_R - 1, n_M = K_M - 1) + \\ &+ P(n_R = K_R, n_M = K_M - 2) \end{aligned} \tag{4.36}$$

$$\begin{aligned} P(n_{FP} = S - 3) &= \sum_{j=0}^{3} P(n_R = K_R - 3 + j,\ n_M = K_M - j) = \\ &= P(n_R = K_R - 3, n_M = K_M) + P(n_R = K_R - 2, n_M = K_M - 1) + \\ &+ P(n_R = K_R - 1, n_M = K_M - 2) + \\ &+ P(n_R = K_R, n_M = K_M - 3) \end{aligned} \tag{4.37}$$

If we generalize the above statements, it is obvious that:

$$P(n_{FP} = S - i) = \sum_{j=0}^{i} P(n_R = K_R - i + j,\ n_M = K_M - j) \tag{4.38}$$

Since we assume that n_R and n_M are independent, it turns out that:

$$P(n_{FP} = S - i) \approx \sum_{j=0}^{i} P(n_R = K_R - i + j)\, P(n_M = K_M - j) \tag{4.39}$$

These probabilities are utilized to calculate the average finished product queue and the backorder queue. The calculation of the probabilities needs solving Markov chains for states (n_{BR}, n_{DR}) and (n_{BM}, n_{DM}) where the first dimensions represent the current number in kanban queues for remanufacturing and manufacturing, respectively, and the second dimensions represent the current number of demand (backorder) queues for them. n_R represents the first dimension of the state space n_{BR} when it has positive value, and n_{DR} when it has negative value, as mentioned before. Below is the Markov chain of state (n_{BR}, n_{DR}) depicted (Figure 4.16). After that, the analysis is given by means of marginal probabilities.

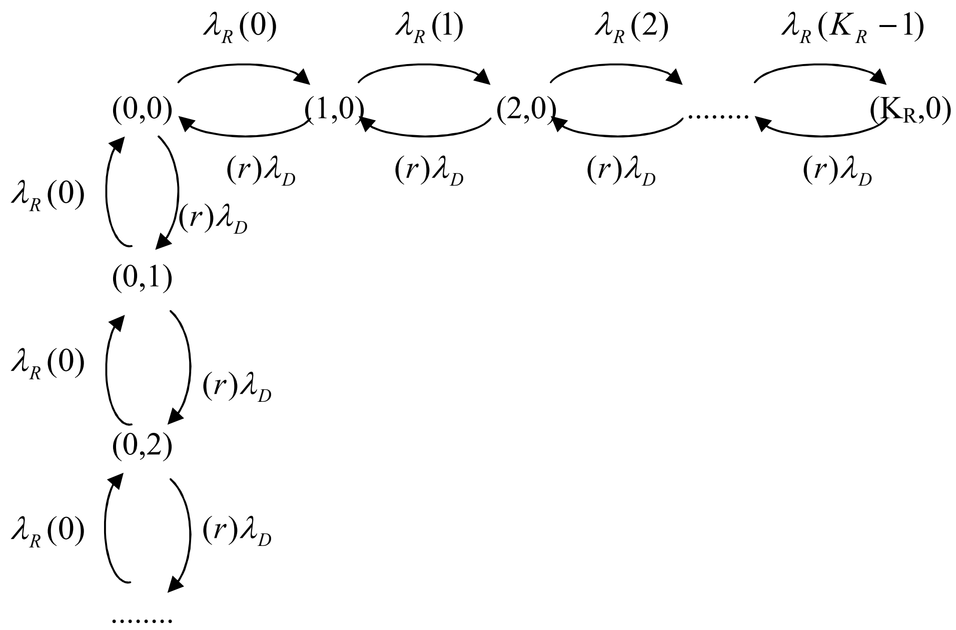

Figure 4.16. Markov chain representing the behavior of the state (n_{BR}, n_{DR})

$$P(n_R = i) = \frac{\prod_{n=0}^{i-1} \lambda_R(n)}{(r\lambda_D)^i} P(n_R = 0) \quad \text{for } i = 1,\dots,K_R \tag{4.40}$$

$$P(n_R = -i) = \left(\frac{r\lambda_D}{\lambda_R(0)} \right)^i P(n_R = 0) \quad \text{for } i = 1,\dots,\infty \tag{4.41}$$

$$P(n_R = 0) = \left(\frac{1}{1 - \dfrac{r\lambda_D}{\lambda_R(0)}} + \sum_{i=1}^{K_R} \frac{\prod_{n=0}^{i-1} \lambda_R(n)}{(r\lambda_D)^i} \right)^{-1} \tag{4.42}$$

$$\widetilde{P}(n_{BR} = i) = P(n_R = i) \qquad \text{for } i = 1,..., K_R \tag{4.43}$$

$$\widetilde{P}(n_{BR} = 0) = \sum_{i=0}^{\infty} P(n_R = -i) \tag{4.44}$$

State (n_{BM}, n_{DM}) displays exactly the same behavior with state (n_{BR}, n_{DR}). Thus, there is no need of constructing the underlying Markov chain since it is analyzed the same way. For the steady-state probabilities and marginal probabilities it is sufficient to replace index *R* with *M*, and probability *r* with (1-*r*).

Upon the convergence of the algorithm, the average queue lengths such as the average finished product queue and the average backorder queue can be easily obtained by means of expected values using the marginal probabilities calculated above.

4.2.4. Base Stock Control System (BSCS)

In (Dallery and Liberopoulos, 2000) it is pointed out that the EKCS with $K_i = \infty$, $S_i \geq 0, i = 1,..., N$, is equivalent to the BSCS having a base stock of S_i finished parts in stage *i*, according to property eight. The underlying proof can be found in this study.

In order to be able to solve our BSCS applied to a hybrid production environment analytically, we use this beneficial property. We make use of the same algorithm that we used for EKCS where we set the total kanban size equal to infinity.

In the BSCS, the incoming external demand generates directly a demand that is transmitted either to remanufacturing or manufacturing process. In the EKCS converted to a closed network, in order an external demand to generate a demand that is transmitted either to remanufacturing or manufacturing process, the associated kanban must be available. The indicator, that points us out the similarity between the EKCS and BSCS is the value of probabilities $P(n_{BR} = 0)$ and $P(n_{BM} = 0)$. We want the demand to immediately enter the system in the EKCS so that it can resemble BSCS. This is possible, in case that there are kanbans available in the system. Without any kanbans the demand is not allowed to be transmitted. Hence, the idea is that, the more kanbans the EKCS has, the closer it gets to BSCS.

4.3. Validation

In this section, we try to show how far our analytical results are from the simulation results, namely, we test the accuracy of the approximation techniques proposed in the previous section using related simulation models.

For each control policy we model the system as a stochastic simulation model using ARENA 9.0. We use the following replication parameters. We conduct our experiments over 30 replications whose length is chosen as 102,000 time units for each. The warm-up period is selected as 2,000 time units for each replication. The length of replications as time units is chosen considering convergence of average queues. Since any increase in that length does not have any additional effect on average queue lengths, we are not in doubt about the replication length and warm-up period. The "relative error", which is obtained as stated in formula 4.45, shows how close our approximations get to the simulation results.

$$\textit{Relative error} = (\textit{Approximation-Simulation})/\textit{Simulation} \qquad (4.45)$$

For each comparison set, both the remanufacturing service rate and the manufacturing service rate and the demand arrival rate are set equal to one. Using these parameter values, we conduct experiments where we change the values of static routing probabilities, return arrival rates, the buffer size of cores, and the related pull control parameter such as kanban sizes or base stock levels. We basically set the system parameters equal to some values and let one parameter change from one value to another and observe the differences between the analytical and simulation results. The comparison of each approximation considered is given in the following sections.

4.3.1. Kanban Control System (KCS)

We choose the buffer size of cores and the total kanban size as five, respectively. The service rates are set equal to one for both remanufacturing and manufacturing machines. The demand arrival rate is one while the return rate is 0.9. Using these parameter values and making changes only in routing probabilities, we obtain our approximation results by running the Mathematica code and the simulation results by running the ARENA model.

Table 4.2 represents the comparison of the approximation and simulation results as the static routing probability *r* changes.

We compare the average queue length of cores, *QR*, the average queue length of finished goods, *QFG*, the average queue length of backorders, *QD*, the average queue length of work-in-process for remanufacturing process, *WIPR,* and the average queue length of work-in-process for manufacturing process, *WIPM*. The expected throughput of remanufacturing process and the expected throughput of manufacturing process always converge to $(r)\lambda_D$ and $(1-r)\lambda_D$, respectively. The total throughput always converges to λ_D since the backorder queue has no limit and the total capacity of the hybrid system is adequate to supply the demand. Simulation results show that the analytical method calculates the throughputs with a relative error of zero. That is why we do not need to point out the expected throughputs in validation tables.

Table 4.2. Comparison of the approximation and simulation results as *r* changes

		App	Sim	RE
***r* = 0.6**	**QR**	3.2843	3.2912	-0.2%
	QFG	2.8511	2.8728	-0.8%
	QD	0.3350	0.4608	-27.3%
	WIPR	1.3315	1.3139	1.3%
	WIPM	0.6501	0.6448	0.8%
***r* = 0.5**	**QR**	3.8211	3.8198	0.0%
	QFG	3.0654	3.0767	-0.4%
	QD	0.1955	0.2368	-17.5%
	WIPR	0.9415	0.9319	1.0%
	WIPM	0.9415	0.9397	0.2%
***r* = 0.4**	**QR**	4.2146	4.2082	0.2%
	QFG	3.0069	3.0085	-0.1%
	QD	0.2388	0.3029	-21.2%
	WIPR	0.6500	0.6440	0.9%
	WIPM	1.3307	1.3345	-0.3%
***r* = 0.3**	**QR**	4.5021	4.4916	0.2%
	QFG	2.7075	2.6890	0.7%
	QD	0.5000	0.6789	-26.4%
	WIPR	0.4254	0.4205	1.2%
	WIPM	1.8652	1.8879	-1.2%
***r* = 0.2**	**QR**	4.7144	4.7072	0.2%
	QFG	2.1459	2.1262	0.9%
	QD	1.4058	1.7569	-20.0%
	WIPR	0.2497	0.2468	1.2%
	WIPM	2.6042	2.6267	-0.9%

In the next experiment set we set the static routing probabilities equal to 0.4, and change the return rate γ, while keeping the values of other parameters the same. The comparison is given in Table 4.3. The highest relative error occurs in estimating the average backorder queue whereas average return queue, finished products' queue and work-in-process queue are estimated with very small relative errors.

Table 4.3. Comparison of the approximation and simulation results as γ changes

		App	Sim	RE
γ = 0.6	**QR**	3.2843	3.2481	1.1%
	QFG	2.8511	2.8319	0.7%
	QD	0.3350	0.5314	-37.0%
	WIPR	0.6501	0.6306	3.1%
	WIPM	1.3315	1.3372	-0.4%
γ = 0.7	**QR**	3.7539	3.7372	0.4%
	QFG	2.9577	2.9575	0.0%
	QD	0.2646	0.3462	-23.6%
	WIPR	0.6498	0.6372	2.0%
	WIPM	1.3301	1.3357	-0.4%
γ = 0.8	**QR**	4.0332	4.0278	0.1%
	QFG	2.9933	2.9981	-0.2%
	QD	0.2453	0.3081	-20.4%
	WIPR	0.6499	0.6399	1.6%
	WIPM	1.3304	1.3349	-0.3%
γ = 0.9	**QR**	4.2146	4.2082	0.2%
	QFG	3.0069	3.0085	-0.1%
	QD	0.2388	0.3029	-21.2%
	WIPR	0.6500	0.6440	0.9%
	WIPM	1.3307	1.3345	-0.3%
γ = 1.0	**QR**	4.3404	4.3358	0.1%
	QFG	3.0127	3.0136	0.0%
	QD	0.2363	0.3018	-21.7%
	WIPR	0.6500	0.6425	1.2%
	WIPM	1.3310	1.3373	-0.5%

Table 4.4 represents the comparison of the approximation and simulation results as the kanban size K changes, where the return rate is set equal to 0.9 and the static routing probability is set equal to 0.4. The larger the kanban size is getting, the better average queues are estimated. However, the backorder queue has the largest relative error.

Table 4.4. Comparison of the approximation and simulation results as K changes

		App	Sim	RE
***K* = 5**	**QR**	4.2146	4.2082	0.2%
	QFG	3.0069	3.0085	-0.1%
	QD	0.2388	0.3029	-21.2%
	WIPR	0.6500	0.6440	0.9%
	WIPM	1.3307	1.3345	-0.3%
***K* = 10**	**QR**	4.2139	4.2118	0.0%
	QFG	7.8356	7.8489	-0.2%
	QD	0.0169	0.0210	-19.8%
	WIPR	0.6663	0.6603	0.9%
	WIPM	1.4843	1.4771	0.5%
***K* = 15**	**QR**	4.2139	4.2134	0.0%
	QFG	12.8207	12.8255	0.0%
	QD	0.0013	0.0019	-32.2%
	WIPR	0.6667	0.6628	0.6%
	WIPM	1.4987	1.4973	0.1%
***K* = 20**	**QR**	4.2139	4.2123	0.0%
	QFG	17.8196	17.8176	0.0%
	QD	0.0001	0.0001	-25.8%
	WIPR	0.6667	0.6630	0.6%
	WIPM	1.4999	1.5051	-0.3%

Table 4.5 reports the comparison of the approximation and simulation results as the buffer size of cores B changes. The return rate is set equal to 0.9 and the static routing probability is set equal to 0.4. The kanban size K is chosen as five. Change in B results in relative errors within 22 per cent for the average backorder queue, and within 5.5 per cent for the others.

The results of the experiments point out, that the analytical method we use can estimate the output values with a very small error in terms of simulation results. The largest errors occur in the estimation of the average backorder queue which is an expected result considering the relative errors reported by similar studies, e.g. (Baynat, 1995). The other queues show acceptably accurate results. Since the half widths of the simulation results always take very small values, we omit the confidence intervals as they do not contribute any additional insight.

Table 4.5. Comparison of the approximation and simulation results as *B* changes

		App	Sim	RE
***B* = 1**	**QR**	0.5556	0.5559	-0.1%
	QFG	2.7138	2.7389	-0.9%
	QD	0.3692	0.4338	-14.9%
	WIPR	0.6457	0.6130	5.3%
	WIPM	1.3118	1.3190	-0.5%
***B* = 3**	**QR**	2.2715	2.2662	0.2%
	QFG	2.9591	2.9696	-0.4%
	QD	0.2586	0.3205	-19.3%
	WIPR	0.6494	0.6310	2.9%
	WIPM	1.3282	1.3344	-0.5%
***B* = 5**	**QR**	4.2146	4.2082	0.2%
	QFG	3.0069	3.0085	-0.1%
	QD	0.2388	0.3029	-21.2%
	WIPR	0.6500	0.6440	0.9%
	WIPM	1.3307	1.3345	-0.3%
***B* = 10**	**QR**	9.2003	9.1971	0.0%
	QFG	3.0184	3.0209	-0.1%
	QD	0.2342	0.2957	-20.8%
	WIPR	0.6501	0.6439	1.0%
	WIPM	1.3313	1.3351	-0.3%

4.3.2. Generalized Kanban Control System (GKCS)

The validation of the approximation method for GKCS is realized for two cases. The first case includes experiment sets where $K \geq S$ and the second case includes experiment sets where $K < S$. Here, in addition to the previous performance measures, the average free kanban queue, *QB*, and the average backorder queue, *QA*, are reported. All the other queues are represented using the same abbreviations.

- *First case:* ($K \geq S$)

The buffer size of cores and the kanban size are set equal to five, respectively. The base stock level is chosen as four. The service rates for both processes are set equal to one. The static routing probability remains at value 0.2 whereas the return rates vary from 0.9 to 0.5. Here are the results of the experiments mentioned.

Table 4.6. Comparison of the approximation and simulation results as γ changes

		App	Sim	RE
γ = 0.9	**QR**	4.7144	4.7050	0.2%
	QB	2.1459	2.1353	0.5%
	QFG	1.4961	1.4908	0.4%
	QA	1.7559	2.0717	-15.2%
	WIPR	0.2497	0.2481	0.6%
	WIPM	2.6042	2.6162	-0.5%
γ = 0.8	**QR**	4.6670	4.6564	0.2%
	QB	2.1458	2.1334	0.6%
	QFG	1.4960	1.4900	0.4%
	QA	1.7560	2.1066	-16.6%
	WIPR	0.2497	0.2469	1.1%
	WIPM	2.6042	2.6190	-0.6%
γ = 0.7	**QR**	4.6008	4.5861	0.3%
	QB	2.1455	2.1378	0.4%
	QFG	1.4957	1.4930	0.2%
	QA	1.7563	2.0688	-15.1%
	WIPR	0.2497	0.2481	0.6%
	WIPM	2.6040	2.6129	-0.3%
γ = 0.6	**QR**	4.5021	4.4828	0.4%
	QB	2.1446	2.1384	0.3%
	QFG	1.4950	1.4933	0.1%
	QA	1.7575	2.0792	-15.5%
	WIPR	0.2497	0.2485	0.5%
	WIPM	2.6037	2.6101	-0.2%
γ = 0.5	**QR**	4.3403	4.3157	0.6%
	QB	2.1414	2.1377	0.2%
	QFG	1.4924	1.4926	0.0%
	QA	1.7622	2.0728	-15.0%
	WIPR	0.2497	0.2463	1.4%
	WIPM	2.6025	2.6070	-0.2%

Table 4.7 reports the results by holding the static routing probability r at value 0.4 and all the other parameter values remain the same as in the previous experiment set.

Table 4.7. Comparison of the approximation and simulation results as γ changes

		App	Sim	RE
γ = 0.6	**QR**	3.2843	3.2465	1.2%
	QB	2.8511	2.8346	0.6%
	QFG	2.0208	2.0126	0.4%
	QA	0.5047	0.7096	-28.9%
	WIPR	0.6501	0.6323	2.8%
	WIPM	1.3315	1.3338	-0.2%
γ = 0.7	**QR**	3.7539	3.7388	0.4%
	QB	2.9577	2.9536	0.1%
	QFG	2.1058	2.1054	0.0%
	QA	0.4127	0.5048	-18.2%
	WIPR	0.6498	0.6390	1.7%
	WIPM	1.3301	1.3380	-0.6%
γ = 0.8	**QR**	4.0332	4.0262	0.2%
	QB	2.9933	2.9896	0.1%
	QFG	2.1347	2.1340	0.0%
	QA	0.3868	0.4586	-15.7%
	WIPR	0.6499	0.6394	1.6%
	WIPM	1.3304	1.3427	-0.9%
γ = 0.9	**QR**	4.2146	4.2089	0.1%
	QB	3.0069	3.0080	0.0%
	QFG	2.1460	2.1491	-0.1%
	QA	0.3779	0.4417	-14.4%
	WIPR	0.6500	0.6419	1.3%
	WIPM	1.3307	1.3365	-0.4%
γ = 1.0	**QR**	4.3404	4.3365	0.1%
	QB	3.0127	3.0143	-0.1%
	QFG	2.1508	2.1543	-0.2%
	QA	0.3744	0.4417	-15.2%
	WIPR	0.6500	0.6415	1.3%
	WIPM	1.3310	1.3379	-0.5%

As the return rate increases, all relative error values decrease, and the largest errors occur in the estimation of average backorder queues (Table 4.7).

Table 4.8 demonstrates the cases where the buffer size of cores abbreviated as B and the kanban size K are set equal to five. The base stock level abbreviated as S is chosen as four. The remanufacturing and manufacturing service rates are one both. The external demand rate is one whereas the return rate is equal to 0.9. The static routing probabilities r vary from 0.6 to 0.2.

Table 4.8. Comparison of the approximation and simulation results as *r* changes

		App	Sim	RE
***r* = 0.6**	**QR**	3.2843	3.2873	-0.1%
	QB	2.8511	2.8666	-0.5%
	QFG	2.0208	2.0381	-0.8%
	QA	0.5047	0.6373	-20.8%
	WIPR	1.3315	1.3194	0.9%
	WIPM	0.6501	0.6423	1.2%
***r* = 0.5**	**QR**	3.8211	3.8202	0.0%
	QB	3.0654	3.0756	-0.3%
	QFG	2.1902	2.2005	-0.5%
	QA	0.3203	0.3632	-11.8%
	WIPR	0.9415	0.9315	1.1%
	WIPM	0.9415	0.9413	0.0%
***r* = 0.4**	**QR**	4.2146	4.2089	0.1%
	QB	3.0069	3.0080	0.0%
	QFG	2.1460	2.1491	-0.1%
	QA	0.3779	0.4417	-14.4%
	WIPR	0.6500	0.6419	1.3%
	WIPM	1.3307	1.3365	-0.4%
***r* = 0.3**	**QR**	4.5021	4.4894	0.3%
	QB	2.7075	2.6953	0.5%
	QFG	1.9158	1.9092	0.3%
	QA	0.7083	0.8772	-19.3%
	WIPR	0.4254	0.4214	0.9%
	WIPM	1.8652	1.8806	-0.8%
***r* = 0.2**	**QR**	4.7144	4.7050	0.2%
	QB	2.1459	2.1353	0.5%
	QFG	1.4961	1.4908	0.4%
	QA	1.7559	2.0717	-15.2%
	WIPR	0.2497	0.2481	0.6%
	WIPM	2.6042	2.6162	-0.5%

An increase in static routing probability results in smaller values of relative errors for each average queue length. *QR*, *QB*, *QFG*, *WIPR* and *WIPM* are estimated quite successfully. The backorder queue is estimated as the worst, but it still has acceptable error values (Table 4.8).

Table 4.9 reports the cases where B = 5, S = 4, the service rates are one for both processes and the demand rate is chosen as one. The return rate and the static routing probability are set equal to 0.9 and 0.6, respectively.

Table 4.9. Comparison of the approximation and simulation results as K changes

		App	Sim	RE
***K* = 5**	**QR**	3.2843	3.2873	-0.1%
	QB	2.8511	2.8666	-0.5%
	QFG	2.0208	2.0381	-0.8%
	QA	0.5047	0.6373	-20.8%
	WIPR	1.3315	1.3194	0.9%
	WIPM	0.6501	0.6423	1.2%
***K* = 10**	**QR**	3.2657	3.2543	0.3%
	QB	7.6054	7.6424	-0.5%
	QFG	2.0366	2.0633	-1.3%
	QA	0.4581	0.4714	-2.8%
	WIPR	1.4855	1.4413	3.1%
	WIPM	0.6664	0.6670	-0.1%
***K* = 15**	**QR**	3.2636	3.2549	0.3%
	QB	12.5743	12.6010	-0.2%
	QFG	2.0368	2.0600	-1.1%
	QA	0.4678	0.4675	0.1%
	WIPR	1.4990	1.4602	2.7%
	WIPM	0.6667	0.6652	0.2%
***K* = 20**	**QR**	3.2634	3.2547	0.3%
	QB	17.5705	17.6116	-0.2%
	QFG	2.0368	2.0647	-1.4%
	QA	0.4784	0.4536	5.5%
	WIPR	1.4999	1.4572	2.9%
	WIPM	0.6667	0.6637	0.4%

The kanban sizes vary from five to twenty. As we can see from the Table 4.9, the more the kanban size is getting, the smaller are the relative errors related to the backorder queue lengths.

Table 4.10 exhibits the effects of changes in base stock level S while all other parameter values remain as follows: The cores' buffer has a size of five whereas a total number of seven kanbans are available. The service rates and the demand rate are set equal to one. The return rate and the routing probability are 0.9 and 0.6, respectively.

Table 4.10. Comparison of the approximation and simulation results as *S* changes

		App	Sim	RE
***S* = 1**	**QR**	3.2723	3.2739	-0.1%
	QB	4.6934	4.7302	-0.8%
	QFG	0.2186	0.2251	-2.9%
	QA	1.4508	1.6823	-13.8%
	WIPR	1.4344	1.4010	2.4%
	WIPM	0.6632	0.6571	0.9%
***S* = 2**	**QR**	3.2723	3.2739	-0.1%
	QB	4.6934	4.7302	-0.8%
	QFG	0.6629	0.6787	-2.3%
	QA	1.0317	1.1359	-9.2%
	WIPR	1.4344	1.4010	2.4%
	WIPM	0.6632	0.6571	0.9%
***S* = 3**	**QR**	3.2723	3.2739	-0.1%
	QB	4.6934	4.7302	-0.8%
	QFG	1.2853	1.3102	-1.9%
	QA	0.7013	0.7674	-8.6%
	WIPR	1.4344	1.4010	2.4%
	WIPM	0.6632	0.6571	0.9%
***S* = 4**	**QR**	3.2723	3.2739	-0.1%
	QB	4.6934	4.7302	-0.8%
	QFG	2.0345	2.0664	-1.5%
	QA	0.4655	0.5236	-11.1%
	WIPR	1.4344	1.4010	2.4%
	WIPM	0.6632	0.6571	0.9%
***S* = 5**	**QR**	3.2723	3.2739	-0.1%
	QB	4.6934	4.7302	-0.8%
	QFG	2.8701	2.9062	-1.2%
	QA	0.3052	0.3634	-16.0%
	WIPR	1.4344	1.4010	2.4%
	WIPM	0.6632	0.6571	0.9%
***S* = 6**	**QR**	3.2723	3.2739	-0.1%
	QB	4.6934	4.7302	-0.8%
	QFG	3.7630	3.8007	-1.0%
	QA	0.1989	0.2579	-22.9%
	WIPR	1.4344	1.4010	2.4%
	WIPM	0.6632	0.6571	0.9%

An increase in *S* results in an increase in *QFG* and a decline in *QA*, and no change in other queues, obviously. The relative errors have a decreasing trend.

We also conduct experiments where the buffer size of cores varies while all other parameter values remain the same (Table 4.11). The kanban size and the base stock level are chosen as five and four, respectively. The service rates and the demand rate are set equal to one. The return rate is 0.9 while the static routing probability is 0.6.

Table 4.11. Comparison of the approximation and simulation results as *B* changes

		App	Sim	RE
***B* = 5**	**QR**	3.2843	3.2873	-0.1%
	QB	2.8511	2.8666	-0.5%
	QFG	2.0208	2.0381	-0.8%
	QA	0.5047	0.6373	-20.8%
	WIPR	1.3315	1.3194	0.9%
	WIPM	0.6501	0.6423	1.2%
***B* = 10**	**QR**	8.0453	8.0813	-0.4%
	QB	2.9976	3.0011	-0.1%
	QFG	2.1389	2.1436	-0.2%
	QA	0.3869	0.4564	-15.2%
	WIPR	1.3315	1.3357	-0.3%
	WIPM	0.6502	0.6433	1.1%
***B* = 15**	**QR**	13.0072	13.0555	-0.4%
	QB	3.0158	3.0140	0.1%
	QFG	2.1536	2.1541	0.0%
	QA	0.3734	0.4400	-15.1%
	WIPR	1.3314	1.3404	-0.7%
	WIPM	0.6501	0.6429	1.1%

The relative errors associated with all the average queue lengths except the backorder queue are about one per cent (Table 4.11).

- Second case: ($K < S$)

The buffer size of cores and the base stock level are set equal to five, respectively. The kanban size is chosen as four. The service rates for both processes are set equal to one. The static routing probability remains at value 0.4 whereas the return rates vary from 0.6 to 1.0. The results of the experiments mentioned are given in Table 4.12.

Table 4.12. Comparison of the approximation and simulation results as γ changes

		App	Sim	RE
γ = 0.6	**QR**	3.2950	3.2573	1.2%
	QB	1.9922	1.9823	0.5%
	QFG	2.8111	2.7850	0.9%
	QA	0.3866	0.6599	-41.4%
	WIPR	0.6323	0.6166	2.5%
	WIPM	1.2362	1.2327	0.3%
γ = 0.7	**QR**	3.7579	3.7421	0.4%
	QB	2.0783	2.0729	0.3%
	QFG	2.9192	2.9030	0.6%
	QA	0.3096	0.4451	-30.5%
	WIPR	0.6321	0.6212	1.8%
	WIPM	1.2355	1.2437	-0.7%
γ = 0.8	**QR**	4.0348	4.0231	0.3%
	QB	2.1083	2.1054	0.1%
	QFG	2.9563	2.9454	0.4%
	QA	0.2870	0.3851	-25.5%
	WIPR	0.6322	0.6250	1.1%
	WIPM	1.2358	1.2435	-0.6%
γ = 0.9	**QR**	4.2152	4.2072	0.2%
	QB	2.1202	2.1183	0.1%
	QFG	2.9708	2.9613	0.3%
	QA	0.2789	0.3733	-25.3%
	WIPR	0.6323	0.6268	0.9%
	WIPM	1.2361	1.2430	-0.6%
γ = 1.0	**QR**	4.3407	4.3389	0.0%
	QB	2.1254	2.1268	-0.1%
	QFG	2.9771	2.9718	0.2%
	QA	0.2756	0.3676	-25.0%
	WIPR	0.6323	0.6246	1.2%
	WIPM	1.2363	1.2429	-0.5%

As shown in Table 4.12, the higher the return rate is getting, the smaller are the relative errors associated with backorder queue. The average queue lengths except the backorder queue are estimated very successfully.

The next experiment set (Table 4.13) includes the same parameter values as the previous one except the return rate and the static routing probability. The return rate is set equal to 0.9. Meanwhile, the static routing probabilities vary from 0.6 to 0.2.

Table 4.13. Comparison of the approximation and simulation results as *r* changes

		App	Sim	RE
***r* = 0.6**	**QR**	3.2950	3.3047	-0.3%
	QB	1.9922	2.0051	-0.6%
	QFG	2.8111	2.8143	-0.1%
	QA	0.3866	0.5642	-31.5%
	WIPR	1.2362	1.2275	0.7%
	WIPM	0.6323	0.6251	1.1%
***r* = 0.5**	**QR**	3.8244	3.8218	0.1%
	QB	2.1599	2.1644	-0.2%
	QFG	3.0230	3.0213	0.1%
	QA	0.2883	0.3154	-8.6%
	WIPR	0.8974	0.8913	0.7%
	WIPM	0.8974	0.8976	0.0%
***r* = 0.4**	**QR**	4.2152	4.2072	0.2%
	QB	2.1202	2.1183	0.1%
	QFG	2.9708	2.9613	0.3%
	QA	0.2789	0.3733	-25.3%
	WIPR	0.6323	0.6268	0.9%
	WIPM	1.2361	1.2430	-0.6%
***r* = 0.3**	**QR**	4.5022	4.4894	0.3%
	QB	1.8997	1.8879	0.6%
	QFG	2.6849	2.6604	0.9%
	QA	0.5368	0.7377	-27.2%
	WIPR	0.4200	0.4161	0.9%
	WIPM	1.6784	1.6933	-0.9%
***r* = 0.2**	**QR**	4.7144	4.7031	0.2%
	QB	1.4902	1.4811	0.6%
	QFG	2.1376	2.1158	1.0%
	QA	1.4318	1.8039	-20.6%
	WIPR	0.2488	0.2468	0.8%
	WIPM	2.2608	2.2716	-0.5%

An increase in static routing probability results in smaller values of relative errors for each average queue length. The worst estimation occurs in the average backorder queues.

Another experiment set (Table 4.14) includes the buffer size of cores as five and the base stock level as seven. The service rates of either process and the external demand rate are set equal to one. The return rate and the static routing probability are chosen as 0.9 and 0.6, respectively. Here we change with kanban sizes.

Table 4.14. Comparison of the approximation and simulation results as *K* changes

		App	Sim	RE
***K* = 2**	**QR**	3.3332	3.3320	0.0%
	QB	0.5103	0.5085	0.4%
	QFG	3.7927	3.6537	3.8%
	QA	0.7690	1.1396	-32.5%
	WIPR	0.8893	0.8885	0.1%
	WIPM	0.5286	0.5272	0.3%
***K* = 3**	**QR**	3.3106	3.3192	-0.3%
	QB	1.2015	1.2068	-0.4%
	QFG	4.4276	4.3466	1.9%
	QA	0.2733	0.4896	-44.2%
	WIPR	1.0946	1.0901	0.4%
	WIPM	0.5969	0.5927	0.7%
***K* = 4**	**QR**	3.2950	3.3047	-0.3%
	QB	1.9922	2.0051	-0.6%
	QFG	4.6038	4.5721	0.7%
	QA	0.1793	0.3220	-44.3%
	WIPR	1.2362	1.2275	0.7%
	WIPM	0.6323	0.6251	1.1%
***K* = 5**	**QR**	3.2843	3.2873	-0.1%
	QB	2.8511	2.8666	-0.5%
	QFG	4.6637	4.6595	0.1%
	QA	0.1476	0.2587	-42.9%
	WIPR	1.3315	1.3194	0.9%
	WIPM	0.6501	0.6423	1.2%
***K* = 6**	**QR**	3.2771	3.2868	-0.3%
	QB	3.7565	3.7874	-0.8%
	QFG	4.6855	4.7124	-0.6%
	QA	0.1347	0.2073	-35.0%
	WIPR	1.3941	1.3738	1.5%
	WIPM	0.6589	0.6520	1.1%

The larger the kanban size is getting, the better average queues are estimated. However, the backorder queue has the largest relative error.

In the last experiment set (Table 4.15) we change the buffer size of cores where we set the kanban size and the base stock level equal to four and five, respectively. The service rates and the demand rate are chosen as one. The return rate is 0.9 while the static routing probability is 0.6. The average queue lengths except the backorder queue is estimated with small relative error values.

Table 4.15. Comparison of the approximation and simulation results as *B* changes

		App	Sim	RE
***B* = 5**	**QR**	3.2950	3.3047	-0.3%
	QB	1.9922	2.0051	-0.6%
	QFG	2.8111	2.8143	-0.1%
	QA	0.3866	0.5642	-31.5%
	WIPR	1.2362	1.2275	0.7%
	WIPM	0.6323	0.6251	1.1%
***B* = 10**	**QR**	8.0501	8.0967	-0.6%
	QB	2.1138	2.1153	-0.1%
	QFG	2.9625	2.9560	0.2%
	QA	0.2852	0.3939	-27.6%
	WIPR	1.2368	1.2422	-0.4%
	WIPM	0.6325	0.6263	1.0%
***B* = 15**	**QR**	13.0084	13.0807	-0.6%
	QB	2.1286	2.1292	0.0%
	QFG	2.9808	2.9739	0.2%
	QA	0.2742	0.3701	-25.9%
	WIPR	1.2367	1.2414	-0.4%
	WIPM	0.6324	0.6276	0.8%

The approximation method that we used for solving GKCS analytically, seems to be very accurate if we investigate all the above tables. The relative errors are small. The largest errors occur in the estimation of the average backorder queues. The other average queues are always estimated with a relative error of smaller than four per cent.

4.3.3. Extended Kanban Control System (EKCS)

This section includes observations on the comparison of analytical and simulation results for extended kanban control system. The average queue length of backorders is represented as *QBL*, while the remaining abbreviations for average queue lengths are the same as that of KCS.

Firstly, we investigate cases, where the parameters excluding the return rate remain the same. Here, the buffer size of cores is five, the remanufacturing and manufacturing kanban sizes are set equal to three, respectively, the base stock level is chosen as two, and both service rates for remanufacturing and manufacturing processes and the demand rate is one while the static routing probability is chosen as 0.4. The results are given in Table 4.16.

Table 4.16. Comparison of the approximation and simulation results as γ changes

		App	Sim	RE
γ = 0.5	**QR**	2.4259	2.3080	5.1%
	QFG	0.5781	0.5455	6.0%
	QBL	1.3777	2.0666	-33.3%
	WIPR	0.6417	0.6143	4.5%
	WIPM	1.1760	1.1764	0.0%
γ = 0.6	**QR**	3.3002	3.2702	0.9%
	QFG	0.6712	0.6682	0.4%
	QBL	1.0134	1.0928	-7.3%
	WIPR	0.6294	0.6166	2.1%
	WIPM	1.1760	1.1758	0.0%
γ = 0.7	**QR**	3.7601	3.7605	0.0%
	QFG	0.7001	0.7010	-0.1%
	QBL	0.9322	0.9443	-1.3%
	WIPR	0.6256	0.6210	0.7%
	WIPM	1.1760	1.1766	-0.1%
γ = 0.8	**QR**	4.0356	4.0441	-0.2%
	QFG	0.7110	0.7123	-0.2%
	QBL	0.9058	0.9038	0.2%
	WIPR	0.6245	0.6210	0.6%
	WIPM	1.1760	1.1766	-0.1%
γ = 0.9	**QR**	4.2156	4.2237	-0.2%
	QFG	0.7155	0.7171	-0.2%
	QBL	0.8956	0.8930	0.3%
	WIPR	0.6241	0.6226	0.2%
	WIPM	1.1760	1.1753	0.1%
γ = 1.0	**QR**	4.3409	4.3517	-0.2%
	QFG	0.7176	0.7184	-0.1%
	QBL	0.8912	0.8877	0.4%
	WIPR	0.6240	0.6230	0.2%
	WIPM	1.1760	1.1761	0.0%

As the return rate increases, the relative errors decrease, and the average backorder queue length is estimated with very small errors, especially for large values of γ (Table 4.16).

In the second experiment set conducted (Table 4.17), the buffer size for cores, both kanban sizes, the base stock level, the service rates and the demand rate remain the same as in the previous case. The return rate is set equal to 0.9, and we investigate the effect of changes in static routing probabilities on the system.

Table 4.17. Comparison of the approximation and simulation results as *r* changes

		App	Sim	RE
***r* = 0.7**	**QR**	2.6035	2.6075	-0.2%
	QFG	0.4886	0.4998	-2.2%
	QBL	2.3224	2.5640	-9.4%
	WIPR	1.5337	1.5036	2.0%
	WIPM	0.4170	0.4174	-0.1%
***r* = 0.6**	**QR**	3.3018	3.3224	-0.6%
	QFG	0.6633	0.6720	-1.3%
	QBL	1.0811	1.1084	-2.5%
	WIPR	1.1774	1.1627	1.3%
	WIPM	0.6240	0.6227	0.2%
***r* = 0.5**	**QR**	3.8262	3.8384	-0.3%
	QFG	0.7315	0.7345	-0.4%
	QBL	0.7940	0.7977	-0.5%
	WIPR	0.8755	0.8701	0.6%
	WIPM	0.8750	0.8773	-0.3%
***r* = 0.4**	**QR**	4.2156	4.2237	-0.2%
	QFG	0.7155	0.7171	-0.2%
	QBL	0.8956	0.8930	0.3%
	WIPR	0.6241	0.6226	0.2%
	WIPM	1.1760	1.1753	0.1%
***r* = 0.3**	**QR**	4.5022	4.5048	-0.1%
	QFG	0.6293	0.6282	0.2%
	QBL	1.3933	1.4047	-0.8%
	WIPR	0.4170	0.4169	0.0%
	WIPM	1.5330	1.5361	-0.2%
***r* = 0.2**	**QR**	4.7145	4.7148	0.0%
	QFG	0.4800	0.4770	0.6%
	QBL	2.7301	2.7619	-1.2%
	WIPR	0.2480	0.2486	-0.2%
	WIPM	1.9520	1.9576	-0.3%

For all static routing probability values, the analytic method provides excellent results, even for the estimation of backorder queue length (Table 4.17).

Table 4.18 demonstrates the experiment results of when we hold the buffer size of cores, the base stock level, the service rates, the demand rate at five, two, one for both and one, respectively. The return rate is set equal to 0.9 whereas the static routing probability is chosen as 0.6.

Table 4.18. Comparison of the approximation and simulation results as kanban sizes change

		App	Sim	RE
$K_R = 1$	QR	3.3655	3.3877	-0.7%
$K_M = 1$	QFG	0.6621	0.4787	38.3%
	QBL	1.0806	1.3959	-22.6%
	WIPR	0.6000	0.6006	-0.1%
	WIPM	0.4000	0.3993	0.2%
$K_R = 2$	QR	3.3255	3.3433	-0.5%
$K_M = 2$	QFG	0.6629	0.6623	0.1%
	QBL	1.0794	1.1590	-6.9%
	WIPR	0.9606	0.9569	0.4%
	WIPM	0.5600	0.5600	0.0%
$K_R = 3$	QR	3.3018	3.3224	-0.6%
$K_M = 3$	QFG	0.6633	0.6720	-1.3%
	QBL	1.0811	1.1084	-2.5%
	WIPR	1.1774	1.1627	1.3%
	WIPM	0.6240	0.6227	0.2%
$K_R = 4$	QR	3.2873	3.3021	-0.4%
$K_M = 4$	QFG	0.6635	0.6763	-1.9%
	QBL	1.0836	1.0851	-0.1%
	WIPR	1.3075	1.2830	1.9%
	WIPM	0.6496	0.6490	0.1%
$K_R = 5$	QR	3.2783	3.2830	-0.1%
$K_M = 5$	QFG	0.6637	0.6772	-2.0%
	QBL	1.0860	1.0726	1.2%
	WIPR	1.3855	1.3571	2.1%
	WIPM	0.6598	0.6577	0.3%

The higher the kanban sizes are getting, the smaller are the relative errors. All of the queue lengths are estimated successfully, including the backorder queue.

Table 4.19 reports the cases where we have the core buffer size as five and a total number of kanbans of ten, five of them dedicated for remanufacturing process and the remaining five of them dedicated for manufacturing process. The service rates and the return rate are set equal to one. The return rate is chosen as 0.9 whereas the static routing probability is 0.6.

Table 4.19. Comparison of the approximation and simulation results as *S* changes

		App	Sim	RE
***S* = 2**	**QR**	3.2783	3.2830	-0.1%
	QFG	0.6637	0.6772	-2.0%
	QBL	1.0860	1.0726	1.2%
	WIPR	1.3855	1.3571	2.1%
	WIPM	0.6598	0.6577	0.3%
***S* = 3**	**QR**	3.2783	3.2830	-0.1%
	QFG	1.2867	1.3076	-1.6%
	QBL	0.7090	0.7030	0.9%
	WIPR	1.3855	1.3571	2.1%
	WIPM	0.6598	0.6577	0.3%
***S* = 4**	**QR**	3.2783	3.2830	-0.1%
	QFG	2.0370	2.0631	-1.3%
	QBL	0.4593	0.4585	0.2%
	WIPR	1.3855	1.3571	2.1%
	WIPM	0.6598	0.6577	0.3%
***S* = 5**	**QR**	3.2783	3.2830	-0.1%
	QFG	2.8738	2.9030	-1.0%
	QBL	0.2961	0.2984	-0.8%
	WIPR	1.3855	1.3571	2.1%
	WIPM	0.6598	0.6577	0.3%
***S* = 6**	**QR**	3.2783	3.2830	-0.1%
	QFG	3.7680	3.7949	-0.7%
	QBL	0.1903	0.1958	-2.8%
	WIPR	1.3855	1.3571	2.1%
	WIPM	0.6598	0.6577	0.3%

An increase in base stock level causes a decline in relative errors. All of the queue lengths are estimated with relative errors of three per cent at most.

The last experiment set of EKCS includes three kanbans dedicated for remanufacturing and manufacturing process, respectively. The base stock level is chosen as two. The service rates and the demand rate are one. The return rate is set equal to 0.9 whereas the static routing probability is 0.4. The buffer sizes of cores are changing. The results are reported in the following table (Table 4.20).

Table 4.20. Comparison of the approximation and simulation results as *B* changes

		App	Sim	RE
B **= 5**	**QR**	4.2156	4.2237	-0.2%
	QFG	0.7155	0.7171	-0.2%
	QBL	0.8956	0.8930	0.3%
	WIPR	0.6241	0.6226	0.2%
	WIPM	1.1760	1.1753	0.1%
B **= 10**	**QR**	9.2003	9.2136	-0.1%
	QFG	0.7199	0.7185	0.2%
	QBL	0.8868	0.8874	-0.1%
	WIPR	0.6240	0.6267	-0.4%
	WIPM	1.1760	1.1761	0.0%
B **= 15**	**QR**	14.2000	14.2124	-0.1%
	QFG	0.7200	0.7176	0.3%
	QBL	0.8867	0.8892	-0.3%
	WIPR	0.6240	0.6264	-0.4%
	WIPM	1.1760	1.1782	-0.2%

As it is obvious from Table 4.20, the relative errors related to changes in the core buffer size is not more than 0.4 per cent.

The relative errors point out that the approximation method used for solving EKCS analytically seems to be accurate. The errors are small. In most cases, even the average backorder queue lengths are estimated with a relative error of smaller than ten per cent.

4.3.4. Base Stock Control System (BSCS)

As mentioned previously, the approximation method used for BSCS, is based upon the analytical method utilized for EKCS. The average queue lengths of BSCS are abbreviated as the same as those of EKCS.

Firstly, we change the return rates. The buffer size of cores and the base stock level are set equal to five and two, respectively. The demand rate and the service rates are chosen as one. The static routing probability is 0.4.

Table 4.21. Comparison of the approximation and simulation results as γ changes

		App	Sim	RE
γ = 1.0	**QR**	4.3402	4.3401	0.0%
	QFG	0.7176	0.7186	-0.1%
	QBL	0.8911	0.8895	0.2%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
γ = 0.9	**QR**	4.2139	4.2147	0.0%
	QFG	0.7155	0.7163	-0.1%
	QBL	0.8960	0.8990	-0.3%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
γ = 0.8	**QR**	4.0313	4.0307	0.0%
	QFG	0.7106	0.7127	-0.3%
	QBL	0.9085	0.9066	0.2%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
γ = 0.7	**QR**	3.7479	3.7459	0.1%
	QFG	0.6985	0.7017	-0.5%
	QBL	0.9464	0.9415	0.5%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
γ = 0.6	**QR**	3.2634	3.2604	0.1%
	QFG	0.6638	0.6716	-1.2%
	QBL	1.0939	1.0817	1.1%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
γ = 0.5	**QR**	2.3107	2.3025	0.4%
	QFG	0.5438	0.5497	-1.1%
	QBL	2.0212	1.9985	1.1%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%

As can be seen from Table 4.21, the decline in the return rate does not affect the relative errors very significantly. They are not more than 1.2 per cent.

After that, we change the static routing probabilities. The return rate is set equal to 0.9, and all the other parameter values are same as the previous case. The table below is a demonstration of this situation (Table 4.22).

Table 4.22. Comparison of the approximation and simulation results as *r* changes

		App	Sim	RE
***r* = 0.7**	**QR**	2.4963	2.4803	0.6%
	QFG	0.5009	0.5224	-4.1%
	QBL	2.2589	2.2220	1.7%
	WIPR	2.3331	2.2356	4.4%
	WIPM	0.4286	0.4296	-0.2%
***r* = 0.6**	**QR**	3.2634	3.2571	0.2%
	QFG	0.6638	0.6773	-2.0%
	QBL	1.0939	1.0696	2.3%
	WIPR	1.5000	1.4555	3.1%
	WIPM	0.6667	0.6655	0.2%
***r* = 0.5**	**QR**	3.8162	3.8095	0.2%
	QFG	0.7312	0.7367	-0.7%
	QBL	0.7974	0.7887	1.1%
	WIPR	1.0000	0.9834	1.7%
	WIPM	1.0000	1.0014	-0.1%
***r* = 0.4**	**QR**	4.2139	4.2147	0.0%
	QFG	0.7155	0.7163	-0.1%
	QBL	0.8960	0.8990	-0.3%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
***r* = 0.3**	**QR**	4.5021	4.5002	0.0%
	QFG	0.6293	0.6275	0.3%
	QBL	1.3933	1.4048	-0.8%
	WIPR	0.4286	0.4291	-0.1%
	WIPM	2.3330	2.3462	-0.6%
***r* = 0.2**	**QR**	4.7144	4.7128	0.0%
	QFG	0.4800	0.4781	0.4%
	QBL	2.7301	2.7298	0.0%
	WIPR	0.2500	0.2505	-0.2%
	WIPM	3.9849	4.0012	-0.4%

The farther the static routing probability is from the return rate, the smaller are the relative errors. The analytical method reflects accurate results considering the small values of relative errors (Table 4.22).

In the next experiment set the buffer size of cores is set to five (Table 4.23). The return rate is 0.9 whereas the static routing probability is 0.4. The service rates and the demand rate are set equal to one. The base stock levels differ from one to fifteen.

Table 4.23. Comparison of the approximation and simulation results as *S* changes

		App	Sim	RE
***S* = 1**	**QR**	4.2139	4.2147	0.0%
	QFG	0.2382	0.2385	-0.1%
	QBL	1.4187	1.4212	-0.2%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
***S* = 2**	**QR**	4.2139	4.2147	0.0%
	QFG	0.7155	0.7163	-0.1%
	QBL	0.8960	0.8990	-0.3%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
***S* = 5**	**QR**	4.2139	4.2147	0.0%
	QFG	3.0270	3.0275	0.0%
	QBL	0.2076	0.2102	-1.3%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
***S* = 10**	**QR**	4.2139	4.2147	0.0%
	QFG	7.8360	7.8346	0.0%
	QBL	0.0166	0.0173	-4.0%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
***S* = 15**	**QR**	4.2139	4.2147	0.0%
	QFG	12.8208	12.8184	0.0%
	QBL	0.0013	0.0011	20.9%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%

As we can observe from Table 4.23, an increase in base stock level does not cause any effect on relative errors.

Finally, we conduct experiments where the return rate and the static routing probability are 0.9 and 0.4, respectively (Table 4.24). The demand rate and the service rates are set equal to one. The base stock level is chosen as two, and we look at what is going on if we make changes in the buffer size of cores.

Table 4.24. Comparison of the approximation and simulation results as *B* changes

		App	Sim	RE
***B* = 1**	**QR**	0.5556	0.5545	0.2%
	QFG	0.6041	0.6174	-2.2%
	QBL	1.1263	1.1062	1.8%
	WIPR	0.6667	0.6327	5.4%
	WIPM	1.5000	1.4983	0.1%
***B* = 3**	**QR**	2.2702	2.2657	0.2%
	QFG	0.6971	0.7025	-0.8%
	QBL	0.9340	0.9251	1.0%
	WIPR	0.6667	0.6514	2.3%
	WIPM	1.5000	1.4995	0.0%
***B* = 5**	**QR**	4.2139	4.2147	0.0%
	QFG	0.7155	0.7163	-0.1%
	QBL	0.8960	0.8990	-0.3%
	WIPR	0.6667	0.6635	0.5%
	WIPM	1.5000	1.5056	-0.4%
***B* = 10**	**QR**	9.2002	9.1955	0.1%
	QFG	0.7199	0.7195	0.1%
	QBL	0.8868	0.8858	0.1%
	WIPR	0.6667	0.6687	-0.3%
	WIPM	1.5000	1.4974	0.2%

Any change in the core buffer size does not affect the relative errors significantly. Even for the smallest value of the core buffer size the error is not more than 5.4 per cent (Table 4.24).

As it is obvious, the average queue lengths are obtained quite accurately in terms of relative errors that occur. The method proposed to solve the BSCS analytically gives acceptable results.

4.4. Cost Optimization

After obtaining satisfactory results from the accuracy tests of the analytical methods, we compare our four control policies on a hybrid production system. The comparison is made in terms of minimum costs achieved for each control policy under a given parameter set. The total cost function consists of four components, viz. holding cost, backorder cost, production cost and disposal cost.

Total Cost = Holding Cost + Backorder Cost + Production Cost + Disposal Cost (4.46)

The cost parameters are calculated according to Teunter *et al.*, (2000). Here, the fifth method for setting the holding cost rates in an average cost model outperforms other methods. The method is based on the idea that the alternative for stocking a non-serviceable item is remanufacturing rather than disposing. In their study, this method led to near optimal strategies for all examples considered.

Our models consider three types of stocked items, namely, non-serviceable items- these are cores that are not yet remanufactured-, remanufactured items and manufactured items. Thus, holding cost is easily calculable using average queue lengths and holding cost rates. The alternative for stocking a non-serviceable item is remanufacturing it. Remanufacturing the item means postponing the manufacturing of another item, thus, it turns out, that there are savings of $c_M - c_R$. The holding cost rate for non-serviceable items becomes $h + \alpha(c_M - c_R)$, i.e. the sum of the out-of-pocket holding cost rate and opportunity cost rate. The method considers the cost for collecting cores as fixed. Therefore, this cost is not included in the opportunity cost rate. Moreover, the cost associated with transporting a returned non-serviceable item to either its stocking location or to the disposal facility is ignored. In our models, both types of serviceable items can be in stock simultaneously. However, the method that we use eliminates the need for a stock depletion rule by using the same holding cost rate for all serviceable items. It turns out, that holding cost rates for remanufactured items and manufactured items become $h + \alpha c_M$, respectively. Holding cost is calculated by multiplying the holding cost rate by the corresponding average queue length.

If an arriving demand cannot be satisfied immediately, it is backordered until a finished product becomes available. This matter leads to a backorder cost. It is obtained by multiplying the unit backorder cost by the average queue length of backorders.

When the current number in the cores' buffer reaches the maximum capacity, further incoming cores are disposed of. Refusing a core causes an expense called the disposal cost. The difference between the return arrival rate and the throughput of the remanufacturing process implies the disposal rate of the cores. Multiplying the unit disposal cost by this difference denotes the disposal cost. In fact, disposal cost can be positive or negative. If an end of life product has a salvage value, it may be sold to a recycling facility. Here, disposal cost takes negative value since it implies revenue. However, in case the end of life product does not have a salvage value, disposal results in cost and not in revenue, and disposal cost takes a positive value.

The production cost implies processing cost of machines and all other operation costs like cost of consumable parts/liquids etc. It is obtained by multiplying the unit production cost by the corresponding expected throughput.

Before summing up the above explanations using formulas, the necessary notations are as follows:

Table 4.25. The necessary notations for cost structure

λ_D	external demand rate
γ	return rate
r	static routing probability
α	inventory carrying charge per month
h	unit out-of-pocket holding cost rate per month
c_M	marginal cost for manufacturing one item
c_R	marginal cost for remanufacturing one item
c_D	cost for disposing one item
b	unit backorder cost per month

Table 4.26. Underlying cost components

Holding cost for non-serviceable items = $[h+\alpha(c_M - c_R)][QR + WIPR + WIPM]$
Holding cost for serviceable items = $[h+\alpha\, c_M][QFG]$
Backorder cost = $b[QD]$
Disposal cost = $c_D[\gamma - r\lambda_D]$
Production cost for remanufacturing = $c_R[TH_R]$
Production cost for manufacturing = $c_M[TH_M]$

The components of the total cost function are given in Table 4.26. Here, *QR* stands for the average queue length of cores. *WIPR* and *WIPM* represent the average work-in-process of remanufacturing and manufacturing (non-serviceable items), respectively. *QFG* stands for the average queue length of finished (serviceable items) goods. *QD* represents the average backorder queue. TH_R and TH_M imply expected throughputs of remanufacturing and manufacturing processes, respectively. Summation of the above cost components becomes our total cost function as given below.

$$Z = [h+\alpha(c_M - c_R)][QR + WIPR + WIPM] + [h+\alpha\, c_M][QFG] + b[QD] + \\ + c_D[\gamma - r\lambda_D] + c_R[TH_R] + c_M[TH_M] \quad (4.47)$$

4.4.1. Parameter Optimization for KCS

To make reasonable comparisons among KCS, BSCS, EKCS and GKCS we need to minimize the above total cost function. We want to minimize the total cost with respect to the static routing probability and the kanban size, since KCS has one-parameter per stage, viz. the kanban size. We search for the configuration which has the minimum cost.

To find the optimal configuration, an enumerative method is utilized that is proposed by Duri *et al.*, (2000b). All possible configurations are tested in the order of given by the algorithm. Considering the total cost as a function of the kanban size and the static routing probability, we initialize these parameters, then we increase them by one and 0.05, respectively. This incrementation goes on until the parameters get their upper limits. For each configuration, the performance measures are estimated using the analytical method described in the previous section.

The underlying algorithm is given below. In the algorithm, Z^* represents the minimum total average cost. K^* and r^* stand for optimal kanban size and optimal static routing probability, respectively. Since any additional increase upon the upper limit 20 does not result in significant changes in the total average cost, we chose the upper limit of parameter values as 20. We perform a full state space search and obtain near-optimal results for cost and control values.

Algorithm:

Set Z^* equal to a very large number
For $K = 1$ to 20
    For $r = 0.05$ to min $(1, \gamma / \lambda_D)$
        Calculate total cost Z by using the approximation method
        Optimality check: If $Z < Z^*$, then $Z^* = Z$, $K^* = K$ and $r^* = r$
    End For
End For

Figure 4.17. Optimization algorithm for KCS

4.4.2. Parameter Optimization for GKCS

Since GKCS has two parameters per stage, namely, both kanban size and base stock level, we not only optimize the static routing probability and the kanban size, but also the base stock level where the total cost is minimized.

As we described in the previous section, GKCS is analyzed by examining two cases. The first case is where $K \geq S$, and second case is where $K < S$. Therefore, cost optimization is also done using two algorithms. Algorithm 1 represents the first case, and algorithm 2 represents the second case.

Algorithm 1:

Set Z^* equal to a very large number

For $K = 1$ to 20

For $S = 0$ to K

For $r = 0.05$ to $\min(1, \gamma / \lambda_D)$

Calculate total cost Z by using the approximation method

Optimality : If $Z < Z^*$, then $Z^* = Z$, $K^* = K$, $S^* = S$ and $r^* = r$

End For

End For

End For

Figure 4.18. Optimization algorithm for GKCS-1st case

Algorithm 2:

Set Z^* equal to a very large number

For $K = 1$ to 20

For $S = K+1$ to 20

For $r = 0.05$ to $\min(1, \gamma / \lambda_D)$

Calculate total cost Z by using the approximation method

Optimality : If $Z < Z^*$, then $Z^* = Z$, $K^* = K$, $S^* = S$ and $r^* = r$

End For

End For

End For

Figure 4.19. Optimization algorithm for GKCS-2nd case

4.4.3. Parameter Optimization for EKCS

EKCS has again two parameters, viz. the kanban size and the base stock level. Since we have dedicated kanbans, we not only optimize the base stock level and the static routing probability, but also the remanufacturing and manufacturing kanbans. The underlying algorithm is given below.

K_R and K_M stand for kanbans dedicated for remanufacturing process and manufacturing process, respectively.

```
Algorithm:
Set Z* equal to a very large number
For K_R = 1 to 10
        For K_M = 1 to 10
          For S = 0 to K_R + K_M
              For r = 0.05 to min (1, γ / λ_D)
                  Calculate total cost Z by using the approximation method
                  Optimality : If Z < Z* , then Z* = Z, K_R * = K_R, K_M *= K_M,
     S* = S and r* = r
              End For
          End For
        End For
End For
```

Figure 4.20. Optimization algorithm for EKCS

4.4.4. Parameter Optimization for BSCS

In the BSCS, we optimize the base stock level and the static routing probability while we minimize the total average cost.

Algorithm:

Set Z^* equal to a very large number

Set K_R and K_M to a very large number

For $S = 0$ to 20

    For $r = 0.05$ to min $(1, \gamma / \lambda_D)$

        Calculate total cost Z by using the approximation method

        Optimality : If $Z < Z^*$, then $Z^* = Z$, $S^* = S$ and $r^* = r$

    End For

End For

Figure 4.21. Optimization algorithm for BSCS

5. NUMERICAL ANALYSIS

We design our experiment data set according to Taguchi method using an orthogonal array $L_{18}(2^1 3^7)$ where we assign two levels to factor b, and three levels to the remaining factors. There are six cost parameters of interest to us in this study, viz. b, h, α, c_R, c_M and c_D (Table 4.25). Since remanufacturing is cheaper than manufacturing plus disposing, $c_R < c_M + c_D$ (Teunter *et al.*, 2000) has to be observed. Initially, we conduct experiments where the return arrival rate and the demand arrival rate are set to 360 and 400 (products/month), respectively. The service rates for remanufacturing and manufacturing machines are chosen as 800 (products/month), respectively. The tables below (Table 5.1 to Table 5.5) demonstrate the results when the buffer size of cores is equal to ten, seven, five, three and one. For each configuration of six factors, we compare the optimal total average cost of each control policy. Then, from each experiment set we pick out the experiments for sensitivity analysis whose calculated per cent cost savings over BSCS are significantly high in order to be able to observe the behavior of the system easier. According to calculated per cent cost savings, experiment 11 and experiment five are selected for B=10 and B=1, respectively.

Table 5.1. Cost comparison among control policies in case B = 10

Ex	***b***	***h***	***α***	c_R	c_M	c_D	**KCS**	**BSCS**	**EKCS**	**GKCS**
1	1600	400	0.005	5	20	-2.5	5941.24	7601.56	5619.15	5873.49
2	1600	400	0.010	10	25	0	7991.71	9792.95	7669.65	7924.07
3	1600	400	0.020	15	30	2.5	10042.71	11944.35	9720.69	9975.34
4	1600	800	0.005	5	25	0	8404.60	11257.49	8094.46	8404.60
5	1600	800	0.010	10	30	2.5	10455.20	13408.23	10144.97	10455.20
6	1600	800	0.020	15	20	-2.5	11454.87	13607.62	11144.59	11454.87
7	1600	1200	0.005	10	30	-2.5	12381.53	15916.97	11937.22	12329.01
8	1600	1200	0.010	15	20	0	13531.34	16566.66	13087.03	13478.82
9	1600	1200	0.020	5	25	2.5	10482.96	14219.10	10038.65	10430.43
10	3200	400	0.005	15	25	2.5	10631.08	11882.09	9804.76	10157.16
11	3200	400	0.010	5	30	-2.5	7519.79	9283.82	6605.95	6958.47
12	3200	400	0.020	10	20	0	8532.27	9683.57	7755.94	8108.52
13	3200	800	0.005	10	20	2.5	10831.96	13402.23	10187.77	10696.37
14	3200	800	0.010	15	25	-2.5	12732.32	15002.80	12088.15	12596.85
15	3200	800	0.020	5	30	0	9684.39	13006.26	9040.25	9549.18
16	3200	1200	0.005	15	30	0	15195.34	19071.77	14604.17	15195.34
17	3200	1200	0.010	5	20	2.5	11245.71	15272.35	10654.54	11245.71
18	3200	1200	0.020	10	25	-2.5	13146.65	16873.63	12555.52	13146.65

Table 5.2. Cost comparison among control policies in case $B = 7$

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	1600	400	0.005	5	20	-2.5	5757.03	6861.06	5243.62	5649.23
2	1600	400	0.010	10	25	0	7857.46	9061.60	7294.10	7699.77
3	1600	400	0.020	15	30	2.5	9958.38	11262.75	9345.12	9750.97
4	1600	800	0.005	5	25	0	7753.85	9911.13	7143.42	7722.69
5	1600	800	0.010	10	30	2.5	9804.27	12111.70	9193.88	9773.11
6	1600	800	0.020	15	20	-2.5	10804.08	11911.26	10193.73	10772.92
7	1600	1200	0.005	10	30	-2.5	11117.56	13812.75	10470.46	11084.52
8	1600	1200	0.010	15	20	0	12267.47	14229.36	11620.37	12234.43
9	1600	1200	0.020	5	25	2.5	9218.66	12114.14	8571.37	9185.62
10	3200	400	0.005	15	25	2.5	10282.25	11148.84	9528.21	9982.64
11	3200	400	0.010	5	30	-2.5	7335.01	8750.14	6329.15	6895.68
12	3200	400	0.020	10	20	0	8133.23	8900.00	7479.30	7883.94
13	3200	800	0.005	10	20	2.5	10213.58	11813.08	9436.79	10205.95
14	3200	800	0.010	15	25	-2.5	12013.93	13313.54	11337.20	12006.41
15	3200	800	0.020	5	30	0	9315.75	11524.28	8288.84	9100.31
16	3200	1200	0.005	15	30	0	14393.24	16673.73	13325.00	14286.27
17	3200	1200	0.010	5	20	2.5	10493.57	12903.57	9375.21	10336.54
18	3200	1200	0.020	10	25	-2.5	12294.43	14425.12	11275.92	12237.40

Table 5.3. Cost comparison among control policies in case $B = 5$

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	1600	400	0.005	5	20	-2.5	5663.35	6452.52	5235.88	5574.16
2	1600	400	0.010	10	25	0	7813.73	8702.97	7336.33	7674.58
3	1600	400	0.020	15	30	2.5	9939.15	10953.94	9437.32	9775.51
4	1600	800	0.005	5	25	0	7567.33	9139.54	7099.05	7567.33
5	1600	800	0.010	10	30	2.5	9667.73	11339.94	9149.51	9667.73
6	1600	800	0.020	15	20	-2.5	10248.44	10787.28	9805.46	10248.44
7	1600	1200	0.005	10	30	-2.5	10796.46	12396.08	10118.16	10796.46
8	1600	1200	0.010	15	20	0	11696.38	12690.83	11074.98	11696.38
9	1600	1200	0.020	5	25	2.5	8997.50	10797.19	8218.92	8997.50
10	3200	400	0.005	15	25	2.5	10121.54	10747.89	9461.02	9892.85
11	3200	400	0.010	5	30	-2.5	7392.41	8415.56	6461.96	6931.81
12	3200	400	0.020	10	20	0	7922.36	8448.84	7362.11	7744.04
13	3200	800	0.005	10	20	2.5	9776.31	10842.03	9171.34	9776.31
14	3200	800	0.010	15	25	-2.5	11476.61	12242.42	10947.65	11476.61
15	3200	800	0.020	5	30	0	9128.14	10706.78	8273.31	8949.81
16	3200	1200	0.005	15	30	0	13711.13	15182.64	12859.02	13711.13
17	3200	1200	0.010	5	20	2.5	9861.40	11432.93	8959.24	9861.40
18	3200	1200	0.020	10	25	-2.5	11562.15	12893.68	10759.90	11562.15

Table 5.4. Cost comparison among control policies in case B = 3

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	1600	400	0.005	5	20	-2.5	5698.17	6163.27	5285.76	5632.24
2	1600	400	0.010	10	25	0	7898.46	8463.62	7436.13	7813.00
3	1600	400	0.020	15	30	2.5	10099.09	10728.56	9586.96	9964.05
4	1600	800	0.005	5	25	0	7574.47	8468.88	7028.89	7574.47
5	1600	800	0.010	10	30	2.5	9724.78	10719.27	9179.20	9724.78
6	1600	800	0.020	15	20	-2.5	9617.19	9721.57	9214.54	9556.36
7	1600	1200	0.005	10	30	-2.5	10423.11	11249.86	9822.65	10423.11
8	1600	1200	0.010	15	20	0	10986.68	11217.73	10344.80	10832.13
9	1600	1200	0.020	5	25	2.5	8723.90	9750.62	8123.24	8723.90
10	3200	400	0.005	15	25	2.5	10060.05	10454.92	9513.82	9902.31
11	3200	400	0.010	5	30	-2.5	7660.80	8276.19	6766.90	7264.63
12	3200	400	0.020	10	20	0	7806.67	8105.87	7314.61	7692.34
13	3200	800	0.005	10	20	2.5	9445.40	9963.19	8878.43	9445.40
14	3200	800	0.010	15	25	-2.5	10943.18	11263.50	10443.37	10943.18
15	3200	800	0.020	5	30	0	9197.39	10127.80	8372.75	9065.83
16	3200	1200	0.005	15	30	0	13140.38	13866.38	12422.42	13140.38
17	3200	1200	0.010	5	20	2.5	9340.57	10199.14	8622.61	9340.57
18	3200	1200	0.020	10	25	-2.5	10915.30	11517.10	10223.20	10915.30

Table 5.5. Cost comparison among control policies in case B = 1

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	1600	400	0.005	5	20	-2.5	5999.54	6038.10	5866.67	5999.54
2	1600	400	0.010	10	25	0	8299.85	8356.90	8175.86	8299.85
3	1600	400	0.020	15	30	2.5	10600.54	10657.60	10476.55	10600.54
4	1600	800	0.005	5	25	0	8082.92	8094.56	7825.16	8028.31
5	1600	800	0.010	10	30	2.5	10333.25	10394.81	10125.39	10328.56
6	1600	800	0.020	15	20	-2.5	9007.02	8820.73	8590.24	8725.06
7	1600	1200	0.005	10	30	-2.5	10596.75	10514.01	10146.68	10461.96
8	1600	1200	0.010	15	20	0	10274.30	9955.05	9632.23	9907.86
9	1600	1200	0.020	5	25	2.5	9197.32	9144.24	8791.50	9067.25
10	3200	400	0.005	15	25	2.5	10265.89	10305.71	10072.70	10263.35
11	3200	400	0.010	5	30	-2.5	8187.36	8221.91	8004.25	8138.56
12	3200	400	0.020	10	20	0	7907.21	7912.23	7676.96	7893.67
13	3200	800	0.005	10	20	2.5	9472.59	9423.70	9053.20	9411.81
14	3200	800	0.010	15	25	-2.5	10686.11	10523.93	10201.05	10511.60
15	3200	800	0.020	5	30	0	9800.05	9877.26	9534.36	9800.05
16	3200	1200	0.005	15	30	0	13074.01	12906.41	12454.64	12880.85
17	3200	1200	0.010	5	20	2.5	9474.14	9306.55	8854.77	9280.99
18	3200	1200	0.020	10	25	-2.5	10674.60	10507.03	10055.22	10481.47

As observed from Table 5.1 to Table 5.4, the worst control policy is BSCS, the second worst policy is KCS. GKCS is better than KCS, and EKCS performs the best. However, based on Table 5.5, we observe, that when b and h take their highest values at the same time, or also, when *b* takes its smallest value and *h* takes its highest value, BSCS performs better than KCS. This fact is further analyzed by the sensitivity analysis.

In the following sections, tables related to the average queue lengths of each policy will be given when necessary. *QR*, *QFP*, *QBL*, *WIPR*, *WIPM* and *FB* are abbreviations for the average return queue, finished product queue, backlog queue, work-in-process for remanufacturing, work-in-process for manufacturing and fraction of backorders, respectively.

5.1. Change in Unit Backorder Cost

Cost comparison among control policies as *b* changes is given in Table 5.6, together with related cost parameters. The increasing behavior of backorder cost forces not only the EKCS, but all the systems to get finished products available, in order to respond to the demand quicker (Table 5.9). An increase in unit backorder cost per month causes a decline in optimal static routing probability r^* and an increase in optimal base stock level S^* in the BSCS (Table 5.7). That is why the return queue is getting larger since more items are pulled by the manufacturing process, hence *WIPR* is decreasing while *WIPM* is increasing (Table 5.9). Similarly, in the KCS as backorder cost increases, optimal total cost also increases where optimal static routing probability r^* decreases, but optimal kanban size K^* increases. These reasons lead these two systems, viz. BSCS and KCS to have a worse performance than the more complicated GKCS and EKCS, especially for higher backorder costs (Figure 5.1).

In the GKCS, the optimal static routing probability and optimal kanban size remain the same whereas optimal base stock level rises with an increase in unit backorder cost. GKCS allows the finished product queue to increase by ascending *S*, and, it does not change *WIP* values by holding *K* constant (Table 5.9), because the transfer of demands from downstream to upstream can be done independently of the consumption of a finished product. In other words, its two-parameter structure helps GKCS have a better performance than KCS. In the EKCS, manufacturing kanbans increase while static routing probabilities do not change with an increase in *b*. High routing probabilities mean, that the demands are mainly directed to remanufacturing process, and so the related *WIPR*s take high values in comparison to *WIPM*. That's why the manufacturing kanbans increase instead of remanufacturing kanbans. Optimal base stock levels also increase. The changes in these control parameters are compensated by holding r^* constant. This is the main reason for EKCS being the best of all policies. Table 5.8 and Figure 5.2 indicate that KCS is concave whereas EKCS is converging. This is because as the backorder cost increases, EKCS becomes the most beneficial policy due to its multi-parameter structure. KCS has one control parameter which is not sufficient for higher punishments. Therefore, one should avoid using simple control policies such as KCS and BSCS for higher values of backorder punishment, whereas for lower backorder punishment values, only avoiding BSCS is sufficient since the other three policies behave almost the same (Figure 5.1 and Figure 5.2).

Table 5.6. Cost comparison among control policies as *b* changes and *B*=10 (Ex11)

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	400	400	0.01	5	30	-2.5	5728.07	7208.15	5621.07	5712.02
2	800	400	0.01	5	30	-2.5	6053.39	7980.17	5898.20	6053.39
3	1200	400	0.01	5	30	-2.5	6297.79	8345.33	6069.51	6297.79
4	1600	400	0.01	5	30	-2.5	6542.19	8643.66	6220.13	6474.59
5	2000	400	0.01	5	30	-2.5	6786.59	8923.30	6360.48	6625.42
6	2400	400	0.01	5	30	-2.5	7030.99	9043.47	6442.30	6772.30
7	2800	400	0.01	5	30	-2.5	7275.39	9163.65	6524.13	6865.38
8	3200	400	0.01	5	30	-2.5	7519.79	9283.82	6605.95	6958.47
9	3600	400	0.01	5	30	-2.5	7752.51	9389.84	6687.78	7051.56
10	4000	400	0.01	5	30	-2.5	7872.59	9468.57	6758.90	7144.65

Table 5.7. Optimal parameters as the unit backorder cost changes and *B*=10 (Ex11)

	KCS			BSCS			EKCS					GKCS			
b	Z^*	r^*	K^*	Z^*	r^*	S^*	Z^*	r^*	S^*	K_R^*	K_M^*	Z^*	r^*	S^*	K^*
400	5728.1	0.85	1	7208.2	0.80	1	5621.1	0.85	1	1	1	5712.0	0.85	1	2
800	6053.4	0.85	2	7980.2	0.75	1	5898.2	0.85	1	1	1	6053.4	0.85	2	2
1200	6297.8	0.85	2	8345.3	0.75	2	6069.5	0.85	2	1	1	6297.8	0.85	2	2
1600	6542.2	0.85	2	8643.7	0.75	2	6220.1	0.85	2	1	1	6474.6	0.85	3	2
2000	6786.6	0.85	2	8923.3	0.70	2	6360.5	0.85	3	1	2	6625.4	0.85	3	2
2400	7031.0	0.85	2	9043.5	0.70	2	6442.3	0.85	3	1	2	6772.3	0.85	4	2
2800	7275.4	0.85	2	9163.6	0.70	2	6524.1	0.85	3	1	2	6865.4	0.85	4	2
3200	7519.8	0.85	2	9283.8	0.70	2	6606.0	0.85	3	1	2	6958.5	0.85	4	2
3600	7752.5	0.80	2	9389.8	0.70	3	6687.8	0.85	3	1	2	7051.6	0.85	4	2
4000	7872.6	0.80	2	9468.6	0.70	3	6758.9	0.85	4	1	3	7144.6	0.85	4	2

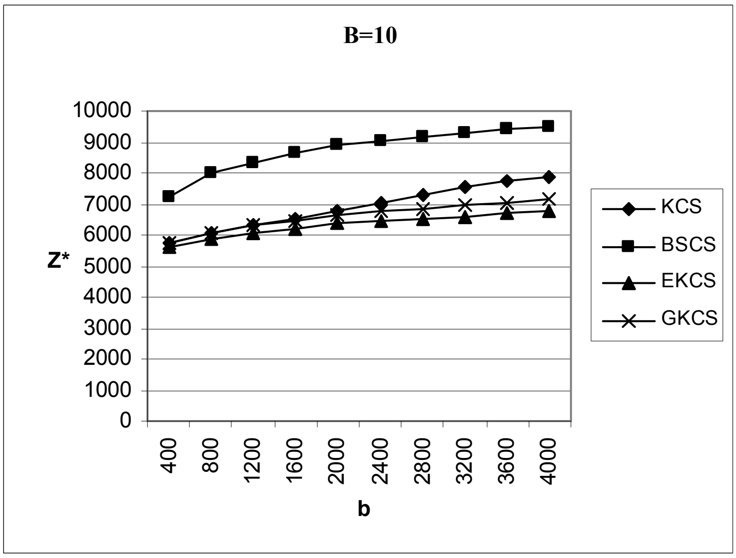

Figure 5.1. Optimal average total cost comparison as the unit backorder cost per month changes (Ex11)

Table 5.8. Per cent cost savings over BSCS as *b* changes and *B*=10 (Ex11)

b	**KCS**	**EKCS**	**GKCS**
400	20.5%	22.0%	20.8%
800	24.1%	26.1%	24.1%
1200	24.5%	27.3%	24.5%
1600	24.3%	28.0%	25.1%
2000	23.9%	28.7%	25.8%
2400	22.3%	28.8%	25.1%
2800	20.6%	28.8%	25.1%
3200	19.0%	28.8%	25.0%
3600	17.4%	28.8%	24.9%
4000	16.9%	28.6%	24.5%

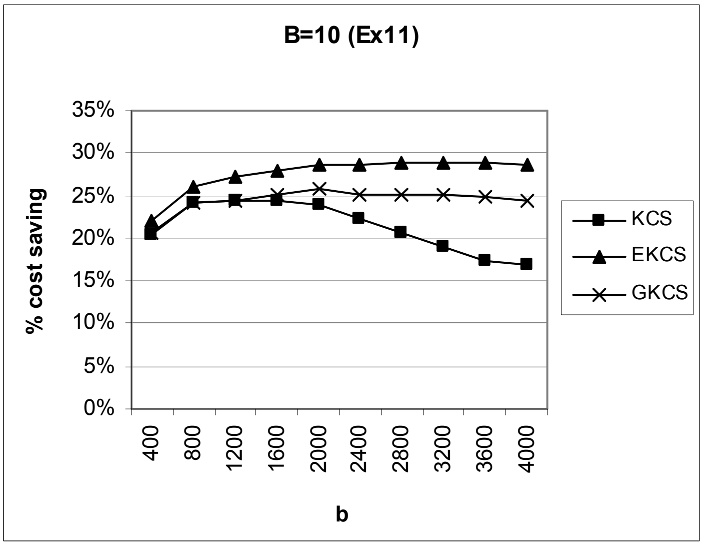

Figure 5.2. Per cent cost savings over BSCS as *b* changes

Table 5.9. Average queue lengths of each policy

	KCS (B=10) and Ex11						BSCS (B=10) and Ex11					
b	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**
400	3.81	0.38	1.00	0.42	0.07	0.62	4.46	0.39	2.63	0.67	0.11	0.61
800	3.59	1.01	0.61	0.61	0.08	0.38	5.81	0.47	1.02	0.60	0.14	0.53
1200	3.59	1.01	0.61	0.61	0.08	0.38	5.81	1.20	0.75	0.60	0.14	0.28
1600	3.59	1.01	0.61	0.61	0.08	0.38	5.81	1.20	0.75	0.60	0.14	0.28
2000	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
2400	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
2800	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
3200	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
3600	5.08	1.19	0.30	0.55	0.11	0.28	6.78	2.20	0.20	0.54	0.18	0.10
4000	5.08	1.19	0.30	0.55	0.11	0.28	6.78	2.20	0.20	0.54	0.18	0.10
	EKCS (B=10) and Ex11						**GKCS (B=10) and Ex11**					
b	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**
400	3.81	0.42	0.69	0.43	0.07	0.58	3.59	0.39	0.99	0.61	0.08	0.61
800	3.81	0.42	0.69	0.43	0.07	0.58	3.59	1.01	0.61	0.61	0.08	0.38
1200	3.81	1.11	0.38	0.43	0.07	0.32	3.59	1.01	0.61	0.61	0.08	0.38
1600	3.81	1.11	0.38	0.43	0.07	0.32	3.59	2.63	0.23	0.61	0.08	0.14
2000	3.81	1.93	0.20	0.43	0.08	0.17	3.59	2.63	0.23	0.61	0.08	0.14
2400	3.81	1.93	0.20	0.43	0.08	0.17	3.59	2.63	0.23	0.61	0.08	0.14
2800	3.81	1.93	0.20	0.43	0.08	0.17	3.59	2.63	0.23	0.61	0.08	0.14
3200	3.81	1.93	0.20	0.43	0.08	0.17	3.59	2.63	0.23	0.61	0.08	0.14
3600	3.81	1.93	0.20	0.43	0.08	0.17	3.59	2.63	0.23	0.61	0.08	0.14
4000	3.81	2.84	0.11	0.43	0.08	0.09	3.59	2.63	0.23	0.61	0.08	0.14

The increase in the punishment of backorder cost forces the backorder queue to decrease in all policies. When buffer size of the cores is small, optimal routing probabilities of all policies are decreasing, because there occurs a delay due to having very few cores. This matter results in the preference of having manufactured products in the output buffer, instead of remanufactured products. Therefore, when the backorder cost increases, less demand is directed to the remanufacturing facility, resulting in a *WIPM* increase and a *WIPR* decrease simultaneously (Table 5.13). In the BSCS, there is a decline in routing probabilities and an increase in optimal base stock sizes (Table 5.11). Hence, less demand is satisfied by remanufacturing. In the KCS, as the backorder cost increases, the optimal static routing probability decreases, but the optimal kanban size remains the same. So, BSCS performs worse than KCS, since BSCS results in higher inventory holding costs. On the other hand, in the GKCS, as the backorder cost increases, the optimal routing probabilities decrease whereas optimal base stock level increases and kanban size remains the same. Thus, when S is not equal to K, it results in a lower cost than KCS as a consequence of its lower finished product inventory costs. However, when S is equal to K, it is equivalent to KCS (Figure 5.4). In the EKCS, an increase in backorder cost results in a decline in optimal static routing probabilities. The output buffer is mostly filled with manufactured products instead of remanufactured ones. The increase in manufacturing kanban sizes is an indicator to this behavior. Since Figure 5.3 does not provide a very clear picture, we draw a graph that reflects per cent cost savings over BSCS (Figure 5.4). For small values of b, KCS performs worse than BSCS, whereas for larger values of b, it performs better than BSCS. As b is getting larger, GKCS behaves like KCS. This follows from the fact that, when GKCS has S=K, it is equivalent to KCS (Frein *et al.*, 1995). However, EKCS is the leading policy when considering its increasing per cent cost savings with an increase in b (Table 5.12 and Figure 5.4).

Table 5.10. Cost comparison among control policies as b changes and B=1 (Ex5)

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	400	800	0.01	10	30	2.5	8654.73	8586.76	8470.68	8527.84
2	800	800	0.01	10	30	2.5	9453.84	9479.35	9303.98	9394.90
3	1200	800	0.01	10	30	2.5	9975.63	10006.58	9781.26	9916.43
4	1600	800	0.01	10	30	2.5	10333.25	10394.81	10125.39	10328.56
5	2000	800	0.01	10	30	2.5	10639.98	10719.57	10424.69	10639.98
6	2400	800	0.01	10	30	2.5	10859.75	10950.84	10624.12	10859.75
7	2800	800	0.01	10	30	2.5	11079.52	11182.11	10823.55	11079.52
8	3200	800	0.01	10	30	2.5	11273.09	11326.42	10983.56	11273.09
9	3600	800	0.01	10	30	2.5	11412.32	11469.71	11094.97	11412.32
10	4000	800	0.01	10	30	2.5	11551.54	11612.99	11206.38	11551.54

Table 5.11. Optimal parameters as the unit backorder cost changes and *B*=1 (Ex5)

	KCS			**BSCS**			**EKCS**					**GKCS**			
b	***Z****	***r****	***K****	***Z****	***r****	***S****	***Z****	***r****	***S****	K_R*	K_M*	***Z****	***r****	***S****	***K****
400	8654.7	0.75	3	8586.8	0.70	1	8470.7	0.70	1	2	1	8527.8	0.70	1	3
800	9453.8	0.70	3	9479.3	0.65	2	9304.0	0.65	2	2	1	9394.9	0.70	2	3
1200	9975.6	0.65	3	10006.6	0.65	2	9781.3	0.65	2	2	1	9916.4	0.65	2	3
1600	10333.3	0.65	3	10394.8	0.60	2	10125.4	0.60	2	1	1	10328.6	0.60	2	3
2000	10640.0	0.60	3	10719.6	0.60	3	10424.7	0.55	2	1	1	10640.0	0.60	3	3
2400	10859.7	0.60	3	10950.8	0.60	3	10624.1	0.55	2	1	1	10859.7	0.60	3	3
2800	11079.5	0.60	3	11182.1	0.60	3	10823.6	0.55	2	1	1	11079.5	0.60	3	3
3200	11273.1	0.55	3	11326.4	0.55	3	10983.6	0.55	3	1	2	11273.1	0.55	3	3
3600	11412.3	0.55	3	11469.7	0.55	3	11095.0	0.55	3	1	2	11412.3	0.55	3	3
4000	11551.5	0.55	3	11613.0	0.55	3	11206.4	0.55	3	1	2	11551.5	0.55	3	3

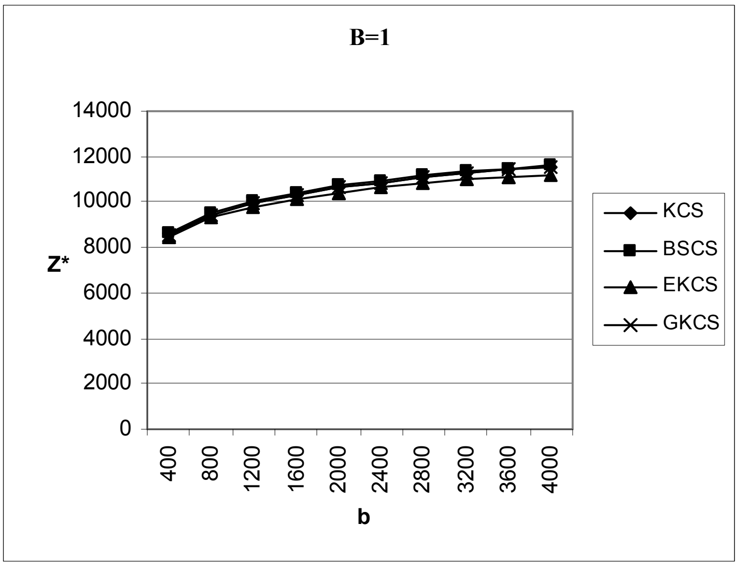

Figure 5.3. Optimal average total cost comparison as the unit backorder cost per month changes (Ex5)

Table 5.12. Per cent cost savings over BSCS as *b* changes and *B*=1 (Ex5)

b	KCS	EKCS	GKCS
400	-0.8%	1.4%	0.7%
800	0.3%	1.9%	0.9%
1200	0.3%	2.3%	0.9%
1600	0.6%	2.6%	0.6%
2000	0.7%	2.8%	0.7%
2400	0.8%	3.0%	0.8%
2800	0.9%	3.2%	0.9%
3200	0.5%	3.0%	0.5%
3600	0.5%	3.3%	0.5%
4000	0.5%	3.5%	0.5%

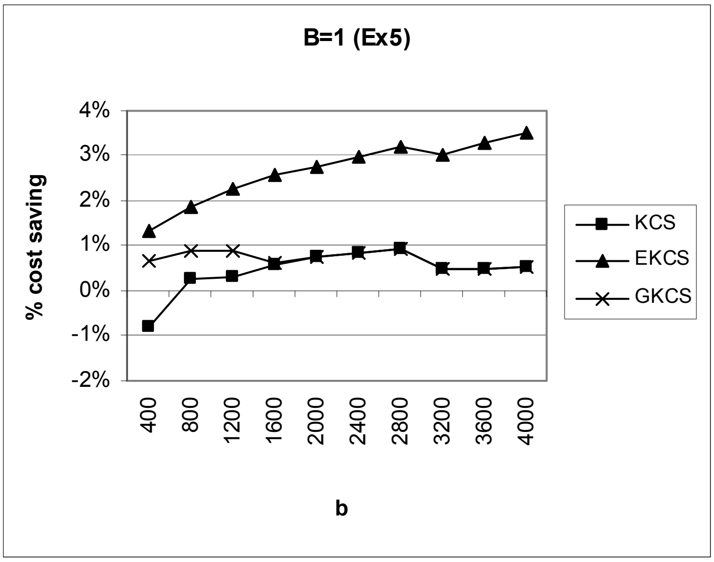

Figure 5.4. Per cent cost savings over BSCS as *b* changes

Table 5.13. Average queue lengths of each policy

	KCS (B=1) and Ex5						BSCS (B=1) and Ex5					
b	QR	QFP	QBL	WIPR	WIPM	FB	QR	QFP	QBL	WIPR	WIPM	FB
400	0.17	0.88	2.75	0.56	0.14	0.58	0.22	0.22	2.66	0.54	0.18	0.78
800	0.22	1.14	1.52	0.51	0.18	0.46	0.28	0.75	1.32	0.48	0.21	0.52
1200	0.28	1.37	0.89	0.46	0.21	0.37	0.28	0.75	1.32	0.48	0.21	0.52
1600	0.28	1.37	0.89	0.46	0.21	0.37	0.33	0.86	0.87	0.43	0.25	0.45
2000	0.33	1.56	0.55	0.41	0.25	0.30	0.33	1.57	0.58	0.43	0.25	0.29
2400	0.33	1.56	0.55	0.41	0.25	0.30	0.33	1.57	0.58	0.43	0.25	0.29
2800	0.33	1.56	0.55	0.41	0.25	0.30	0.33	1.57	0.58	0.43	0.25	0.29
3200	0.39	1.72	0.35	0.37	0.28	0.24	0.39	1.73	0.36	0.38	0.29	0.23
3600	0.39	1.72	0.35	0.37	0.28	0.24	0.39	1.73	0.36	0.38	0.29	0.23
4000	0.39	1.72	0.35	0.37	0.28	0.24	0.39	1.73	0.36	0.38	0.29	0.23
	EKCS (B=1) and Ex5						GKCS (B=1) and Ex5					
b	QR	QFP	QBL	WIPR	WIPM	FB	QR	QFP	QBL	WIPR	WIPM	FB
400	0.22	0.21	2.57	0.47	0.15	0.79	0.22	0.21	2.58	0.51	0.18	0.79
800	0.28	0.74	1.19	0.43	0.18	0.52	0.22	0.61	1.98	0.51	0.18	0.60
1200	0.28	0.74	1.19	0.43	0.18	0.52	0.28	0.74	1.26	0.46	0.21	0.52
1600	0.33	0.83	0.81	0.30	0.20	0.46	0.33	0.85	0.84	0.41	0.25	0.45
2000	0.39	0.96	0.50	0.28	0.23	0.39	0.33	1.56	0.55	0.41	0.25	0.30
2400	0.39	0.96	0.50	0.28	0.23	0.39	0.33	1.56	0.55	0.41	0.25	0.30
2800	0.39	0.96	0.50	0.28	0.23	0.39	0.33	1.56	0.55	0.41	0.25	0.30
3200	0.39	1.74	0.28	0.28	0.28	0.22	0.39	1.72	0.35	0.37	0.28	0.24
3600	0.39	1.74	0.28	0.28	0.28	0.22	0.39	1.72	0.35	0.37	0.28	0.24
4000	0.39	1.74	0.28	0.28	0.28	0.22	0.39	1.72	0.35	0.37	0.28	0.24

5.2. Change in Out-of-Pocket Holding Cost

In order to observe the impact of the out-of-pocket holding cost, we choose the 11th experiment from Table 5.1 (B=10), as explained in section five. The parameter values except the out-of-pocket holding cost remain constant, as given in Table 5.14. Table 5.15 presents the optimal parameters for each control policy as this change occurs.

In the BSCS, as h increases, remanufacturing facility is utilized more as a result of increasing routing probabilities and decreasing base stock levels. Hence, return queue decreases, while *WIPR* rises and *WIPM* decreases (Table 5.17). An increase in *WIPR* alone is, however, not enough to supply demand quickly due to decreasing optimal base stock levels. In the KCS, on the other hand, as the out-of-pocket holding cost rate h increases, optimal total average cost increases where optimal routing probability r^* remains the same and optimal kanban size K^* decreases. KCS compensates the decline in kanban size by holding the routing probabilities constant. That's why KCS's cost is lower than BSCS's

(Figure 5.6). GKCS decreases the finished product queue by descending *S*, and, it does not change *WIP* values by holding *K* constant, because the transfer of demands from downstream to upstream can be done independently of the consumption of a finished product. This is the main reason for GKCS performing better than KCS (Figure 5.6). In the EKCS, as *h* increases, optimal routing probability and remanufacturing kanban size remain the same, but the base stock level and the manufacturing kanban size decrease. GKCS seems to be worse than EKCS, and better than BSCS. GKCS's behavior is similar to that of KCS for *S*=*K* (Table 5.16 and Figure 5.6). The multi-parameter structure of EKCS results in the lowest minimum expected total costs (Figure 5.5). As *h* increases, when the buffer for cores has a high capacity, it is reasonable to prefer KCS. As the difference between the best control policy and this one is not significant, it is not worth investing in a sophisticated control system.

Table 5.14. Cost comparison among control policies as *h* changes and *B*=10 (Ex11)

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	3200	400	0.01	5	30	-2.5	7519.79	9283.82	6605.95	6958.47
2	3200	800	0.01	5	30	-2.5	9633.01	12803.99	8988.85	9497.58
3	3200	1200	0.01	5	30	-2.5	11746.23	16033.66	11155.08	11746.24
4	3200	1600	0.01	5	30	-2.5	13859.46	18865.33	13239.15	13859.46
5	3200	2000	0.01	5	30	-2.5	15972.68	21674.80	15131.89	15943.40
6	3200	2400	0.01	5	30	-2.5	17913.28	24484.26	17024.63	17808.23
7	3200	2800	0.01	5	30	-2.5	19789.74	27293.72	18917.37	19673.06
8	3200	3200	0.01	5	30	-2.5	21666.20	30103.19	20810.11	21537.89
9	3200	3600	0.01	5	30	-2.5	23542.66	32608.65	22702.85	23402.72
10	3200	4000	0.01	5	30	-2.5	25419.12	34862.06	24595.59	25267.55

Table 5.15. Optimal parameters as the unit out-of-pocket holding cost rate *h* changes and *B*=10 (Ex11)

	KCS			BSCS			EKCS					GKCS			
h	***Z****	***r****	***K****	***Z****	***r****	***S****	***Z****	***r****	***S****	K_R*	K_M*	***Z****	***r****	***S****	***K****
400	7519.8	0.85	2	9283.8	0.70	2	6606.0	0.85	3	1	2	6958.5	0.85	4	2
800	9633.0	0.85	2	12804.0	0.70	2	8988.8	0.85	2	1	1	9497.6	0.85	3	2
1200	11746.2	0.85	2	16033.7	0.75	2	11155.1	0.85	2	1	1	11746.2	0.85	2	2
1600	13859.5	0.85	2	18865.3	0.75	1	13239.2	0.85	1	1	1	13859.5	0.85	2	2
2000	15972.7	0.85	2	21674.8	0.75	1	15131.9	0.85	1	1	1	15943.4	0.85	1	2
2400	17913.3	0.85	1	24484.3	0.75	1	17024.6	0.85	1	1	1	17808.2	0.85	1	2
2800	19789.7	0.85	1	27293.7	0.75	1	18917.4	0.85	1	1	1	19673.1	0.85	1	2
3200	21666.2	0.85	1	30103.2	0.75	1	20810.1	0.85	1	1	1	21537.9	0.85	1	2
3600	23542.7	0.85	1	32608.6	0.80	1	22702.9	0.85	1	1	1	23402.7	0.85	1	2
4000	25419.1	0.85	1	34862.1	0.80	1	24595.6	0.85	1	1	1	25267.6	0.85	1	2

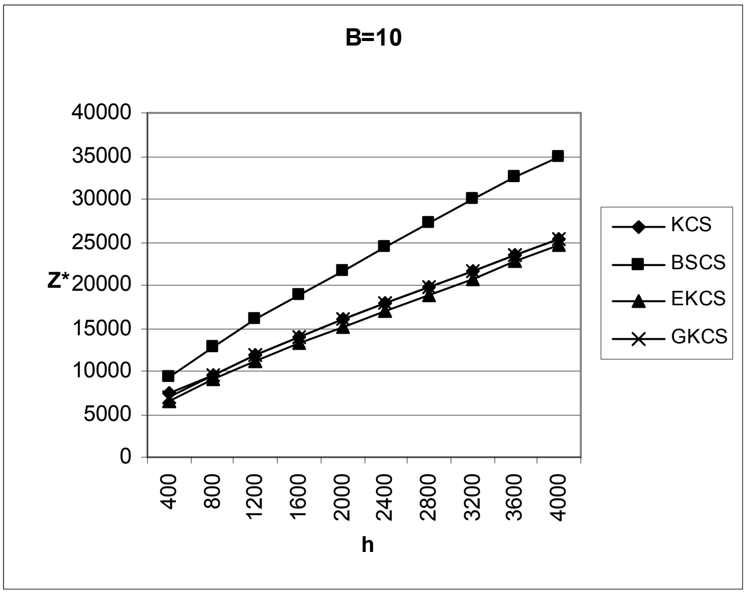

Figure 5.5. Optimal average total cost comparison as the unit out-of-pocket holding cost rate per month changes (Ex11)

Table 5.16. Per cent cost savings over BSCS as *h* changes and *B*=10 (Ex11)

h	KCS	EKCS	GKCS
400	19.0%	28.8%	25.0%
800	24.8%	29.8%	25.8%
1200	26.7%	30.4%	26.7%
1600	26.5%	29.8%	26.5%
2000	26.3%	30.2%	26.4%
2400	26.8%	30.5%	27.3%
2800	27.5%	30.7%	27.9%
3200	28.0%	30.9%	28.5%
3600	27.8%	30.4%	28.2%
4000	27.1%	29.4%	27.5%

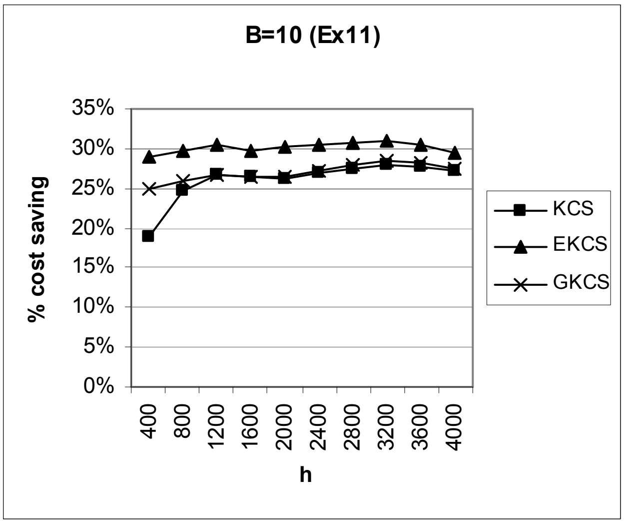

Figure 5.6. Per cent cost savings over BSCS as *h* changes

Table 5.17. Average queue lengths of each policy

	KCS (B=10) and Ex11						BSCS (B=10) and Ex11					
h	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**
400	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
800	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
1200	3.59	1.01	0.61	0.61	0.08	0.38	5.81	1.20	0.75	0.60	0.14	0.28
1600	3.59	1.01	0.61	0.61	0.08	0.38	5.81	0.47	1.02	0.60	0.14	0.53
2000	3.59	1.01	0.61	0.61	0.08	0.38	5.81	0.47	1.02	0.60	0.14	0.53
2400	3.81	0.38	1.00	0.42	0.07	0.62	5.81	0.47	1.02	0.60	0.14	0.53
2800	3.81	0.38	1.00	0.42	0.07	0.62	5.81	0.47	1.02	0.60	0.14	0.53
3200	3.81	0.38	1.00	0.42	0.07	0.62	5.81	0.47	1.02	0.60	0.14	0.53
3600	3.81	0.38	1.00	0.42	0.07	0.62	4.46	0.39	2.63	0.67	0.11	0.61
4000	3.81	0.38	1.00	0.42	0.07	0.62	4.46	0.39	2.63	0.67	0.11	0.61
	EKCS (B=10) and Ex11						**GKCS (B=10) and Ex11**					
h	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**
400	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
800	3.81	1.11	0.38	0.43	0.07	0.32	3.59	3.54	0.14	0.61	0.08	0.09
1200	3.81	1.11	0.38	0.43	0.07	0.32	3.59	1.01	0.61	0.61	0.08	0.38
1600	3.81	0.42	0.69	0.43	0.07	0.58	3.59	1.01	0.61	0.61	0.08	0.38
2000	3.81	0.42	0.69	0.43	0.07	0.58	3.59	0.39	0.99	0.61	0.08	0.61
2400	3.81	0.42	0.69	0.43	0.07	0.58	3.59	0.39	0.99	0.61	0.08	0.61
2800	3.81	0.42	0.69	0.43	0.07	0.58	3.59	0.39	0.99	0.61	0.08	0.61
3200	3.81	0.42	0.69	0.43	0.07	0.58	3.59	0.39	0.99	0.61	0.08	0.61
3600	3.81	0.42	0.69	0.43	0.07	0.58	3.59	0.39	0.99	0.61	0.08	0.61
4000	3.81	0.42	0.69	0.43	0.07	0.58	3.59	0.39	0.99	0.61	0.08	0.61

Next, we choose the fifth experiment set from Table 5.5 (B=1) and investigate the impact of the out-of-pocket holding cost. The parameter values except the out-of-pocket holding cost rate remain constant, as seen in Table 5.18. Table 5.19 reflects the optimal parameters for each control policy as this change occurs.

The fact that the routing probabilities remain the same with an increase in h, makes BSCS a preferable control system, in this case (Table 5.19). It is better than KCS in terms of average total costs, for higher values of h (Table 5.18). In the KCS, as h increases, the optimal static routing probability decreases, whereas the optimal kanban size decreases. Hence, when h takes small values, KCS is directed to remanufacture cores, while for the higher values of h, manufacturing of new products is preferable. In either case, the demands cannot be satisfied, efficiently. The average backorder queue grows very rapidly. High kanban sizes result in high holding costs, and thus KCS gets worse than BSCS (Figure 5.8). GKCS is better than KCS, because the transfer of demands from downstream to upstream can be done independently of the consumption of a finished product. In the

EKCS, routing probabilities, kanban sizes and base stock level decrease. The backorder queue is smaller than that of the other three policies (Table 5.21). Furthermore, as a consequence of its multi-parameter structure, EKCS results in lower inventory holding costs. Hence, it becomes the best of all policies (Figure 5.8). Since Figure 5.7 does not provide a clear picture, we draw a figure that gives per cent cost savings over BSCS (Figure 5.8). Table 5.20 presents per cent cost savings as *h* changes. When the buffer size of cores is small, KCS performs the worst as *h* increases, whereas the other three policies perform almost the same, especially for small values of out-of-pocket holding cost. EKCS gives still better results as a consequence of its multi-parameter structure (Figure 5.8).

Table 5.18. Cost comparison among control policies as *h* changes and *B*=1 (Ex5)

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	1600	400	0.01	10	30	2.5	9306.79	9457.02	9186.99	9283.50
2	1600	800	0.01	10	30	2.5	10333.25	10394.81	10125.39	10328.56
3	1600	1200	0.01	10	30	2.5	11196.97	11143.87	10791.16	11066.88
4	1600	1600	0.01	10	30	2.5	11900.98	11733.95	11335.89	11644.05
5	1600	2000	0.01	10	30	2.5	12604.98	12263.15	11786.56	12163.93
6	1600	2400	0.01	10	30	2.5	13245.03	12792.36	12237.22	12683.80
7	1600	2800	0.01	10	30	2.5	13880.63	13321.57	12687.89	13203.68
8	1600	3200	0.01	10	30	2.5	14516.24	13850.77	13138.56	13651.94
9	1600	3600	0.01	10	30	2.5	15151.85	14362.11	13589.22	14052.42
10	1600	4000	0.01	10	30	2.5	15787.46	14766.87	13971.50	14449.60

Table 5.19. Optimal parameters as the unit out-of-pocket holding cost rate *h* changes and *B*=1 (Ex5)

	KCS			BSCS			EKCS					GKCS			
h	Z^*	r^*	K^*	Z^*	r^*	S^*	Z^*	r^*	S^*	K_R^*	K_M^*	Z^*	r^*	S^*	K^*
400	9306.8	0.65	4	9457.0	0.60	3	9187.0	0.65	4	2	2	9283.5	0.65	4	3
800	10333.3	0.65	3	10394.8	0.60	2	10125.4	0.60	2	1	1	10328.6	0.60	2	3
1200	11197.0	0.60	2	11143.9	0.60	2	10791.2	0.60	2	1	1	11066.9	0.60	2	3
1600	11901.0	0.60	2	11733.9	0.60	1	11335.9	0.60	1	1	1	11644.1	0.60	1	3
2000	12605.0	0.60	2	12263.2	0.60	1	11786.6	0.60	1	1	1	12163.9	0.60	1	3
2400	13245.0	0.60	2	12792.4	0.60	1	12237.2	0.60	1	1	1	12683.8	0.60	1	3
2800	13880.6	0.60	2	13321.6	0.60	1	12687.9	0.60	1	1	1	13203.7	0.60	1	3
3200	14516.2	0.60	2	13850.8	0.60	1	13138.6	0.60	1	1	1	13651.9	0.60	0	3
3600	15151.8	0.60	2	14362.1	0.60	0	13589.2	0.60	1	1	1	14052.4	0.60	0	3
4000	15787.5	0.60	2	14766.9	0.60	0	13971.5	0.55	0	1	1	14449.6	0.60	0	3

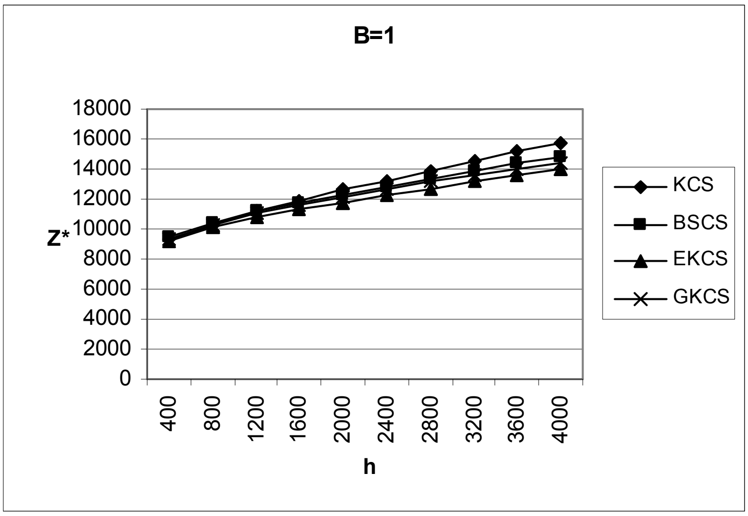

Figure 5.7. Optimal average total cost comparison as the unit out-of-pocket holding cost rate per month changes (Ex5)

Table 5.20. Per cent cost savings over BSCS as *h* changes and *B*=1 (Ex5)

h	KCS	EKCS	GKCS
400	1.6%	2.9%	1.8%
800	0.6%	2.6%	0.6%
1200	-0.5%	3.2%	0.7%
1600	-1.4%	3.4%	0.8%
2000	-2.8%	3.9%	0.8%
2400	-3.5%	4.3%	0.8%
2800	-4.2%	4.8%	0.9%
3200	-4.8%	5.1%	1.4%
3600	-5.5%	5.4%	2.2%
4000	-6.9%	5.4%	2.1%

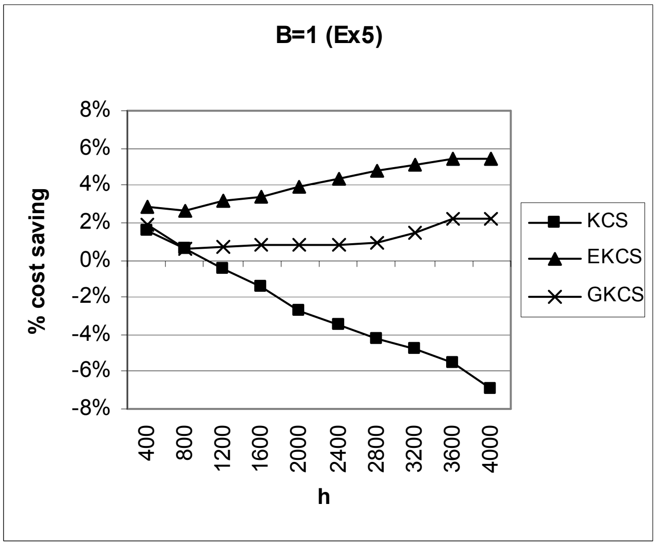

Figure 5.8. Per cent cost savings over BSCS as *h* changes

Table 5.21. Average queue lengths of each policy

	KCS (B=1) and Ex5						BSCS (B=1) and Ex5					
h	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**
400	0.28	2.12	0.64	0.48	0.21	0.26	0.33	1.57	0.58	0.43	0.25	0.29
800	0.28	1.37	0.89	0.46	0.21	0.37	0.33	0.86	0.87	0.43	0.25	0.45
1200	0.33	0.81	0.99	0.38	0.24	0.48	0.33	0.86	0.87	0.43	0.25	0.45
1600	0.33	0.81	0.99	0.38	0.24	0.48	0.33	0.31	1.32	0.43	0.25	0.69
2000	0.33	0.81	0.99	0.38	0.24	0.48	0.33	0.31	1.32	0.43	0.25	0.69
2400	0.28	0.69	1.49	0.42	0.20	0.55	0.33	0.31	1.32	0.43	0.25	0.69
2800	0.28	0.69	1.49	0.42	0.20	0.55	0.33	0.31	1.32	0.43	0.25	0.69
3200	0.28	0.69	1.49	0.42	0.20	0.55	0.33	0.31	1.32	0.43	0.25	0.69
3600	0.28	0.69	1.49	0.42	0.20	0.55	0.33	0.00	2.01	0.43	0.25	1.00
4000	0.28	0.69	1.49	0.42	0.20	0.55	0.33	0.00	2.01	0.43	0.25	1.00
	EKCS (B=1) and Ex5						**GKCS (B=1) and Ex5**					
h	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**	**QR**	**QFP**	**QBL**	**WIPR**	**WIPM**	**FB**
400	0.28	2.12	0.58	0.43	0.21	0.25	0.28	2.92	0.45	0.46	0.21	0.19
800	0.33	0.83	0.81	0.30	0.20	0.46	0.33	0.85	0.84	0.41	0.25	0.45
1200	0.33	0.83	0.81	0.30	0.20	0.46	0.33	0.85	0.84	0.41	0.25	0.45
1600	0.33	0.29	1.27	0.30	0.20	0.71	0.33	0.31	1.29	0.41	0.25	0.69
2000	0.33	0.29	1.27	0.30	0.20	0.71	0.33	0.31	1.29	0.41	0.25	0.69
2400	0.33	0.29	1.27	0.30	0.20	0.71	0.33	0.31	1.29	0.41	0.25	0.69
2800	0.33	0.29	1.27	0.30	0.20	0.71	0.33	0.31	1.29	0.41	0.25	0.69
3200	0.33	0.29	1.27	0.30	0.20	0.71	0.33	0.00	1.48	0.41	0.28	1.00
3600	0.33	0.29	1.27	0.30	0.20	0.71	0.33	0.00	1.86	0.41	0.25	1.00
4000	0.39	0.00	1.54	0.28	0.23	1.00	0.33	0.00	1.86	0.41	0.25	1.00

5.3. Change in Marginal Cost for Remanufacturing

Unit production costs for remanufacturing is determined with respect to the "profitability" of remanufacturing a core as $(c_M+c_D-c_R)/c_M$. We vary the profitability in a decreasing manner, and thereby determine c_R by holding c_D and c_M at constant values.

Initially, we use the 11th experiment, where the buffer size for cores is set equal to ten (Table 5.22). The related cost comparison figure is given in Figure 5.9. Optimal parameter values for this case, where all the parameter values except the unit production costs for remanufacturing are constant, can be found in Table 5.23.

In the BSCS, as c_R increases, the routing probability decreases but base stock level remains constant (Table 5.23). The increase in *WIPM* indicates that, in this policy also the system prefers manufacturing over remanufacturing (Table 5.25). Also, in the KCS, r^* decreases where optimal kanban size K^* remains constant. The less the profitability is, the more the system prefers manufacturing. However, routing probabilities are higher than those of BSCS, this is the reason for KCS being better than BSCS. Moreover, that *WIP* values have no bound in the BSCS, is another drawback, because the related holding costs increase accordingly (Figure 5.9). In the EKCS, since the buffer size of cores is high, the incoming demand is mainly satisfied from remanufactured items considering the routing probabilities. This is valid for GKCS also. However, using dedicated kanbans in the EKCS makes itself still a more preferable system in comparison with GKCS as a consequence of shorter backlog queues and work-in-processes (Table 5.24 and Figure 5.10).

Table 5.22. Cost comparison among control policies as c_R changes and B=10 (Ex11)

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	3200	400	0.01	0.5	30	-2.5	5989.99	8024.16	5076.14	5428.66
2	3200	400	0.01	3.5	30	-2.5	7009.86	8863.93	6096.02	6448.54
3	3200	400	0.01	6.5	30	-2.5	8029.73	9703.71	7115.89	7468.41
4	3200	400	0.01	9.5	30	-2.5	9049.60	10543.48	8135.76	8488.28
5	3200	400	0.01	12.5	30	-2.5	10032.01	11383.26	9155.64	9508.15
6	3200	400	0.01	15.5	30	-2.5	10991.83	12223.03	10175.51	10528.02
7	3200	400	0.01	18.5	30	-2.5	11951.66	13062.81	11195.38	11547.89
8	3200	400	0.01	21.5	30	-2.5	12911.49	13847.12	12215.26	12567.77
9	3200	400	0.01	24.5	30	-2.5	13871.32	14626.87	13235.13	13587.64
10	3200	400	0.01	27.5	30	-2.5	14831.15	15406.62	14255.01	14607.51

Table 5.23. Optimal parameters as c_R changes and B=10 (Ex11)

	KCS			BSCS			EKCS					GKCS			
c_R	*Z**	*r**	*K**	*Z**	*r**	*S**	*Z**	*r**	*S**	K_R*	K_M*	*Z**	*r**	*S**	*K**
0.5	5990.0	0.85	2	8024.2	0.70	2	5076.1	0.85	3	1	2	5428.7	0.85	4	2
3.5	7009.9	0.85	2	8863.9	0.70	2	6096.0	0.85	3	1	2	6448.5	0.85	4	2
6.5	8029.7	0.85	2	9703.7	0.70	2	7115.9	0.85	3	1	2	7468.4	0.85	4	2
9.5	9049.6	0.85	2	10543.5	0.70	2	8135.8	0.85	3	1	2	8488.3	0.85	4	2
12.5	10032.0	0.80	2	11383.3	0.70	2	9155.6	0.85	3	1	2	9508.2	0.85	4	2
15.5	10991.8	0.80	2	12223.0	0.70	2	10175.5	0.85	3	1	2	10528.0	0.85	4	2
18.5	11951.7	0.80	2	13062.8	0.70	2	11195.4	0.85	3	1	2	11547.9	0.85	4	2
21.5	12911.5	0.80	2	13847.1	0.65	2	12215.3	0.85	3	1	2	12567.8	0.85	4	2
24.5	13871.3	0.80	2	14626.9	0.65	2	13235.1	0.85	3	1	2	13587.6	0.85	4	2
27.5	14831.1	0.80	2	15406.6	0.65	2	14255.0	0.85	3	1	2	14607.5	0.85	4	2

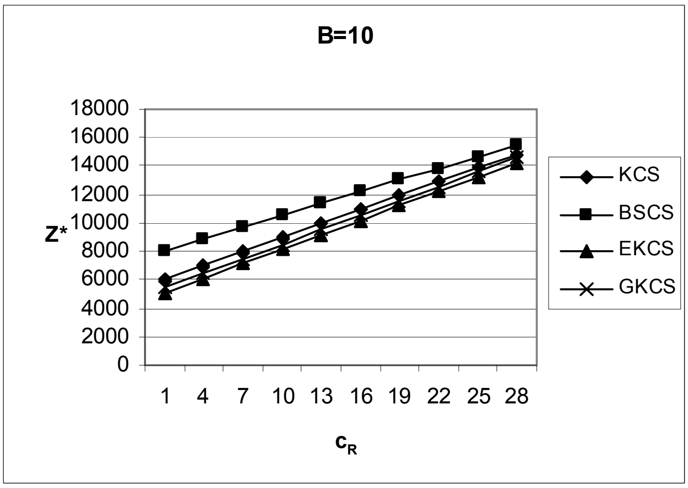

Figure 5.9. Optimal average total cost comparison as c_R changes (Ex11)

Table 5.24. Per cent cost savings over BSCS as c_R changes and B=10 (Ex11)

c_R	KCS	EKCS	GKCS
0.5	25.4%	36.7%	32.3%
3.5	20.9%	31.2%	27.2%
6.5	17.3%	26.7%	23.0%
9.5	14.2%	22.8%	19.5%
12.5	11.9%	19.6%	16.5%
15.5	10.1%	16.8%	13.9%
18.5	8.5%	14.3%	11.6%
21.5	6.8%	11.8%	9.2%
24.5	5.2%	9.5%	7.1%
27.5	3.7%	7.5%	5.2%

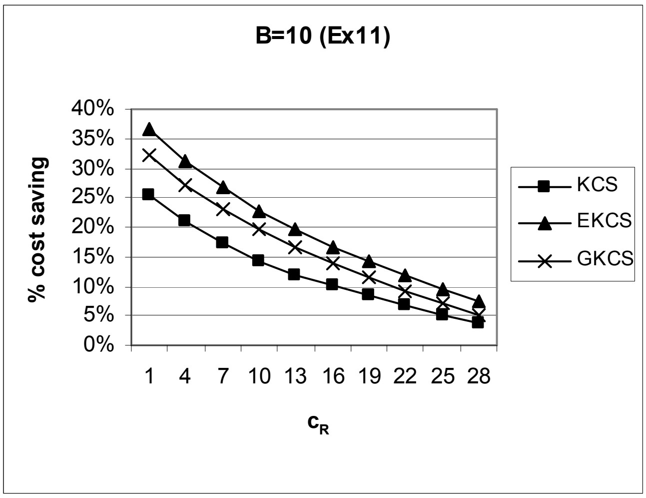

Figure 5.10. Per cent cost savings over BSCS as c_R changes

Table 5.25. Average queue lengths of each policy

KCS (B=10) and Ex11							BSCS (B=10) and Ex11					
c_R	QR	QFP	QBL	WIPR	WIPM	FB	QR	QFP	QBL	WIPR	WIPM	FB
0.5	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
3.5	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
6.5	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
9.5	3.59	1.01	0.61	0.61	0.08	0.38	6.78	1.30	0.30	0.54	0.18	0.22
12.5	5.08	1.19	0.30	0.55	0.11	0.28	6.78	1.30	0.30	0.54	0.18	0.22
15.5	5.08	1.19	0.30	0.55	0.11	0.28	6.78	1.30	0.30	0.54	0.18	0.22
18.5	5.08	1.19	0.30	0.55	0.11	0.28	6.78	1.30	0.30	0.54	0.18	0.22
21.5	5.08	1.19	0.30	0.55	0.11	0.28	7.50	1.36	0.15	0.48	0.21	0.18
24.5	5.08	1.19	0.30	0.55	0.11	0.28	7.50	1.36	0.15	0.48	0.21	0.18
27.5	5.08	1.19	0.30	0.55	0.11	0.28	7.50	1.36	0.15	0.48	0.21	0.18
EKCS (B=10) and Ex11							GKCS (B=10) and Ex11					
c_R	QR	QFP	QBL	WIPR	WIPM	FB	QR	QFP	QBL	WIPR	WIPM	FB
0.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
3.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
6.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
9.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
12.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
15.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
18.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
21.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
24.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09
27.5	3.81	1.93	0.20	0.43	0.08	0.17	3.59	3.54	0.14	0.61	0.08	0.09

The next experiment set presents the results for B=1 and c_R is increasing (Table 5.26). The higher c_R, viz. the less the profitability is, the higher the total average costs become (Figure 5.11). Not only the decline in profitability, but also the small buffer size of cores forces all control mechanisms to remanufacture less as c_R increases (Table 5.27). Each system has a decreasing trend in terms of optimal static routing probability. In each control system there is a rapid decline in optimal static routing probabilities. In other words, there occurs a delay problem in terms of remanufacturing due to that small core buffer size. Thus, the less the profitability is, the more the systems are forced to provide finished products via manufacturing. Since the incoming demand triggers the whole system immediately, BSCS uses less amount of base stocks compared to KCS's kanban sizes. So, KCS results in higher holding inventory costs than BSCS (Figure 5.12). As c_R increases, the profitability decreases. GKCS compensates the decrease in optimal base stock levels by increasing optimal kanban sizes. EKCS, however, does not change kanban sizes, since using dedicated kanbans helps it to compensate the decrease in base stock level. So, its related costs are lower than those of GKCS. Per cent cost savings over BSCS are given in

Table 5.28 and Figure 5.12. KCS's one-parameter structure does not allow it to adjust accordingly. Thus, KCS becomes the worst one in this case. GKCS compensates the problem with its two-parameter structure by increasing one parameter and decreasing the other. EKCS's multi-parameter structure provides the best results among all others (Figure 5.11).

Table 5.26. Cost comparison among control policies as c_R changes and B=1 (Ex5)

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	1600	800	0.01	5.5	30	2.5	9163.29	9276.91	9014.75	9163.29
2	1600	800	0.01	8.5	30	2.5	9943.26	10034.82	9765.40	9943.26
3	1600	800	0.01	11.5	30	2.5	10723.24	10754.79	10485.37	10688.54
4	1600	800	0.01	14.5	30	2.5	11500.16	11474.76	11205.35	11408.51
5	1600	800	0.01	17.5	30	2.5	12220.13	12163.24	11875.20	12128.48
6	1600	800	0.01	20.5	30	2.5	12898.67	12823.21	12535.17	12790.02
7	1600	800	0.01	23.5	30	2.5	13549.65	13483.18	13195.14	13449.99
8	1600	800	0.01	26.5	30	2.5	14149.61	14038.66	13782.89	13990.63
9	1600	800	0.01	29.5	30	2.5	14716.80	14540.56	14310.10	14454.89
10	1600	800	0.01	32.5	30	2.5	15222.80	14983.98	14769.55	14874.85

Table 5.27. Optimal parameters as c_R changes and B=1 (Ex5)

	KCS			BSCS			EKCS					GKCS			
c_R	Z^*	r^*	K^*	Z^*	r^*	S^*	Z^*	r^*	S^*	K_R^*	K_M^*	Z^*	r^*	S^*	K^*
5.5	9163.3	0.65	3	9276.9	0.65	3	9014.7	0.65	3	2	1	9163.3	0.65	3	3
8.5	9943.3	0.65	3	10034.8	0.60	2	9765.4	0.60	2	1	1	9943.3	0.65	3	3
11.5	10723.2	0.65	3	10754.8	0.60	2	10485.4	0.60	2	1	1	10688.5	0.60	2	3
14.5	11500.2	0.60	3	11474.8	0.60	2	11205.3	0.60	2	1	1	11408.5	0.60	2	3
17.5	12220.1	0.60	3	12163.2	0.55	2	11875.2	0.55	2	1	1	12128.5	0.60	2	3
20.5	12898.7	0.55	2	12823.2	0.55	2	12535.2	0.55	2	1	1	12790.0	0.55	2	3
23.5	13549.6	0.50	2	13483.2	0.55	2	13195.1	0.55	2	1	1	13450.0	0.55	2	3
26.5	14149.6	0.50	2	14038.7	0.45	1	13782.9	0.45	1	1	1	13990.6	0.40	1	8
29.5	14716.8	0.45	2	14540.6	0.40	1	14310.1	0.40	1	1	1	14454.9	0.35	1	10
32.5	15222.8	0.40	2	14984.0	0.35	1	14769.5	0.35	1	1	1	14874.9	0.35	1	10

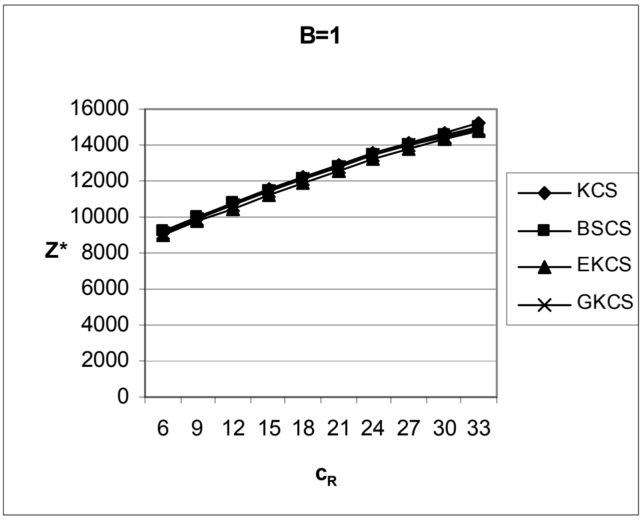

Figure 5.11. Optimal average total cost comparison as c_R changes (Ex5)

Table 5.28. Per cent cost savings over BSCS as c_R changes and B=1 (Ex5)

c_R	KCS	EKCS	GKCS
5.5	1.2%	2.8%	1.2%
8.5	0.9%	2.7%	0.9%
11.5	0.3%	2.5%	0.6%
14.5	-0.2%	2.3%	0.6%
17.5	-0.5%	2.4%	0.3%
20.5	-0.6%	2.2%	0.3%
23.5	-0.5%	2.1%	0.2%
26.5	-0.8%	1.8%	0.3%
29.5	-1.2%	1.6%	0.6%
32.5	-1.6%	1.4%	0.7%

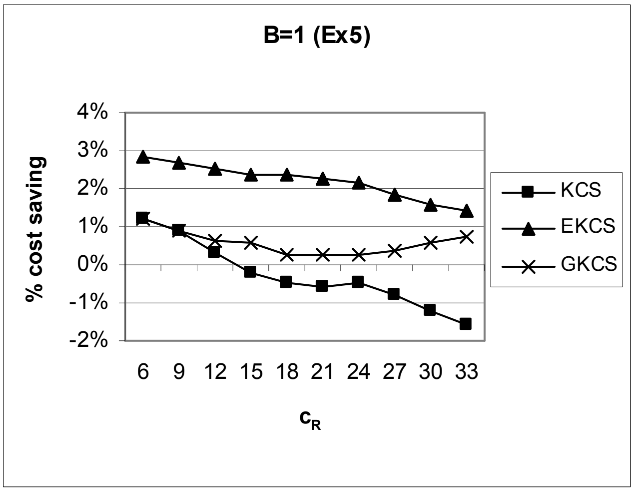

Figure 5.12. Per cent cost savings over BSCS as c_R changes

5.4. Change in Marginal Cost for Manufacturing

Just like the unit remanufacturing costs, the unit production costs for manufacturing is determined using the "profitability" condition. We vary the profitability in an increasing way, and thereby determine c_M by holding c_D and c_R at constant values.

The buffer size for cores is set equal to ten, for the experiment set that is interpreted (Table 5.29). Optimal parameter values for this case, where all the parameter values except the unit production costs for manufacturing are constant, are given in Table 5.30.

The higher the unit production costs for manufacturing are getting, the more profitable the remanufacturing mechanism becomes, and so, the higher per cent cost savings we obtain (Figure 5.14). As a result of the large capacity of the buffer for cores, demands are mainly satisfied through remanufactured products. Here, we observe an increase in optimal static routing probabilities of BSCS and KCS. The reason why KCS performs better than BSCS is, that it can hold r^* at higher values in comparison to BSCS. On the other hand, the common characteristic of GKCS and EKCS is, that the optimal static routing probabilities remain the same with an increase in c_M , they compensate the situation by holding S at high values (Table 5.30). Since GKCS has higher S values than EKCS, it results in higher inventory holding costs in comparison to EKCS. For lower values of c_M, we observe that the control policies demonstrate almost the same behavior, whereas, for higher values of c_M, BSCS performs poorly (Figure 5.13). Here, it is reasonable choosing EKCS which controls the production, in a very qualified way, considering its structure (Table 5.31 and Figure 5.14).

Table 5.29. Cost comparison among control policies as c_M changes and B=10 (Ex11)

Ex	*b*	*h*	*α*	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	3200	400	0.01	5	7.5	-2.5	5830.88	6406.32	5254.55	5606.92
2	3200	400	0.01	5	8.3	-2.5	5894.93	6518.39	5302.60	5654.97
3	3200	400	0.01	5	9.4	-2.5	5983.01	6672.50	5368.67	5721.05
4	3200	400	0.01	5	10.7	-2.5	6087.10	6854.62	5446.75	5799.14
5	3200	400	0.01	5	12.5	-2.5	6231.22	7106.80	5554.86	5907.26
6	3200	400	0.01	5	15.0	-2.5	6431.40	7457.03	5705.02	6057.44
7	3200	400	0.01	5	18.8	-2.5	6735.66	7938.83	5933.25	6285.70
8	3200	400	0.01	5	25.0	-2.5	7219.53	8683.38	6305.64	6658.13
9	3200	400	0.01	5	37.5	-2.5	7970.19	10184.48	7056.42	7408.99
10	3200	400	0.01	5	75.0	-2.5	10222.17	14142.51	9308.76	9661.58

Table 5.30. Optimal parameters as c_M changes and B=10 (Ex11)

	KCS			BSCS			EKCS					GKCS			
c_M	*Z**	*r**	*K**	*Z**	*r**	*S**	*Z**	*r**	*S**	K_R*	K_M*	*Z**	*r**	*S**	*K**
7.5	5830.9	0.80	2	6406.3	0.65	2	5254.5	0.85	3	1	2	5606.9	0.85	4	2
8.3	5894.9	0.80	2	6518.4	0.65	2	5302.6	0.85	3	1	2	5655.0	0.85	4	2
9.4	5983.0	0.80	2	6672.5	0.65	2	5368.7	0.85	3	1	2	5721.0	0.85	4	2
10.7	6087.1	0.80	2	6854.6	0.65	2	5446.7	0.85	3	1	2	5799.1	0.85	4	2
12.5	6231.2	0.80	2	7106.8	0.65	2	5554.9	0.85	3	1	2	5907.3	0.85	4	2
15.0	6431.4	0.80	2	7457.0	0.65	2	5705.0	0.85	3	1	2	6057.4	0.85	4	2
18.8	6735.7	0.80	2	7938.8	0.70	2	5933.3	0.85	3	1	2	6285.7	0.85	4	2
25.0	7219.5	0.85	2	8683.4	0.70	2	6305.6	0.85	3	1	2	6658.1	0.85	4	2
37.5	7970.2	0.85	2	10184.5	0.70	2	7056.4	0.85	3	1	2	7409.0	0.85	4	2
75.0	10222.2	0.85	2	14142.5	0.75	4	9308.8	0.85	3	1	2	9661.6	0.85	4	2

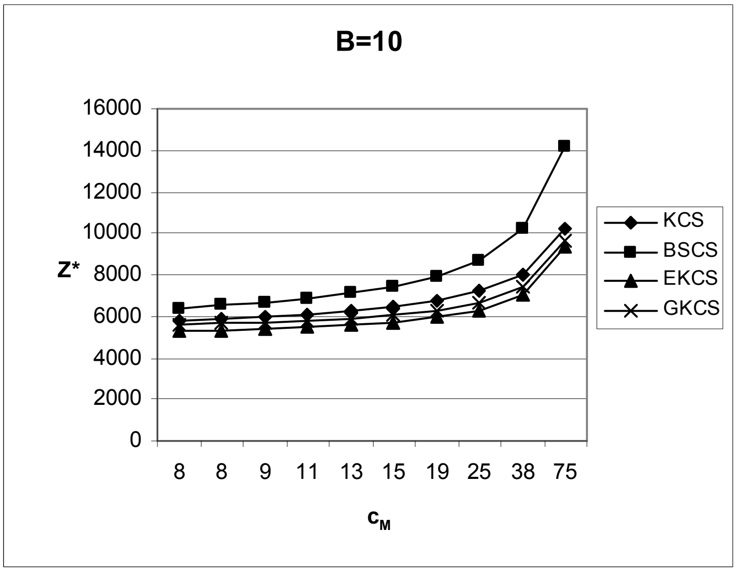

Figure 5.13. Optimal average total cost comparison as c_M changes (Ex11)

Table 5.31. Per cent cost savings over BSCS as c_M changes and B=10 (Ex11)

c_M	KCS	EKCS	GKCS
7.5	9.0%	18.0%	12.5%
8.3	9.6%	18.7%	13.2%
9.4	10.3%	19.5%	14.3%
10.7	11.2%	20.5%	15.4%
12.5	12.3%	21.8%	16.9%
15.0	13.8%	23.5%	18.8%
18.8	15.2%	25.3%	20.8%
25.0	16.9%	27.4%	23.3%
37.5	21.7%	30.7%	27.3%
75.0	27.7%	34.2%	31.7%

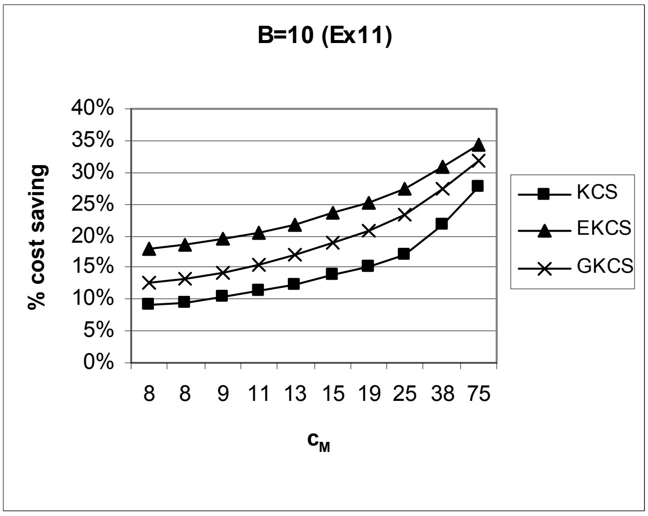

Figure 5.14. Per cent cost savings over BSCS as c_M changes

The buffer size for cores is set equal to one for the next experiment set (Table 5.32). Optimal parameter values for this case, where all the parameter values except the unit production costs for manufacturing are constant, are given in Table 5.33.

When the buffer size of cores is small, all four policies have increasing static routing probabilities as c_M increases (Table 5.33). BSCS uses more base stock for large values of c_M. In the KCS, optimal kanban size grows parallel to the increase of c_M. So, the higher the profitability is, the higher the per cent cost savings of KCS get. Its single-parameter structure just performs better than BSCS for the highest c_M values. In GKCS, as c_M increases, optimal kanban size decreases. The backlog queue is decreased in comparison with that of BSCS (Table 5.35), however, it cannot overcome the performance of EKCS. GKCS in convex, because *S* is increasing while *K* decreasing (Figure 5.16). On the other hand, the advantage of EKCS is, that it increases optimal remanufacturing kanban size for large values of $c_{M,}$ where the profitability caused by remanufacturing will be high. It realizes the satisfaction of demands from remanufactured products, by dedicating more kanbans to remanufacturing. Hence, it gives a rather stable graph. Since Figure 5.15 does not provide clear vision, we depict per cent cost savings over BSCS (Table 5.34 and Figure 5.16). Observing the cost comparison figure (Figure 5.15), it is reasonable using a less complicated system such as KCS or BSCS, since all of the policies reflect the same behavior in average cost change, as c_M increases and when B gets small values. However, for small values of c_M, KCS should be avoided (Figure 5.16).

Table 5.32. Cost comparison among control policies as c_M changes and B=1 (Ex5)

Ex	*b*	*h*	*α*	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	1600	800	0.01	10	7.5	2.5	6222.54	5983.87	5769.44	5874.75
2	1600	800	0.01	10	8.3	2.5	6414.56	6191.89	5977.45	6082.76
3	1600	800	0.01	10	9.4	2.5	6674.57	6476.46	6245.99	6368.78
4	1600	800	0.01	10	10.7	2.5	6960.60	6788.48	6558.01	6706.80
5	1600	800	0.01	10	12.5	2.5	7349.43	7218.58	6962.81	7150.54
6	1600	800	0.01	10	15.0	2.5	7849.48	7768.62	7512.85	7723.79
7	1600	800	0.01	10	18.8	2.5	8572.56	8497.09	8209.06	8463.91
8	1600	800	0.01	10	25.0	2.5	9620.08	9594.72	9325.17	9528.46
9	1600	800	0.01	10	37.5	2.5	11383.42	11497.05	11234.88	11383.42
10	1600	800	0.01	10	75.0	2.5	16042.19	16233.80	15893.76	15956.31

Table 5.33. Optimal parameters as c_M changes and B=1 (Ex5)

	KCS			BSCS			EKCS					GKCS			
c_M	Z^*	r^*	K^*	Z^*	r^*	S^*	Z^*	r^*	S^*	K_R^*	K_M^*	Z^*	r^*	S^*	K^*
7.5	6222.5	0.40	2	5983.9	0.35	1	5769.4	0.35	1	1	1	5874.7	0.35	1	10
8.3	6414.6	0.40	2	6191.9	0.35	1	5977.5	0.35	1	1	1	6082.8	0.35	1	10
9.4	6674.6	0.45	2	6476.5	0.40	1	6246.0	0.40	1	1	1	6368.8	0.35	1	10
10.7	6960.6	0.45	2	6788.5	0.40	1	6558.0	0.40	1	1	1	6706.8	0.35	1	10
12.5	7349.4	0.50	2	7218.6	0.45	1	6962.8	0.45	1	1	1	7150.5	0.40	1	8
15.0	7849.5	0.50	2	7768.6	0.45	1	7512.8	0.45	1	1	1	7723.8	0.45	1	5
18.8	8572.6	0.55	2	8497.1	0.55	2	8209.1	0.55	2	1	1	8463.9	0.55	2	3
25.0	9620.1	0.60	3	9594.7	0.60	2	9325.2	0.55	2	1	1	9528.5	0.60	2	3
37.5	11383.4	0.65	3	11497.0	0.65	3	11234.9	0.65	3	2	1	11383.4	0.65	3	3
75.0	16042.2	0.75	5	16233.8	0.70	3	15893.8	0.75	5	3	2	15956.3	0.75	5	3

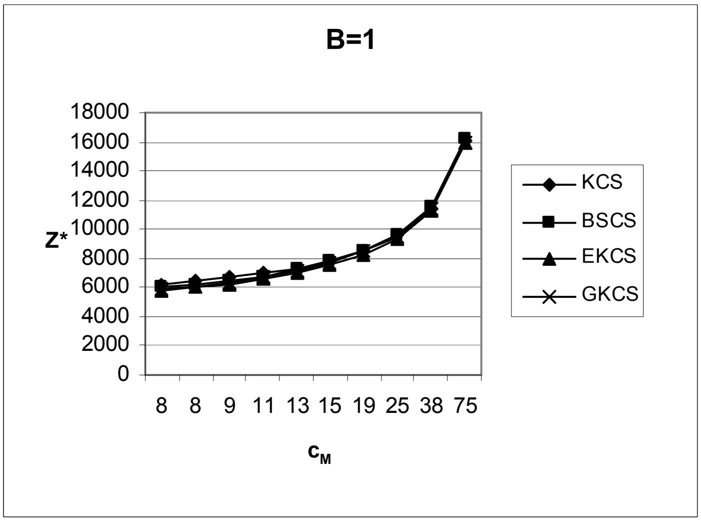

Figure 5.15. Optimal average total cost comparison as c_M changes (Ex5)

Table 5.34. Per cent cost savings over BSCS as c_M changes and B=1 (Ex5)

c_M	KCS	EKCS	GKCS
7.5	-4.0%	3.6%	1.8%
8.3	-3.6%	3.5%	1.8%
9.4	-3.1%	3.6%	1.7%
10.7	-2.5%	3.4%	1.2%
12.5	-1.8%	3.5%	0.9%
15.0	-1.0%	3.3%	0.6%
18.8	-0.9%	3.4%	0.4%
25.0	-0.3%	2.8%	0.7%
37.5	1.0%	2.3%	1.0%
75.0	1.2%	2.1%	1.7%

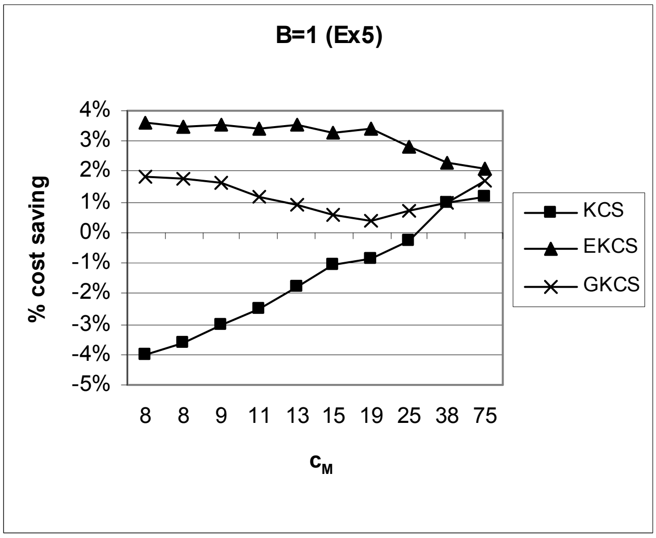

Figure 5.16. Per cent cost savings over BSCS as c_M changes

Table 5.35. Average queue lengths of each policy

KCS (B=1) and Ex5							BSCS (B=1) and Ex5					
c_M	QR	QFP	QBL	WIPR	WIPM	FB	QR	QFP	QBL	WIPR	WIPM	FB
7.5	0.56	1.15	0.29	0.24	0.38	0.29	0.61	0.47	0.41	0.21	0.48	0.53
8.3	0.56	1.15	0.29	0.24	0.38	0.29	0.61	0.47	0.41	0.21	0.48	0.53
9.4	0.50	1.08	0.38	0.27	0.34	0.32	0.56	0.45	0.48	0.25	0.43	0.55
10.7	0.50	1.08	0.38	0.27	0.34	0.32	0.56	0.45	0.48	0.25	0.43	0.55
12.5	0.44	1.01	0.50	0.31	0.31	0.36	0.50	0.42	0.59	0.29	0.38	0.58
15	0.44	1.01	0.50	0.31	0.31	0.36	0.50	0.42	0.59	0.29	0.38	0.58
18.8	0.39	0.92	0.69	0.34	0.27	0.42	0.39	0.96	0.59	0.38	0.29	0.39
25	0.33	1.56	0.55	0.41	0.25	0.30	0.33	0.86	0.87	0.43	0.25	0.45
37.5	0.28	1.37	0.89	0.46	0.21	0.37	0.28	1.38	0.95	0.48	0.21	0.37
75	0.17	2.05	1.89	0.60	0.14	0.39	0.22	1.16	1.60	0.54	0.18	0.46
EKCS (B=1) and Ex5							GKCS (B=1) and Ex5					
c_M	QR	QFP	QBL	WIPR	WIPM	FB	QR	QFP	QBL	WIPR	WIPM	FB
7.5	0.61	0.48	0.37	0.18	0.33	0.52	0.61	0.47	0.35	0.21	0.48	0.53
8.3	0.61	0.48	0.37	0.18	0.33	0.52	0.61	0.47	0.35	0.21	0.48	0.53
9.4	0.56	0.45	0.43	0.20	0.30	0.55	0.61	0.47	0.35	0.21	0.48	0.53
10.7	0.56	0.45	0.43	0.20	0.30	0.55	0.61	0.47	0.35	0.21	0.48	0.53
12.5	0.50	0.42	0.52	0.23	0.28	0.58	0.56	0.45	0.44	0.25	0.43	0.55
15	0.50	0.42	0.52	0.23	0.28	0.58	0.50	0.42	0.56	0.29	0.38	0.58
18.8	0.39	0.96	0.50	0.28	0.23	0.39	0.39	0.95	0.58	0.37	0.28	0.39
25	0.39	0.96	0.50	0.28	0.23	0.39	0.33	0.85	0.84	0.41	0.25	0.45
37.5	0.28	1.37	0.83	0.43	0.18	0.36	0.28	1.37	0.89	0.46	0.21	0.37
75	0.17	2.04	1.82	0.57	0.14	0.39	0.17	2.73	1.52	0.59	0.14	0.32

5.5. Change in Unit Disposal Cost

Unit disposal cost is the third cost component determined using the "profitability" condition. We vary the profitability in an increasing way, and determine c_D while holding c_R and c_M constant.

The buffer size for cores is set equal to ten, for experiment 11 (Table 5.36). Optimal parameter values for this case, where all the parameter values except the unit disposal cost are constant, are to be seen in Table 5.37.

In the BSCS and in the KCS, the higher the profitability is, the more the systems are forced to feed the output buffer with remanufactured products (Table 5.37). When the buffer size is high, an increase in unit disposal cost does not affect optimal base stock level in the BSCS and optimal kanban size in the KCS. However, there is an increase in optimal static routing probabilities, because remanufacturing becomes more profitable as c_D increases. Since KCS has higher routing probabilities than BSCS, its related costs are lower than those of BSCS. GKCS and EKCS, however, have a very slow increasing trend in terms of cost (Figure 5.17). Optimal base stock levels and kanban sizes remain the same with an increase in c_D. Furthermore, the disposal rate takes a very small value as a result of the routing probability. Thus, they result in lower disposal costs in comparison with KCS. In the EKCS, an increase from negative values of unit disposal cost to positive values does not grow total average cost significantly. When B takes large values, BSCS should be avoided (Figure 5.18), regardless of the fact that the disposal cost per one unit is low or high (Table 5.38).

Table 5.36. Cost comparison among control policies as c_D changes and B=10 (Ex11)

Ex	*b*	*h*	*α*	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	3200	400	0.01	5	30	-23.5	6792.44	7458.47	6185.95	6538.47
2	3200	400	0.01	5	30	-20.5	6912.44	7758.47	6245.95	6598.47
3	3200	400	0.01	5	30	-17.5	7032.44	8058.47	6305.95	6658.47
4	3200	400	0.01	5	30	-14.5	7152.44	8323.82	6365.95	6718.47
5	3200	400	0.01	5	30	-11.5	7272.44	8563.82	6425.95	6778.47
6	3200	400	0.01	5	30	-8.5	7392.44	8803.82	6485.95	6838.47
7	3200	400	0.01	5	30	-5.5	7459.79	9043.82	6545.95	6898.47
8	3200	400	0.01	5	30	-2.5	7519.79	9283.82	6605.95	6958.47
9	3200	400	0.01	5	30	0.5	7579.79	9523.82	6665.95	7018.47
10	3200	400	0.01	5	30	3.5	7639.79	9763.82	6725.95	7078.47

Table 5.37. Optimal parameters as c_D changes and B=10 (Ex11)

	KCS			BSCS			EKCS					GKCS			
c_D	*Z**	*r**	*K**	*Z**	*r**	*S**	*Z**	*r**	*S**	K_R*	K_M*	*Z**	*r**	*S**	*K**
-23.5	6792.4	0.80	2	7458.5	0.65	2	6186.0	0.85	3	1	2	6538.5	0.85	4	2
-20.5	6912.4	0.80	2	7758.5	0.65	2	6246.0	0.85	3	1	2	6598.5	0.85	4	2
-17.5	7032.4	0.80	2	8058.5	0.65	2	6306.0	0.85	3	1	2	6658.5	0.85	4	2
-14.5	7152.4	0.80	2	8323.8	0.70	2	6366.0	0.85	3	1	2	6718.5	0.85	4	2
-11.5	7272.4	0.80	2	8563.8	0.70	2	6426.0	0.85	3	1	2	6778.5	0.85	4	2
-8.5	7392.4	0.80	2	8803.8	0.70	2	6486.0	0.85	3	1	2	6838.5	0.85	4	2
-5.5	7459.8	0.85	2	9043.8	0.70	2	6546.0	0.85	3	1	2	6898.5	0.85	4	2
-2.5	7519.8	0.85	2	9283.8	0.70	2	6606.0	0.85	3	1	2	6958.5	0.85	4	2
0.5	7579.8	0.85	2	9523.8	0.70	2	6666.0	0.85	3	1	2	7018.5	0.85	4	2
3.5	7639.8	0.85	2	9763.8	0.70	2	6726.0	0.85	3	1	2	7078.5	0.85	4	2

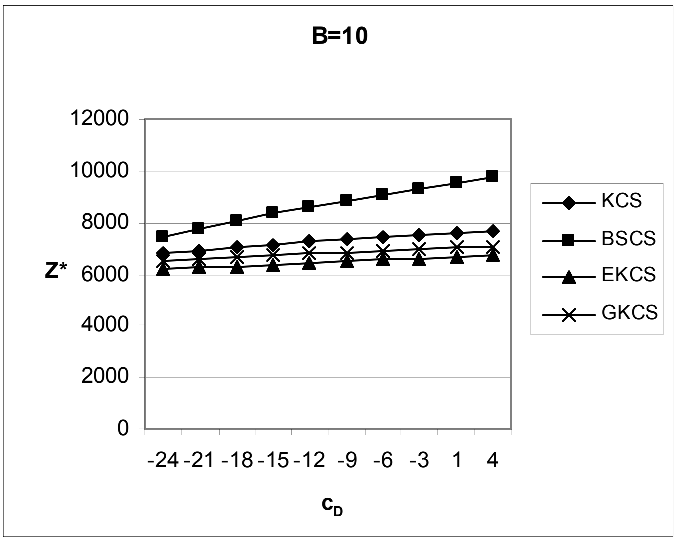

Figure 5.17. Optimal average total cost comparison as c_D changes (Ex11)

Table 5.38. Per cent cost savings over BSCS as c_D changes and B=10 (Ex11)

c_D	KCS	EKCS	GKCS
-23.5	8.9%	17.1%	12.3%
-20.5	10.9%	19.5%	15.0%
-17.5	12.7%	21.7%	17.4%
-14.5	14.1%	23.5%	19.3%
-11.5	15.1%	25.0%	20.8%
-8.5	16.0%	26.3%	22.3%
-5.5	17.5%	27.6%	23.7%
-2.5	19.0%	28.8%	25.0%
0.5	20.4%	30.0%	26.3%
3.5	21.8%	31.1%	27.5%

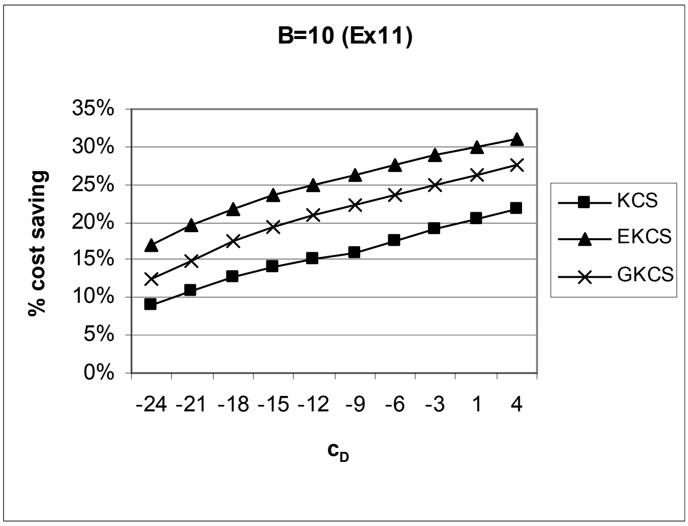

Figure 5.18. Per cent cost savings over BSCS as c_D changes

When the buffer size of cores is small, static routing probabilities of all control policies increase, as the unit disposal cost increases (Table 5.40). The related cost comparison is given in Table 5.39. Small buffer size of cores causes remanufacturing process to delay. As the unit disposal cost increases, the profitability of remanufacturing in comparison to manufacturing also increases. This is the reason for an increase of the routing probability of demands to the remanufacturing process, although it is a bottleneck. An increase in r^*, not only decreases the disposal rate, but also the revenue to be obtained for negative values of c_D. Figure 5.19 gives not a clear picture, so per cent cost savings are depicted in Figure 5.20. For small values of c_D, KCS seems to be the worst policy, whereas EKCS is the best one, obviously (Table 5.41 and Figure 5.20). Nevertheless, when B has low capacity, any change in c_D does not have a major influence on the average cost of any policy (Figure 5.19).

Table 5.39. Cost comparison among control policies as c_D changes and B=1 (Ex5)

Ex	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
1	1600	800	0.01	10	30	-18.5	7423.07	7214.27	6990.30	7105.15
2	1600	800	0.01	10	30	-15.5	7967.02	7820.80	7573.06	7750.83
3	1600	800	0.01	10	30	-12.5	8449.79	8368.85	8113.06	8324.03
4	1600	800	0.01	10	30	-9.5	8908.78	8833.32	8545.26	8800.13
5	1600	800	0.01	10	30	-6.5	9328.78	9253.32	8965.26	9220.13
6	1600	800	0.01	10	30	-3.5	9700.21	9673.32	9385.26	9608.56
7	1600	800	0.01	10	30	-0.5	10033.25	10034.81	9765.39	9968.56
8	1600	800	0.01	10	30	2.5	10333.25	10394.81	10125.39	10328.56
9	1600	800	0.01	10	30	5.5	10633.25	10746.87	10484.71	10633.25
10	1600	800	0.01	10	30	8.5	10933.25	11046.87	10784.71	10933.25

Table 5.40. Optimal parameters as c_D changes and B=1 (Ex5)

	KCS			BSCS			EKCS					GKCS			
c_D	Z^*	r^*	K^*	Z^*	r^*	S^*	Z^*	r^*	S^*	K_R^*	K_M^*	Z^*	r^*	S^*	K^*
-18.5	7423.1	0.40	2	7214.3	0.35	1	6990.3	0.40	1	1	1	7105.1	0.35	1	10
-15.5	7967.0	0.45	2	7820.8	0.40	1	7573.1	0.45	1	1	1	7750.8	0.40	1	8
-12.5	8449.8	0.50	2	8368.9	0.45	1	8113.1	0.45	1	1	1	8324.0	0.45	1	5
-9.5	8908.8	0.55	2	8833.3	0.55	2	8545.3	0.55	2	1	1	8800.1	0.55	2	3
-6.5	9328.8	0.55	2	9253.3	0.55	2	8965.3	0.55	2	1	1	9220.1	0.55	2	3
-3.5	9700.2	0.60	3	9673.3	0.55	2	9385.3	0.55	2	1	1	9608.6	0.60	2	3
-0.5	10033.3	0.65	3	10034.8	0.60	2	9765.4	0.60	2	1	1	9968.6	0.60	2	3
2.5	10333.3	0.65	3	10394.8	0.60	2	10125.4	0.60	2	1	1	10328.6	0.60	2	3
5.5	10633.3	0.65	3	10746.9	0.65	3	10484.7	0.65	3	2	1	10633.3	0.65	3	3
8.5	10933.3	0.65	3	11046.9	0.65	3	10784.7	0.65	3	2	1	10933.3	0.65	3	3

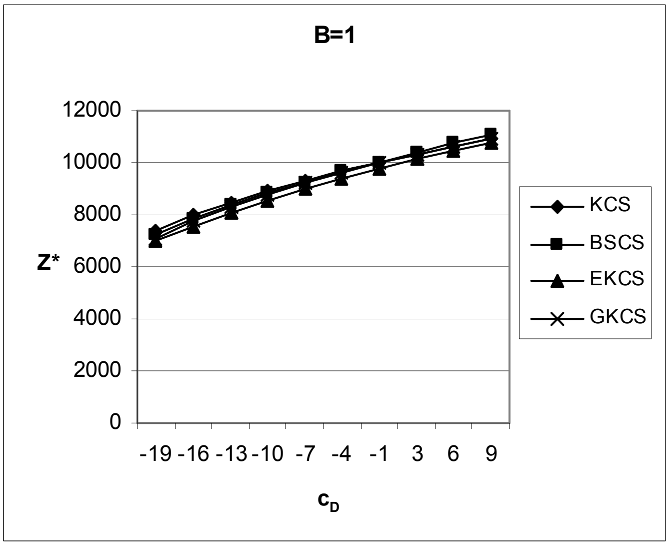

Figure 5.19. Optimal average total cost comparison as c_D changes (Ex5)

Table 5.41. Per cent cost savings over BSCS as c_D changes and B=1 (Ex5)

c_D	KCS	EKCS	GKCS
-18.5	-2.9%	3.1%	1.5%
-15.5	-1.9%	3.2%	0.9%
-12.5	-1.0%	3.1%	0.5%
-9.5	-0.9%	3.3%	0.4%
-6.5	-0.8%	3.1%	0.4%
-3.5	-0.3%	3.0%	0.7%
-0.5	0.0%	2.7%	0.7%
2.5	0.6%	2.6%	0.6%
5.5	1.1%	2.4%	1.1%
8.5	1.0%	2.4%	1.0%

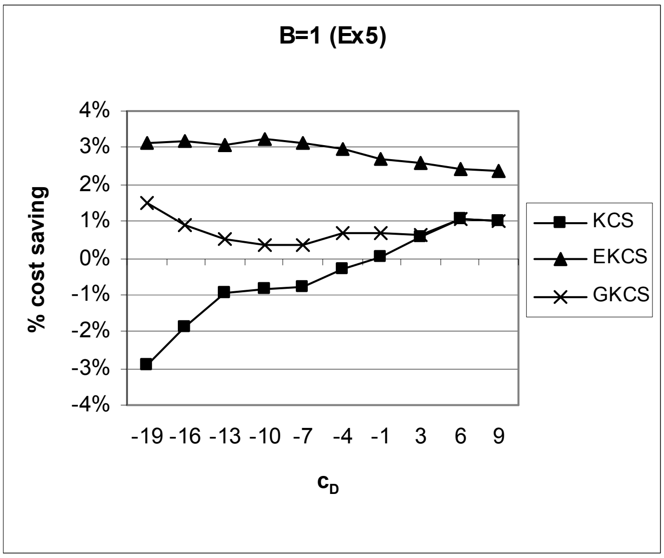

Figure 5.20. Per cent cost savings over BSCS as c_D changes

5.6. Change in the Return Arrival Rate

The buffer size of cores B is set equal to five, whereas the return arrival rate varies from 360 down to 200. The service rates of both processes and the demand arrival rate are as mentioned in section five.

Making use of these tables, we do the sensitivity analysis concerning the return arrival rate γ. Since all the experiment results seem to have the same trends in terms of total average costs, we only pick out one of these 18 experiment sets, and make interpretations on this.

We pick out the fifth experiment, where b = 1600, h = 800, α = 0.01, c_R = 10, c_M = 30 and c_D = 2.5 (Table 5.42), due to the reason explained already in section five. Since the 11th experiment gives similar results, we only use the fifth experiment here. The buffer size of cores B is set equal to five, as mentioned previously. As the return arrival rate decreases, the total average cost of each system increases, because the lower the return arrival rate is, the more the systems are forced providing finished products from manufacturing which is expensive. As the return arrival rate γ decreases, the optimal static routing probabilities also decrease. Hence, (1-r) increases to satisfy the demand as expected (Table 5.43). The optimal base stock level of BSCS and the optimal kanban sizes of KCS do not change. We observe, that when *S*=*K* in the GKCS, it has the same total average costs as KCS. That is why their related graphs overlap (Figure 5.21). The advantage of EKCS, here, is, that it decreases base stock level to provide compensation. EKCS is the best control policy since it has the most parameters among all. However, BSCS is the worst policy (Table 5.44 and Figure 5.22).

Table 5.42. Cost comparison among control policies as γ changes (Ex5)

γ	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
360	1600	800	0.01	10	30	2.5	9667.73	11339.94	9149.51	9670.00
320	1600	800	0.01	10	30	2.5	10297.42	11842.78	9468.22	10297.42
280	1600	800	0.01	10	30	2.5	10897.67	12487.19	10002.06	10897.67
240	1600	800	0.01	10	30	2.5	11483.50	12975.02	10693.82	11483.50
200	1600	800	0.01	10	30	2.5	12190.23	13669.33	11428.00	12190.23

Table 5.43. Optimal parameters as γ changes (Ex5)

	KCS			BSCS			EKCS					GKCS			
γ	Z^*	r^*	K^*	Z^*	r^*	S^*	Z^*	r^*	S^*	K_R^*	K_M^*	Z^*	r^*	S^*	K^*
360	9667.73	0.80	2	11339.94	0.70	1	9149.51	0.85	3	1	2	9667.73	0.80	2	2
320	10297.42	0.70	2	11842.78	0.60	1	9468.22	0.75	2	1	1	10297.42	0.70	2	2
280	10897.67	0.65	2	12487.19	0.50	1	10002.06	0.65	2	1	1	10897.67	0.65	2	2
240	11483.50	0.55	2	12975.02	0.45	1	10693.82	0.55	1	1	1	11483.50	0.55	2	2
200	12190.23	0.45	2	13669.33	0.35	1	11428.00	0.45	1	1	1	12190.23	0.45	2	2

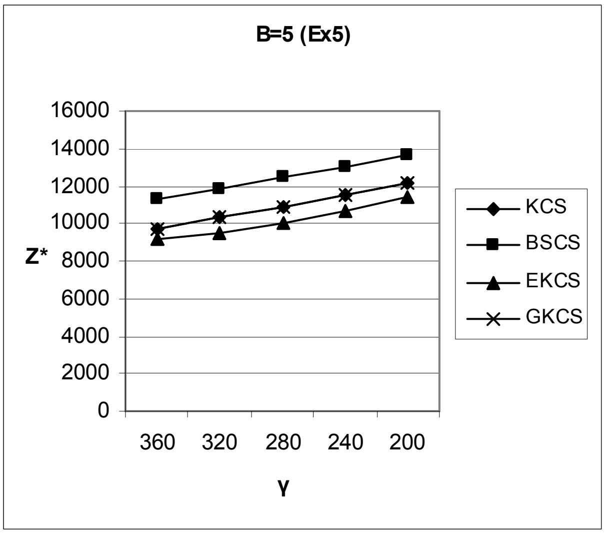

Figure 5.21. Optimal average total cost comparison as γ changes.

Table 5.44. Per cent cost savings over BSCS as γ changes and B=5 (Ex5)

γ	KCS	EKCS	GKCS
360	14.7%	19.3%	14.7%
320	13.0%	20.1%	13.0%
280	12.7%	19.9%	12.7%
240	11.5%	17.6%	11.5%
200	10.8%	16.4%	10.8%

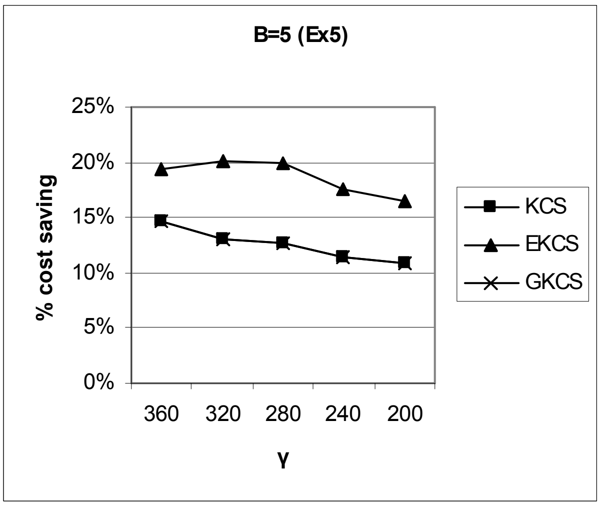

Figure 5.22. Per cent cost savings over BSCS as γ changes

5.7. Change in Demand Arrival Rate

The buffer size of cores B is set equal to five, whereas the demand arrival rate varies from 400 down to 300. The return arrival rate is set to be 360. The service rates of both processes are as mentioned in section five.

Next, we choose the fifth experiment due to the reason explained already in section five (Table 5.45). The optimal static routing probabilities increase in all policies as expected (Table 5.46). Comments on cost components are given according to Table 5.48 which gives average queue lengths. In the KCS, as the demand arrival rate decreases, the holding cost associated with both non-serviceable items and finished products decreases as long as $\gamma \leq \lambda_D$, and then increases for $\gamma > \lambda_D$. Meanwhile, the backorder cost increases as long as $\gamma \leq \lambda_D$, and then decreases for $\gamma > \lambda_D$. Here, the disposal cost decreases as long as $\gamma \leq \lambda_D$, and then increases for $\gamma > \lambda_D$. Remanufacturing cost increases as long as $\gamma \leq \lambda_D$, and then decreases for $\gamma > \lambda_D$. Manufacturing cost decreases as λ_D decreases, and these changes lead to a total decrease in overall cost. Table 5.47 and Figure 5.24 show us, that the slower the demands arrive at the system compared to the return flow, the more similar behavior KCS, GKCS and EKCS have. The per cent cost savings increase as long as $\gamma \leq \lambda_D$, and decrease for $\gamma > \lambda_D$. BSCS should be avoided for every arrival rate value. The increase in the GKCS happens quicker than in the EKCS. GKCS seems to be at least as good as KCS. For $\gamma \leq \lambda_D$, one should one should prefer EKCS, whereas for $\gamma > \lambda_D$ it is reasonable to use a less complicated system such as KCS.

Table 5.45. Cost comparison among control policies as λ_D changes (Ex5)

λ_D	b	h	α	c_R	c_M	c_D	KCS	BSCS	EKCS	GKCS
400	1600	800	0.01	10	30	2.5	9667.73	11339.94	9149.51	9667.73
380	1600	800	0.01	10	30	2.5	8999.12	10692.27	8515.89	8961.75
360	1600	800	0.01	10	30	2.5	8327.64	10021.25	7872.61	8131.34
340	1600	800	0.01	10	30	2.5	7482.55	9373.66	7270.03	7373.84
320	1600	800	0.01	10	30	2.5	7113.76	8736.13	7031.54	7113.76
300	1600	800	0.01	10	30	2.5	6974.15	8108.55	6923.28	6974.15

Table 5.46. Optimal parameters as λ_D changes (Ex5)

	KCS			BSCS			EKCS					GKCS			
λ_D	Z^*	r^*	K^*	Z^*	r^*	S^*	Z^*	r^*	S^*	K_R^*	K_M^*	Z^*	r^*	S^*	K^*
400	9667.73	0.80	2	11339.94	0.70	1	9149.51	0.85	3	1	2	9667.73	0.80	2	2
380	8999.12	0.85	2	10692.27	0.70	1	8515.89	0.85	2	1	1	8961.75	0.85	2	1
360	8327.64	0.90	1	10021.25	0.75	1	7872.61	0.90	2	1	1	8131.34	0.90	2	1
340	7482.55	0.95	1	9373.66	0.80	1	7270.03	0.95	2	1	1	7373.84	0.95	2	1
320	7113.76	0.95	1	8736.13	0.85	1	7031.54	0.95	1	1	1	7113.76	0.95	1	1
300	6974.15	0.95	1	8108.55	0.90	1	6923.28	0.95	1	1	1	6974.15	0.95	1	1

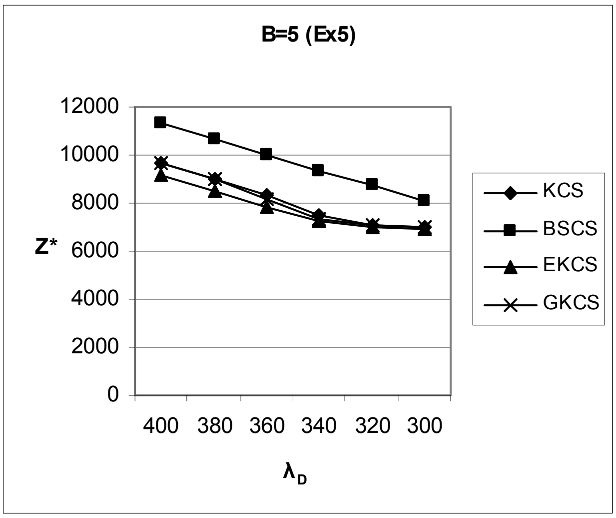

Figure 5.23. Optimal average total cost comparison as λ_D changes.

Table 5.47. Per cent cost savings over BSCS as λ_D changes (Ex5)

λ_D	KCS	EKCS	GKCS
400	14.7%	19.3%	14.7%
380	15.8%	20.4%	16.2%
360	16.9%	21.4%	18.9%
340	20.2%	22.4%	21.3%
320	18.6%	19.5%	18.6%
300	14.0%	14.6%	14.0%

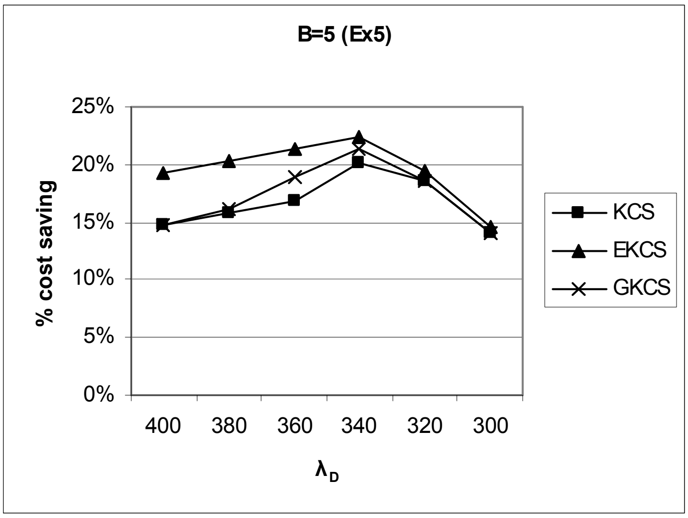

Figure 5.24. Per cent cost savings over BSCS as λ_D changes

Table 5.48. Average queue lengths of each policy

	KCS (B=5) and Ex5						BSCS (B=5) and Ex5					
λ_D	QR	QFP	QBL	WIPR	WIPM	FB	QR	QFP	QBL	WIPR	WIPM	FB
400	1.74	0.91	0.82	0.56	0.11	0.43	2.50	0.43	1.14	0.54	0.18	0.57
380	1.66	0.90	0.88	0.57	0.08	0.44	2.79	0.48	0.77	0.50	0.17	0.52
360	1.76	0.36	1.17	0.40	0.04	0.64	2.71	0.48	0.83	0.51	0.13	0.52
340	1.78	0.38	0.98	0.40	0.02	0.62	2.67	0.49	0.86	0.52	0.09	0.51
320	2.25	0.47	0.60	0.38	0.02	0.53	2.67	0.50	0.85	0.52	0.06	0.50
300	2.64	0.53	0.41	0.36	0.02	0.47	2.71	0.52	0.78	0.51	0.04	0.48
	EKCS (B=5) and Ex5						GKCS (B=5) and Ex5					
λ_D	QR	QFP	QBL	WIPR	WIPM	FB	QR	QFP	QBL	WIPR	WIPM	FB
400	1.26	1.40	0.85	0.43	0.08	0.36	1.74	0.91	0.82	0.56	0.11	0.43
380	1.78	1.00	0.55	0.40	0.07	0.37	1.78	2.40	0.39	0.40	0.07	0.20
360	1.76	1.01	0.55	0.41	0.04	0.37	1.76	2.50	0.31	0.40	0.04	0.17
340	1.78	1.04	0.52	0.40	0.02	0.36	1.78	2.63	0.23	0.40	0.02	0.14
320	2.25	0.48	0.54	0.38	0.02	0.52	2.25	0.47	0.60	0.38	0.02	0.53
300	2.64	0.54	0.37	0.36	0.02	0.46	2.64	0.53	0.41	0.36	0.02	0.47

6. CONCLUSIONS

The objective of this study is to compare the cost effectiveness of pull control policies in hybrid manufacturing / remanufacturing systems. To this end, first we adapted the most common four pull control policies in the literature to the hybrid production system we proposed, viz. KCS, BSCS, EKCS and GKCS. Here, we analyzed these systems using a decomposition approximation. The adapted analytical methods for each control policy provided us performance measures required for the evaluation of these systems, in terms of their total average costs. These performance measures are, for instance, the average queue length of cores, the average finished product queue, the average work-in-process queue of both remanufacturing and manufacturing processes and the average backorder queue.

After obtaining necessary performance measures, we defined a cost function that consists of holding, backorder, variable production and disposal costs. Holding cost for cores, work-in-processes and finished products, and backorder cost are calculated by multiplying their cost rates with their respective average queue lengths. Disposal cost is obtained by multiplying the unit backorder cost with the disposal rate, and, production costs are calculated by multiplying unit production costs with their related expected throughputs. Summation of these components provided us with the cost function that we want to optimize. Then we searched for local minima for the total average cost with respect to the kanban sizes, base stock levels and static routing probabilities.

After defining the cost functions for each control policy, we compared them using an experiment set generated following the Taguchi method. Then we selected some experiments in order to perform sensitivity analysis. We analyzed, how independent changes in backorder cost, holding cost, remanufacturing and manufacturing cost and disposal cost affect the performances of each system.

We observed from the numerical results, that the control policy which provides the best performance in terms of cost is EKCS in all cases considered, because the more complicated the system is getting, and the more parameters it utilizes to control the hybrid production system, the lower total cost it provides. GKCS follows it with a small margine, as the second best control policy. This fact is a consequence of the higher flexibility related to complicated systems that results in the capability of balancing the disposal and, holding and production costs with the backorder cost more effectively. This result is parallel to the results coming from the analysis of pull type controlled ordinary production systems. However, in determining the factors on the superiority of complex pull control systems for hybrid production systems, there are some distinct cases that are different than those of ordinary production systems.

While in traditional production systems the backorder cost is the main motive behind generating complex control mechanisms, in hybrid production systems, two additional factors play an important role. Here, as marginal profitability of remanufacturing increases, complex control mechanisms perform significantly better. Similarly an increase in the return rate of cores while it stays below the demand rate renders better results with complex mechanisms. In our experiments we observed cost saving around 25 per cent. Consequently, if the remanufacturing is not profitable and there is not enough supply of returns for remanufacturing, viz. when the return rate is smaller than the demand rate, or if the remanufacturing is not profitable and when there is excess supply of returns, there is no need to implement complex control mechanisms to the production system.

Another dimension of the problem is the effect of the buffer size on the superiority of the complicated control policies. We observed that, in hybrid production systems with large buffers size of cores, complicated control policies give higher cost savings than those of the systems with smaller buffer sizes. For instance, in cases where the capacity of the core buffer and marginal profit of remanufacturing is high, the cost effectiveness of EKCS and GKCS moves away from that of KCS and BSCS. However, for small capacity of core buffer size, the cost saving of complex control policies is not that significant, even if the marginal profit of remanufacturing is high compared to the manufacturing of raw materials.

When complex control policies do not provide high cost improvements, preferring simple control policies is a more appropriate decision because of their simplicity in factory applications. However, when we are faced with the decision of selecting one of the simple control policies viz. KCS and BSCS, some interesting factors that are unique to the special nature of hybrid production systems become apparent. In fact, as the marginal profitability of remanufacturing increases, as backorder cost increases, holding cost decreases, and as return rate increase, KCS provides a cost saving of about 1-2%. However, as the profitability decreases, as backorder cost decreases and as holding cost increases, KCS is significantly worse than BSCS. These observations are slightly different than those of ordinary systems. Also, we observed that the buffer size of cores plays an important role in deciding which policy is better. In experimental findings, it is apparent that when the buffer of cores has high capacity, KCS is better than BSCS, whereas when the buffer of cores has small capacity; BSCS can perform significantly better than KCS.

In conclusion, we see that pull type control of hybrid production systems with complex control policies displays a better performance than that of simple policies, as in the case of ordinary production system. However, it is also seen that the factors, which make this difference significant, can be different than those of ordinary production systems because of the special characteristics of hybrid production systems. So, there are some interesting cases where unique characteristics of remanufacturing yield unique managerial decisions for the selection problem of pull type control mechanisms, and our research gives a roadmap for those decision makers who are trying to cope with the uncertainties of remanufacturing by using pull type production control mechanisms.

REFERENCES

Aras, N., T. Boyacı, and V. Verter, 2004, "The effect of categorizing returned products in remanufacturing", *IIE Transactions*, Vol. 36, pp. 319-331.

Bayindir, Z. P., N. Erkip, and R. Gullu, 2003, "A model to evaluate inventory costs in a remanufacturing environment", *International Journal of Production Economics*, Vol. 81-82, pp. 597–607.

Bayindir, Z. P., N. Erkip, and R. Gullu, 2007, "Assessing the benefits of remanufacturing option under one-way substitution and capacity constraint", *Computers & Operations Research*, Vol. 34, pp. 487-514.

Baynat, B. and Y. Dallery, 1993, "A unified view of product-form approximation techniques for general closed queuing networks", *Performance Evaluation*, Vol. 18, pp. 205- 224.

Baynat, B., 1995, "Approximate Analysis of Multi-Class Synchronized Closed Queuing Networks", *Third IEEE International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '95)*, pp. 23-27.

Baynat, B. and Y. Dallery, 1996, "A product-form approximation method for general closed queuing networks with several classes of customers", *Performance Evaluation*, Vol. 24, pp. 165–188.

Baynat, B., Y. Dallery, M. Di Mascolo, and Y. Frein, 2001, "A multi-class approximation technique for the analysis of kanban-like control systems", *International Journal of Production Research*, Vol. 39, No. 2, pp. 307-328.

Bruell, S. C. and G. Balbo, 1980, *Computational Algorithms for Closed Queuing Networks*, Elsevier North Holland Inc.

Dallery, Y., and G. Liberopoulos, 2000, "Extended kanban control system: combining kanban and base stock", *IIE Transactions*, Vol. 32, pp. 369-386.

Di Mascolo, M., Y. Frein, and Y. Dallery, 1996, "An analytical method for performance evaluation of kanban controlled production systems", *Operations Research*, Vol. 44, No. 1, pp. 50–64.

Duri, C., Y. Frein and M. Di Mascolo, 2000a, "Performance evaluation and design of base stock systems", *European Journal of Operational Research*, Vol. 127, pp. 172-188.

Duri, C., Y. Frein, and M. Di Mascolo, 2000b, "Comparison among three pull control policies: kanban, base stock and generalized kanban", *Annals of Operations Research*, Vol.93., pp.,41-69.

Frein, Y., M. Di Mascolo, and Y. Dallery, 1995, "On the design of generalized kanban control systems", *International Journal of Operations & Production Management*, Vol. 15, No. 9, pp. 158-184.

Guide, Jr. V. D. R., V. Jayaraman, and R. Srivastava, 1999, "Production planning and control for remanufacturing: a state-of-the-art survey", *Robotics and Computer Integrated Manufacturing*, Vol. 15, pp. 221-230.

Gungor, A., and S. M. Gupta, 1999, "Issues in environmentally conscious manufacturing and product recovery: a survey", *Computers & Industrial Engineering*, Vol. 36, pp. 811-853.

Gupta, S. M., G. Udomsawat, and Y. Al-Turki, 2004, "Multi-kanban model for multi-product disassembly with multiple demands", *Second World Conference on POM and 15th Annual POM Conference*.

Karaesmen, F., and Y. Dallery, 2000, "A performance comparison of pull type control mechanisms for multi-stage manufacturing", *International Journal of Production Economics*, Vol. 68, pp. 59-71.

Korugan, A. and S.M. Gupta, 2001a, “Substitution Policies for a Hybrid System”, *Proceedings of SPIE*.

Korugan A. and S. M. Gupta, 2001b, “Pull Type Control Mechanisms for Single Stage Hybrid Manufacturing Systems,” Laboratory for Responsible Manufacturing, Northeastern University, Boston LRM-001.

Korugan, A. and S.M. Gupta, 2001c, "An Adaptive Kanban Control Mechanism for a Single Stage Hybrid System", *Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing II*, Newton, Massachusetts, October 28-29, pp. 175-181.

Koukoumialos, S. and G. Liberopoulos, 2005, “An analytical method for the performance evaluation of echelon kanban control systems”, *OR Spectrum*, Vol. 27, pp. 339-368.

Liberopoulos, G., and Y. Dallery, 2000, “A unified framework for pull control mechanisms in multi-stage manufacturing systems”, *Annals of Operations Research*, Vol.93, pp.325-355.

Mitra, D., and I. Mitrani, 1990, “Analysis of a kanban discipline for cell coordination in production lines”, *Management Science*, Vol. 36, No. 12, pp. 1548-1566.

Nakashima, K., H. Arimitsu, T. Nose, and S. Kuriyama, 2004, “Cost analysis of a remanufacturing system”, *Asia Pacific Management Review*, Vol. 9, No. 4, pp. 595-602.

Sbiti, N., M. Di Mascolo, T. Bennani, and M. Amghar, 2003, “Modeling and performance evaluation of base-stock controlled assembly systems”, *Analysis & Modeling of Manufacturing Systems*.

Spearman, M.L., D.L. Woodruff, and W. J. Hopp, 1990, “CONWIP: a pull alternative to kanban”, *International Journal of Production Research*, Vol. 28, No. 5, pp. 879-894.

Teunter, R. H., and D. Vlachos, 2002, “On the necessity of a disposal option for cores that can be remanufactured”, *International Journal of Production Economics*, Vol. 75, pp. 257-266.

Teunter, R., E. van der Laan, and K. Inderfurth, 2000, “How to set the holding cost rates in average cost inventory model with reverse logistics?”, *Omega*, Vol. 28, pp.409–415.

Teunter, R., E. van der Laan, and D. Vlachos, 2002, “Inventory strategies for systems with fast remanufacturing”, *Erasmus Research Institute of Management*, ERS-2002-77-LIS.

van der Laan, E. V., M.Salomon, R. Dekker, and L. Wassenhove, 1999, “Inventory control in hybrid systems with remanufacturing”, *Management Science*, Vol. 45, No. 5, pp. 733–747.