An investigation into the extended Kanban control system

Ang, Alvin Wei Hern

2014

Ang, A. W. H. (2014). An investigation into the extended Kanban control system. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/61752

https://doi.org/10.32657/10356/61752

NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

A Thesis Submitted in Fulfilment of the Requirements for the Degree of Doctor of Philosophy

AN INVESTIGATION INTO THE EXTENDED KANBAN CONTROL SYSTEM

Submitted by:

ANG WEI HERN, ALVIN

G0702291G

2014

Supervisor: A/P DR. RAJESH PIPLANI

ABSTRACT

A Kanban Control System (KCS) is a manufacturing/production control system that uses Production Authorization Cards (PAC or also known as Kanbans) to control the Work-In-Process (WIP) of every stage of the Manufacturing Process (MP). It is attached to every finished product, such that once a customer request is received, it is detached and relayed upstream to re-initiate the production process. Thereafter, the finished product is handed over to the customer.

In this thesis, different types of KCS were described and categorized according to their operating behaviours. Three significant pull systems, namely Base Stock (BS), Traditional Kanban Control System (TKCS) and Extended Kanban Control System (EKCS) were investigated. The make-up of the EKCS is a hybrid of both the BS and TKCS combined. It was initially proposed to leverage on the strengths of these two systems. However, thus far, all relevant studies only reported on the qualitative aspect of these systems, but none on their quantitative impact. Thus, the purpose of this research was to study the quantitative performance difference of these systems. Specifically, this thesis's objective was to draw insights from the differences in performance of the EKCS against the BS and TKCS.

This study was conducted in two main phases: first, the analyses of Single Product KCS (SP/KCS) was carried out, followed by Multiple Products KCS (MP/KCS). Both studies assumed only a Single Stage/Server (SS). Matlab was used to optimize the systems, and Arena version 12 used to simulate different parameter settings. The performance comparison was benchmarked with KPIs such as Fill Rate, Average Inventory Level and Average Customer Cycle Time. The results showed that EKCS outperforms its predecessors, TKCS and BS in all scenarios.

There were four key contributions to this research. Firstly, a method was proposed to determine the optimal size of Base Stock, S*, and Number of Kanbans, K*, in a SS/SP/EKCS, which was never done before. Secondly, this report confirmed that EKCS outperforms TKCS and BS in both single and multiple product scenarios. This performance comparison was ensured through simulations. Third, methods to optimize both Multiple-Product Dedicated (MP/De) and Shared (Sh) EKCS systems were proposed. This also had never been done before. Fourth, this research showed

that both De and Sh/EKCS are equivalent. They operate the same way even though their schematics look different.

Despite constant praise for EKCS' performance in the kanban literature, this thesis shows that it outperforms its predecessor, TKCS, only slightly, and only in certain niche scenarios. The worst-performing system turns out to be BS, as it holds a lot of stock in almost all scenarios. Hence, this research has confirmed again that lean or pull production is more effective than push. Current factory floor managers using BS as their production control strategy should consider switching over to the TKCS, or the EKCS in special situations.

ACKNOWLEDGMENTS

The author wishes to express his utmost gratitude first to God, from whom graciously comes all wisdom and knowledge; to Head Of Department Professor Leong Kah Fai for his valuable and supportive advice; to Shahida for her moral support and administrative assistance; to his dear father and mother, who patiently supported their son's work and showing him humility and hard work; to lovely Wendy and baby Tryphena who stood by him; to Dean Alesa Lightbourne, Ph.D., for helping to proof read this thesis; and lastly to every family member and friend who helped in one way or another in the completion of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

BS	*Base Stock*
TKCS	*Traditional Kanban Control System*
EKCS	*Extended Kanban Control System*
KCS	*Kanban Control System*
SS/SP	*Single Stage, Single Product*
SS/MP	*Single Stage, Multiple Product*
MS/SP	*Multiple Stage, Single Product*
i	*index for production stage,* $i = 1, 2$
j	*index for product type,* $j = 1, 2$
B_i^j	*Output buffer of stage i, product j*
B_0^j	*Component buffer of product j*
C_{hj}	*Holding cost per unit per unit time of product j*
C_{sj}	*Shortage cost per unit time of product j*
C	*CONWIP queue containing free CONWIPs*
c	*CONWIP card*
D	*Total demand arrival rate for all product types,* $D = \sum_{j=1}^{m} \lambda_j$
D_i^j	*Demand queue for stage i, product j*
DK_i	*Authorized kanban queue containing stage i kanbans triggered by demand information from the downstream stage*
d	*A demand*

$E[t]$	*1^{st} moment of service time distribution*
$E[t^2]$	*2^{nd} moment of service time distribution*
$G_j(t)$	*General service time distribution for product j*
K_i^j	*Kanban queue containing kanbans of stage i, product j*
k_j	*Number of kanbans for product j*
L	*Total allowable number of products in the system (total WIP for all product types),*
MP_i	*Manufacturing process of production stage i*
N	*Number of stages in a production line*
p	*An actual product*
S_i^j	*Base stock level of stage i, product j*
t_j	*MP processing time for product type j*
$\pi_j(x)$	*Steady state or long run probability of having x units of product type j in the output buffer*
λ_j	*Demand arrival rate for product type j, assuming Poisson process*
μ_A	*Average MP processing rate,*
μ_j	*MP rate of individual type j customer, assuming exponentially distributed service times*
ρ	*Utilization rate, $\rho = \lambda_j / \mu_j$*

For Extended Kanban Control System (EKCS)

$E\{t_{stockout}\}$	*Expected time value of a stock out in the output buffer, B_1*

$E[K_j]$	*Expected number of undispatched kanbans in kanban queue for product j*
$I\,[S_j, K_j]$	*Expected inventory level, as a function of base stock level, S_j, and number of kanbans, K_j, for product j*
K_j	*Number of undispatched kanbans stored in the undispatched kanban queue, K_i^j, for product j*
$(S_j + K_j)$	*Total number of kanbans for product j*

CHAPTER 1

INTRODUCTION

Flow-line manufacturing systems are well known for their efficiency in producing discrete items. A flow-line system can be considered as a sequence of stages, with each stage being a production/inventory system with a manufacturing process and an output buffer. A major challenge in designing and operating flow lines is to achieve high customer service levels while staying lean. Determining the mechanism to control the flow of materials through the manufacturing system is therefore one of the most important decisions of a production company (Boonlertvanich, 2005).

There are two main types of strategies used for production control: push and pull. Push systems schedule periodic release of raw materials into the production line, while pull systems authorize parts to be processed in response to the actual arrival of demand. Push systems batch and control the system's release rate (hence throughput) and monitor work-in-process (WIP) periodically, while pull systems control WIP and monitor throughput. A push strategy pre-schedules production jobs according to capacity. On the other hand, a pull strategy focuses on balancing production flow, and triggers the production of a new job at the arrival of demand or completion of an existing job.

Figure 1.1 shows the difference between a pure push and a pure pull system. The circle represents a manufacturing process, while the inverted-C behind it represents its buffer. The main difference lies in the signal used to trigger production. Pure pull systems rely on signals from downstream (in the form of kanbans, to be defined later) to start production, while pure push systems rely on customer demand forecasts.

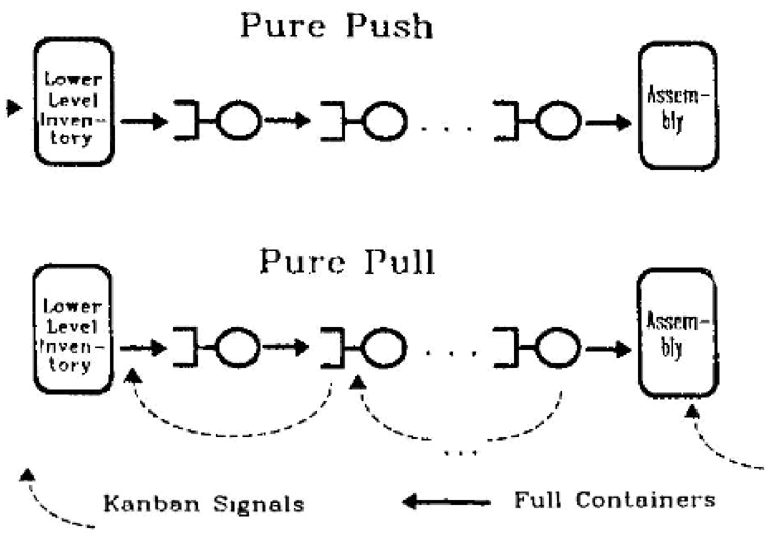

Figure 1.1: Pure Push vs. Pure Pull Systems (Spearman, 1990)

1.1. Push System

The best-known planning approaches used in push-type control systems, Materials Requirements Planning (MRP) and Manufacturing Resource Planning (MRP II), are at the core of Enterprise Resource Planning (ERP) systems. MRP explodes the bill of materials, and develops the schedule of material releases using forecasted demand and expected lead-times. MRP is a push system, as releases are made according to a master production schedule without regard to system status. Hence, no a-priori WIP limits exist (W. J. Hopp & Spearman, 2004). This approach has been widely implemented, especially in industries characterized by complex products, a high-volume of job orders, and/or complex supply chains. The strength of MRP lies in its ability to work through bill-of-materials relationships and provide the basis for coordination among plants, suppliers and customers (Weitzman & Rabinowitz, 2003).

Figure 1.2 shows the MRP system architecture (Spearman, 1990). Based on forecasted customer orders, a Master Production Schedule (MPS) is created (as an outcome of Master Scheduling). A “rough-cut” planning module is added to aid the scheduling task. MRP then releases production orders based on planned lead times and lot sizes.

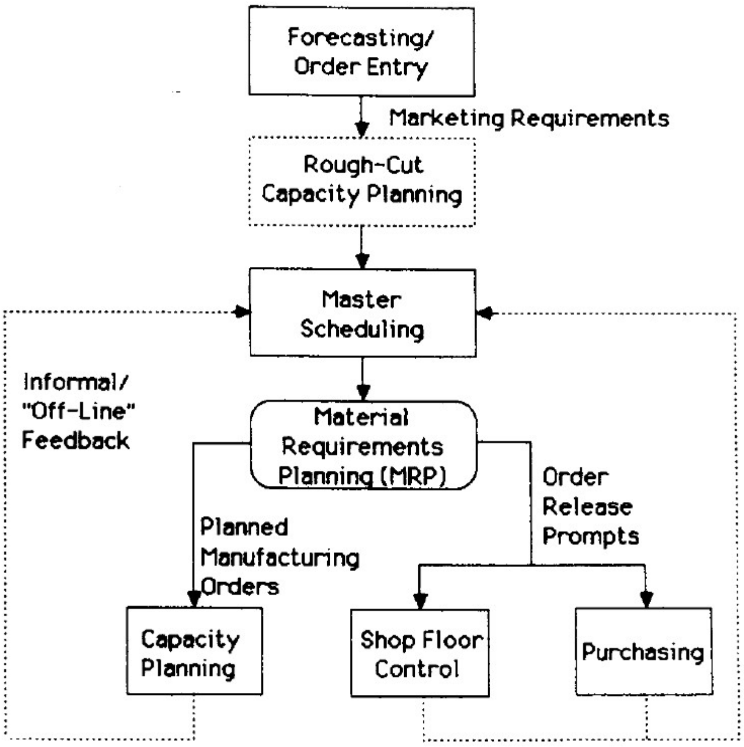

Figure 1.2: MRP System Architecture (Spearman, 1990)

Despite the strengths of MRP as mentioned earlier, it has certain limitations:

1. *MRP uses sequential and independent processing of information*: Material requirement planning is done at a level above shop floor control and without considering manpower and machines. Most of the time, production plans are found to be infeasible too late in time for the production process to recover or restart. To account for this weakness, buffers of inventory are embedded in the system as safety stock.

2. *There are no formal feedback procedures:* Only ad hoc, off-line and manual feedback procedures exist, as shown by the dotted arrows in Figure 1.2. This leads to a "bullwhip" effect, where each stage in the production process tries to forecast demand of the next stage in order to properly position inventory and other resources. Since forecasts are based on previously available statistical data, they are never perfect. Moving up the production process from end-consumer to raw materials, each production stage has greater observed variation in demand, and thus greater need for safety stock. In periods of rising demand, down-stream stages increase

their orders; in periods of falling demand, orders fall or stop in order to reduce inventory.

1.2. Pull System

The best known pull approach is the kanban system, the backbone of the Just-In-Time (JIT) production strategy. It was first adopted by Toyota, the biggest car manufacturer in the world today (Watanabe, 2007). Although Toyota had adopted this system in all its production plants as early as 1962 (Spearman, 1992; Yavuz & Satir, 1995), it was not until 1977 that the first publication in English about the kanban system appeared (Sugimori, Kusunoki, Cho, & Uchikawa, 1977). A vast number of books and papers have since been published (Berkley, 1992; Golhar & Stamm, 1991; Huang & Kusiak, 1996; Keller, 1993; Kumar, 2007; Price, Gravel, & Nsakanda, 1994).

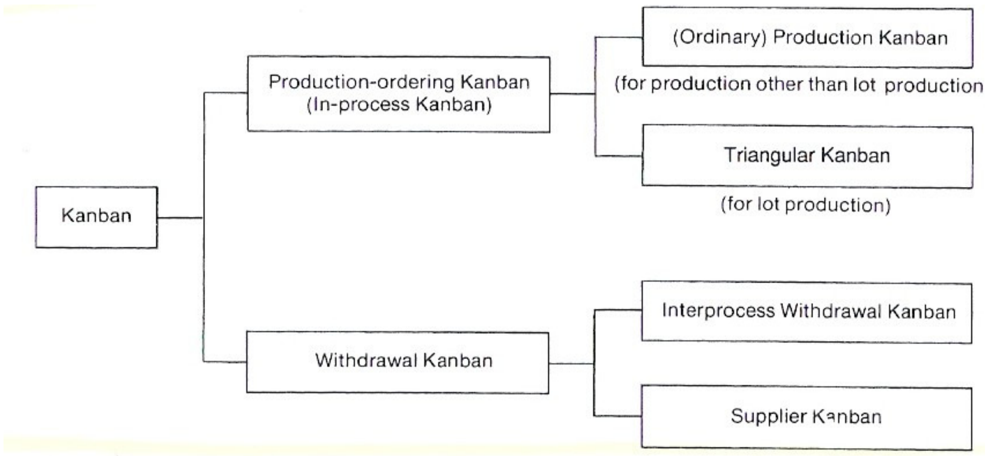

Figure 1.3: Types of kanbans (Monden, 1993)

Kanbans, or production authorization cards, are used to control and limit the releases of parts into each production stage. Figure 1.3 shows the different types of kanbans commonly used by Toyota (Monden, 1993). On modern factory shop-floors, many different types of kanbans can be found (e.g. triangular, supplier and inter-process withdrawal kanban, etc.), each designed for a specific use depending on the needs of the individual factory.

The kanban system is widespread among manufacturing companies today

(Black, 2007; Lee-Mortimer, 2008; Singh, Kwok Hung, & Meloche, 1990; White, Pearson, & Wilson, 1999). The reason is its 'lean' concept. By applying a kanban system to replace a traditional push production strategy or Material Requirement Planning (MRP) system, with certain prerequisites a company can theoretically reduce its inventory levels (Fullerton & McWatters, 2001), which are commonly seen as 'evil' or 'waste' of resources (Monden, 1993). A significant body of literature has already documented the advantages of the pull strategy over the push; examples include Spearman (1990), Weitzman and Rabinowitz (2003) and Wallace J. Hopp and Spearman (2008). Some of these advantages include (W. J. Hopp & Spearman, 2004):

1. *Reduced Work-In-Process (WIP) and Cycle Time*: Kanban regulates WIP by limiting releases into the system, resulting in a lower average WIP level. By Little's Law (Work-In-Process = Cycle Time x Throughput), this also translates into shorter manufacturing cycles.

2. *Smoother Production Flow*: By dampening fluctuations in WIP level, kanban achieves a steadier, more predictable output stream, as opposed to the MRP system which uses forecasting methods, resulting in the "bullwhip" effect.

3. *Improved Quality*: A kanban system applies pressure for better quality parts by only allowing short queues through the "one-piece conveyance flow" concept. This concept, proposed by Ohno (1988), specifies that only one part be allowed to wait in the queue at every stage. Only when the part is depleted by the following stage can the manufacturing process start to produce another. In a real factory scenario, shorter queues allow for easier detection of defects, as the line is required to be shut down once a high level of rework is detected.

However, some of the disadvantages of pull include (Wallace J. Hopp & Spearman, 2008):

1. *Tighter Line Pacing:* It induces a tighter pacing of the line, giving operators less flexibility for working ahead and placing considerable pressure on them to replenish buffers quickly.
2. *Unaccommodating to Product Variety*: The use of product-specific cards means that at least one standard container of each part number must be maintained at each station, to allow the downstream stations to pull what they

need. This makes it impractical for systems with numerous part numbers. In other words, a simple single product kanban system would be more practical for use. A multiple product kanban system would be complex and difficult to implement in practice.

3. *Unaccommodating to Fluctuating Product Volume*: It cannot accommodate a changing product volume, unless a great deal of WIP is loaded into the system; but then again this would go against the lean philosophy. This is because the product-specific card counts rigidly govern the mix of WIP in the system. This fact has also been reported by Monden (1998) that because Toyota had been so strict with fixed kanban numbers that their upstream suppliers find it difficult to match their production.

1.3. Research Objectives and Thesis Layout

In this thesis, a Push system is formally defined as one that does not contain kanban cards; whereas a Pull system is formally defined as one that contains kanbans. This research focuses on lean production systems, better known as pull or Just-In-Time (JIT) systems. Lean production originated from Toyota (Japan), as a 'weapon' to fight against the West (America) when mass production was taking over automobile production, in the 1960s (Womack, Jones Daniel T., & Roos Daniel., 2007). Toyota's chief engineer, Taichii Ohno, devised kanbans to control the WIP of the factory floor. Since then, many new kanban systems have been proposed. Others have been customized to fit individual production floors.

One such important hybrid kanban system is the Extended Kanban Control System (EKCS). This system, first proposed by Dallery (2000) boasts the advantage of both the "push" and the "pull" by merging together–the Base Stock (BS) and the Traditional Kanban Control System (TKCS). As their names suggest, Base Stock (BS) system keeps a certain level of inventory (or "base stock") to satisfy future demand arrivals; its roots lie in the "push" philosophy. The Traditional Kanban Control System (TKCS) is the pure kanban system, described above, that tries to minimize stock; it comes from the "pull" family of systems. Its proponents have claimed that it is even "leaner" (carries lesser stock than the BS and TKCS), yet results in fewer backorders and shorter customer waiting times. Since Extended Kanban Control System (EKCS) was proposed, however, many questions have remained unanswered:

1. Is the Extended Kanban Control System (EKCS) really as powerful as its proponents claim? Much of the kanban literature has proclaimed its benefits over older systems (Chaouiya, Liberopoulos, & Dallery, 2000; Liberopoulos, 2005) while some have even made modifications to it (Claudio & Krishnamurthy, 2008). However these claims are qualitative in nature, and no quantitative comparisons have been conducted to date.
2. How will the Extended Kanban Control System (EKCS) perform in different scenarios, such as in single product or multiple product cases? Will it still outperform the older systems?
3. What benefits will EKCS bring to future factory floor managers?

In Chapter 2 a literature review of the different types of kanban systems proposed to date is presented. These are separated into three main categories: pull, push and hybrid pull systems. In the push system category, the Base Stock (BS). In the pull system category, Traditional Kanban Control System (TKCS) and CONstant Work–In–Process (CONWIP) are discussed. Hybrid pull systems discussed are the Generalized Kanban Control System (GKCS), CONWIP Kanban (CK), Extended CONWIP Kanban (ECK) control system and, lastly, the Extended Kanban Control System (EKCS). In the final section, a table summarizing the characteristics, pros and cons of each system is presented.

Chapter 3 presents the EKCS – its operation, the gaps in current research about it and a suggested method to optimize it. The two main research gaps are: insufficient study of its performance, and a need to optimize EKCS. The most important part of this chapter is the method to optimize EKCS. This work has never been done before, and the method proves useful when compared vis-à-vis other systems in later chapters.

In Chapter 4, a detailed performance comparison of Single Stage, Single Product Kanban Control Systems (SS/SP/KCS) is discussed. This chapter starts off with a description of the methods to optimize the Single Stage, Single Product Base Stock (SS/SP/BS) and Traditional Kanban Control System (TKCS). Thereafter, simulation parameters, snapshots and assumptions are presented. Following that, simulation results and insights from these results are discussed. Lastly, the results are validated, and further conclusions are drawn.

Chapter 5 onwards showcases Multiple Product Kanban Control Systems (MP/ KCS). These are the Single Stage, Multiple Product Base Stock (SS/MP/BS), Dedicated and Shared Traditional Kanban Control System (De and Sh–TKCS), and Dedicated and Shared Extended Kanban Control System (De and Sh–EKCS). Chapter 5 presents their operations and characteristics and discusses optimization of MP/KCS. Since each has different operating characteristics, they also need specialized optimization methods. This chapter presents novel optimization techniques for MP/KCS developed in this research. These optimization methods will help in comparing the performance of the various MP/KCS in the next chapter.

Chapter 6 presents a performance comparison of MP/KCS, utilizing optimization methods presented in Chapter 5. Finally, Chapter 7 presents the conclusion and future work of this research.

CHAPTER 2

LITERATURE REVIEW

Just-In-Time (JIT) is one way to control the flow of products in manufacturing (Sugimori et al., 1977). JIT gained worldwide prominence based on the success of Japanese companies in the early 1980s. The backbone of JIT is the Kanban Control System (KCS), which then became a popular topic in the Western world in the '80s. Manufacturing companies outside Japan soon began to use kanbans to control production and flow of materials.

In Traditional Kanban Control System (TKCS), kanbans are used to control and limit the release of parts into each production stage. The advantage of this mechanism is that the number of parts in a stage is limited by the number of kanbans of that stage.

This chapter is divided into two main parts. The first (sections 2.1 to 2.3) presents kanban's management philosophy, the formulae, single versus dual card kanban type and their limitations. In sections 2.4 to 2.5, different types of push-pull production control policies are elaborated. These hybrid pull policies have been developed to facilitate production control that reacts to actual demand rather than future demand forecasts (Boonlertvanich, 2005). The operational procedures and shortcomings of these policies are also described.

2.1. Kanban Management Philosophy

Proposing a new production control policy is easier than implementing it in a real factory scenario. Successful implementation requires fundamental changes in existing production controls in order to align with the Just-In-Time (JIT) philosophy; this is the most difficult part (Boonlertvanich, 2005). Monden (1998) has stated five important kanban rules for Just-In-Time (JIT) production. Although these rules are only general management concepts of how the kanban system should be applied, they are very important in laying the foundation for extending existing pull control policies.

Rule (1): The next production process should only withdraw the necessary parts from the previous process in the necessary amount, at the necessary time.

Rule (2): A previous process should produce only the necessary quantities withdrawn by the next process.

Rule (3): Defective products should never be passed down to the next process. This rule refers to "Autonomation" or "Jidoka", built-in mechanisms to prevent defective work using "fool-proof" or "Pokayoke" methods.

Rule (4): The total number of kanbans should be minimized. According to Monden (1998), waste originates from an increase in inventory levels; thus only the supervisor for each process has the authority to change the number of kanbans. Since the total number of kanbans allocated in each stage represents the work-in-process for that stage, minimizing the number of kanbans means lowering inventory levels.

Rule (5): A kanban system should be used to adapt to small fluctuations in demand. If a company adopts the system, it can meet demand variations of up to around 10% by changing the frequency of kanban transfers without revising the total number of kanbans. The total number of kanbans should be revised only if there are large seasonal demand changes. Therefore, a system with a fixed number of kanbans should only be used over the short-term.

Though there are many variations of the kanban formula, the most popular is the Toyota Kanban Formula (Sugimori et al., 1977). It can be represented as:

$$y = \frac{D\left(T_w + T_p\right)\left(1+\alpha\right)}{a} \tag{2.1}$$

Where

y : Number of Kanbans

D : Demand per unit time

T_W : Waiting time of kanban

T_P : Processing Time

a : Container Capacity

α: Policy Variable, the parameter that the manager has direct control over

Monden (1993) has called ‘α’ the safety coefficient for the factory to cope with disturbances, while Sugimori et al. (1977) called it a “policy variable” determined according to the factory’s capability to manage external interference. The kanban formula was designed for stable demand arrival rates and stable lead times. The formula does not take into account variability with respect to time, i.e. y is not a function of time. Hence α is regarded as a “safety coefficient” to increase the number of kanbans in time periods when demand and lead time fluctuate.

If D increases, the value of (T_W + T_P) must be reduced so that the demand arrival rate will not exceed the processing rate (else the system will be “choked” with demand arrivals). A factory with insufficient capability for improvement (incapable of reducing the total lead-time) has to cope with the situation by increasing ‘α’, or the number of effective kanbans, over the short-term. Senior managers will then consider ‘α’ as an indicator for shop capability improvement. This is shown in the example below.

In the first period, t = 1, the demand arrival rate D is 0.5 parts per day while T_w and T_p each is 0.5 day. Hence the processing rate is 1 part per day. Since the demand arrival rate is less than the processing rate, the system is able to handle the demand. Furthermore, let ‘a’ be 1 part per container and assume ‘α’ to be 0, i.e. no managerial interference. After inserting the respective values into the formula, the number of kanbans, y, equals to 1 (rounded up from 0.5). Note that after the number of kanbans (y=1) is calculated by the management, this value becomes fixed as long as there are no further revisions in future time periods.

Now for the next time period, t = 2, say the demand arrival rate D increases to 2 per day while the other parameters remain constant (i.e. T_W and T_P each are 0.5 per day; processing rate is 1 part per day; ‘a’ is 1 part per container and ‘α’ is 0). Now y has increased to 2. Since y was determined as 1 in the previous period, the actual y value on the factory floor still remains as 1. But now D is more than the processing rate, and the system will get choked. This means that T_W and T_P have to be lowered to less than 0.25 each for the system to be able to cope with the new demand.

Next assume ‘α’ to be 0.1 at t = 1, while all other parameters remain constant; y now becomes 2 (rounded up from 1.1). Hence when D increases to 2 per day at t = 2, the 2 kanbans will authorize the 2 arriving demands per day into the system. Since the processing rate is unable to handle the increased demand, there will be an additional kanban (representing the extra demand) waiting in queue at the manufacturing process. These additional kanbans act as a form of visual signal for the manager that his shop is in need of improvement to cut down T_W and T_P.

The limitation of this formula is that it is only applicable if the company has stable demand arrivals. Furthermore, it can only be used in the Traditional Kanban Control System (TKCS), and is not applicable to other pull control policies because they have different operating mechanisms.

2.2. One Card Kanban System

The kanban systems can be classified into two main categories: the one-card system (using only the Production Ordering Kanban) and the two-card system (using both the Production Ordering Kanban and the Withdrawal Kanban). Even though the two-card system is used in the industry (Chao-Hsien & Wei-Ling, 1992) and the seminal paper about kanban system described it as a two-card system (Sugimori et al., 1977), most of the kanban research literature uses a one-card system in the models because of its simplicity (Bard & Golany, 1991; Groenevelt & Karmarkar, 1988; Köchel & Nieländer, 2002; Krajewski, King, Ritzman, & Wong, 1987; Krieg, 2005; Marek, Elkins, & Smith, 2001; Sarker & Fitzsimmons, 1989; Sarker & Harris, 1988). Only some papers have also used the two-card system as a basis for their modelling and research (Y. P. Gupta & Gupta, 1989; Nomura & Takakuwa, 2004; Katsuhiko Takahashi, Nakamura, & Ohashi, 1996).

One-card systems are easy to understand and implement because they use the same card to authorize material movement and production. Two-card systems are similar to one-card systems, but differ in the use of two different types of cards to control production and material movement separately (Marek et al., 2001). In fact, the two-card kanban system is the result of an artificial distinction between parts processing and material movement. If both of them are considered as a single process, the two-card kanban system becomes a one-card system with the move card becoming a "production" card for the move process (W. J. Hopp & Spearman, 2004).Since this thesis focus is only on the One Card Kanban System, the Two Card System will not be described.

The Single Card Kanban System is used if the distance between the consecutive workstations is very short. Here, only one type of card is used: the Production Order Kanban (POK). A single buffer is used between the workstations. This buffer acts as an outbound buffer for the current workstation j as well as an inbound buffer for the succeeding workstation j+1.

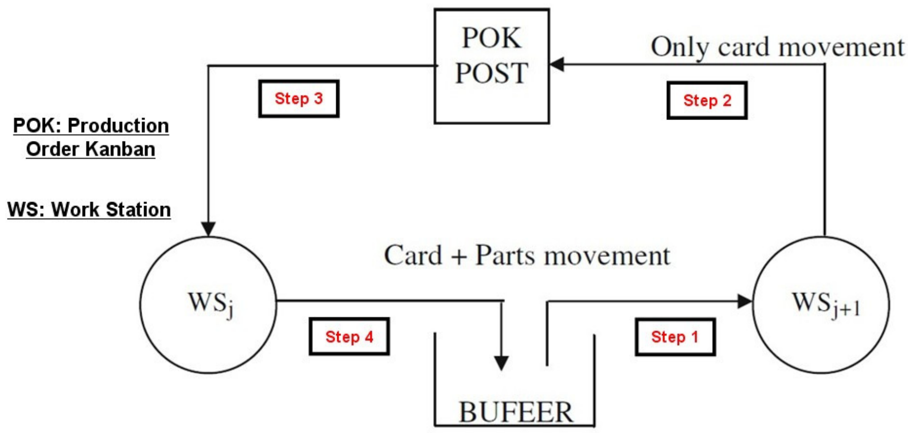

Figure 2. 1: One-Card Kanban System (Kumar, 2007)

The following steps describe the operations of a One-Card Kanban System (Figure 2.1):

Step (1): When a part is to be processed by WS_{j+1}, the part together with its attached POK card is removed from the buffer.

Step (2): Before the part is processed in WS_{j+1}, the POK card is detached from the part and attached to the POK POST. This is a scheduling board which signals the type and quantity WS_j is to produce.

Step (3): When WS_j becomes available for production, it refers to the POK POST, removes the earliest POK card from the POST and produces accordingly.

Step (4): After WS_j finishes producing the required part type, the POK card is attached to the finished part and both of them are placed in the output buffer. The process then repeats itself.

2.3. Classification of Push and Pull Production Control Policies

The founder of the Toyota Production System (TPS), Taichii Ohno, once said that "TPS has to 'evolve' constantly to cope with severe competition in the global marketplace" (Monden, 1998). Even since then, researchers have been trying to improve the system by proposing variations to the Traditional Kanban Control System (TKCS) (K. Takahashi, Morikawa, & Chen, 2007). According to Kolchel & Nielainder (2002), kanban research has taken two directions: synthesis and analysis. The synthesis approach aims at designing a new kanban system that fulfils predefined conditions. The analysis approach, on the other hand, deals with performance analysis of kanban systems under different structural deviations (Hao & Shen, 2008). It has been noticed that most of the researchers in the kanban literature:

a. Propose a new pull-control mechanism (Boonlertvanich, 2005) or modify existing pull-control mechanisms (Al-Tahat & Mukattash, 2006; A. Krishnamurthy & Suri, 2006); or

b. Compare existing pull-control mechanisms (Jodlbauer & Huber, 2008).

Even though many pull systems have been proposed, their deficiencies remain: they can only be used if there is smooth or level daily production (Monden, 1993). Level daily production can only take place if there is stable demand. This is an intrinsic limitation of kanban systems (Section 2.1 Rule (5):).

In case of Toyota's plants in Japan, advanced planning must be done before the kanban system can be applied, and Toyota must inform the predetermined production schedule to all their suppliers one month in advance. Monden (1993) warned that if not implemented properly, the kanban system will become a "reverse weapon," causing more harm than good to the company. It has also been reported that most of Toyota's subcontractors in Japan keep large amounts of inventory to deal with the eventuality of their main customer, Toyota, varying its demand (Monden, 1998). This had created the need to improve the Traditional Kanban Control System (TKCS).

There are two categories of production systems: Push and Pull. Base Stock (BS) falls under the Push category because it does not contain kanbans. Traditional

Kanban Control System (TKCS) and CONstant Work-In-Process (CONWIP) fall under the Pull category because they contain kanbans, yet do not have the Base Stock (BS) element. That is, they do not have instantaneous transmission of demands. The Generalized Kanban Control System (GKCS), Extended Kanban Control System (EKCS), CONWIP Kanban (CK) and Extended CONWIP Kanban (ECK) fall under the hybrid category because they are hybrids of the BS and TKCS. The classification of the pull systems is shown in Figure 2.2.

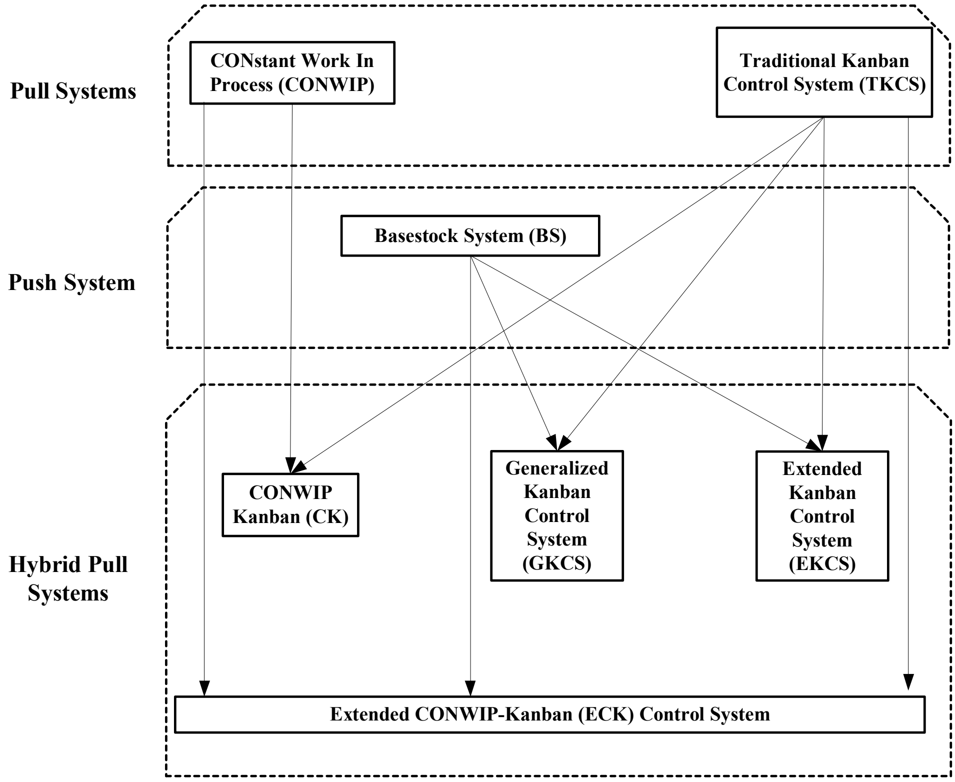

Figure 2. 2: Different Push-Pull Production Systems

2.4. Push System

2.4.1. Base Stock (BS) System

Base-Stock (BS) is largely classified as a push system due to its distinct characteristic of holding inventory, known as base stock. However, some authors have labelled it as a pull system because it only responds to customer demands. In this thesis, it will be labelled as Push, because it does not contain kanbans, and is important as it is part of Extended Kanban Control System (EKCS)–the focus of this research. Base Stock (BS) was first proposed by Clark and Scarf (1960).

Figure 2.3 shows a two stage production line controlled by Base Stock (BS). It does not limit Work–In–Process (WIP) (W. J. Hopp & Spearman, 2004), but limits inventory stored in output buffers by setting the base stock level. Also, it does not use cards as feedback signals to previous stages–hence no coordination exists between consecutive stages. To compensate, it has instantaneous transmission of demand signals to all production stages. This signal can be sent by using either a card-based or a computer-based system. Queues D_i, where i = 1, 2, 3, contain the customer demands.

When the system is in its initial state (before any demands arrive), Buffer B_i, i = 1, 2, contains S_i, i = 1, 2, number of base stocks of finished products. Buffer B_0 is the components buffer and is assumed to contain an infinite quantity of components.

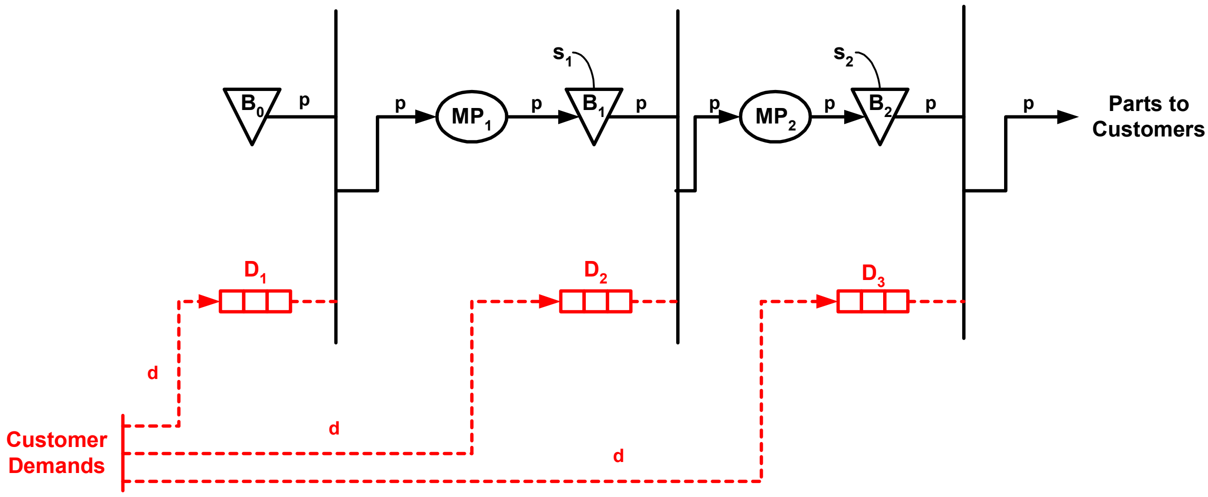

Figure 2.3: A two-stage production line controlled by a Base Stock (BS) System (Clark & Scarf, 1960)

The Base Stock (BS) System operates as follows:

1. When a customer demand arrives at the system, it is replicated into N+ 1 demands, and each one is immediately transmitted to its respective queue D_i. The last one joins Queue D_3, requesting the release of a finished product from buffer B_2 to the customer.

2. At this point there are two possibilities:

   a. If a product is available in B_i, it is released immediately to the downstream stage, and MP_{i+1} will produce one part to top up the base stock s_{i+1} in B_{i+1}.

If this is the last stage, and a part is available in B_2, then a finished part is released to the customer and the demand d_3 is satisfied.

b. If no product is available in B_i, the demand is backordered and waits in Queue D_{i+1} until a new part completes the upstream stage.

Base Stock (BS) depends on one parameter per stage, namely S_i, $i = 1, \ldots, N$. The production capacity of the system does not depend on S_i. That is, the output buffers are bounded by the base stock level, but the WIP levels in each stage are unbounded.

The advantage of Base Stock (BS) is that it tries to set a target in the output buffer level by bounding each stage's base stock level. There is also no demand information blockage because of instantaneous transmission of demands to all production stages.

Base Stock (BS) has no feedback system, and hence no coordination exists between stages. This system does not set a limit on WIP levels in each stage or for the entire production line. When a stage fails, the demand process will continue to remove parts from the output buffer, and the machines downstream of the failed machine will operate normally until it becomes starved of parts to process. The upstream stages continue to receive direct demand information and will operate and release parts as usual. This may lead to an unbounded build-up of inventory in front of the failed machine.

2.5. Traditional Pull Systems

2.5.1. Traditional Kanban Control System (TKCS)

Traditional Kanban Control System (TKCS) was first proposed by Sugimori et al. (1977). Figure 2.4 shows a two-stage production line controlled by a TKCS. The number of kanbans limit the WIP (W. J. Hopp & Spearman, 2004). This system is the most famous pull mechanism proposed in the last few decades. It limits the amount of inventory for each stage, such that the maximum WIP is equal to the number of kanbans circulating in that stage.

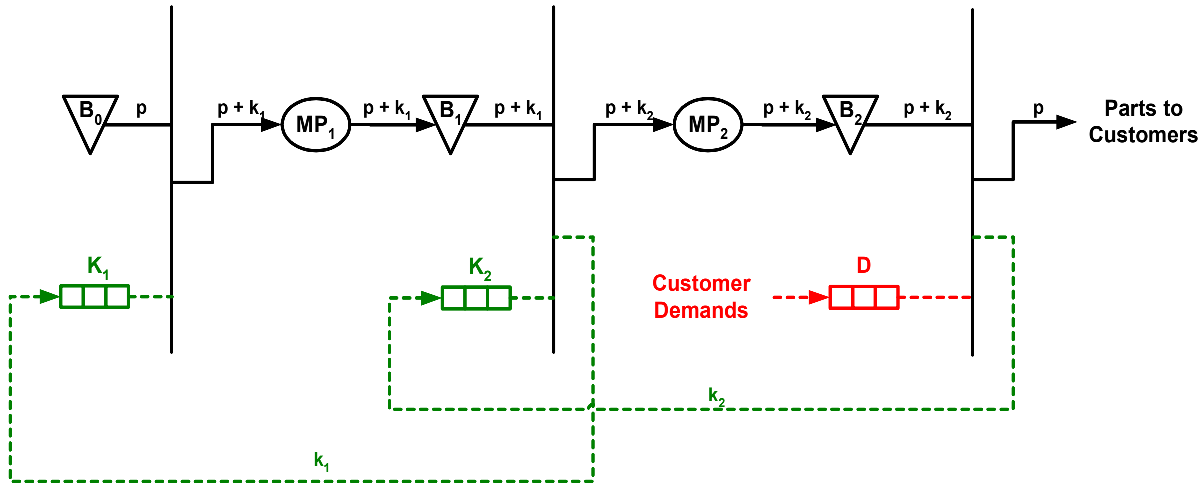

Figure 2.4: A two-stage production line controlled by a Traditional Kanban Control System (TKCS) (Sugimori et al., 1977)

Buffers B_i, i = 1, 2 are the output buffers of stage i containing both finished parts and stage i kanbans. Queue K_i contains stage i kanbans. The kanban movement is shown by the dashed line. Buffer B_0 is the component buffer and contains an infinite number of parts. When the system is in its initial state, B_i contains k_i finished parts, each part having a stage i kanban attached to it. All other queues are empty. Traditional Kanban System (TKCS) operates as follows:

1. When a customer demand arrives at the system it joins Queue D, requesting the release of a finished product from B_2 to the customer.

2. At that time there are two possibilities:

   a. If a product is available in B_2 (which is initially the case), it is released to the customer after detaching the stage 2 kanban. This kanban is transferred upstream to queue K_2, carrying with it a demand signal for the production of a new stage 2 finished part.

   b. If no product is available in B_2, the demand is backordered and waits in Queue D until a new part is completed and arrives in B_2. The newly finished part will be released to the customer instantly, and the detached kanban will transfer to Queue K_2 instantly as well.

As soon as a kanban signal arrives in K_2, it authorizes the production of a new part in stage 2. Again, at this time one of two things may happen:

1. If a product is available in B_1 (with a stage 1 kanban attached), the stage 1 kanban is instantaneously detached and a stage 2 kanban attached to it. At the same time, the pair (i.e. the product and the attached stage 2 kanban) is released into MP_2. The stage 1 kanban is transferred upstream to K_1, authorizing the release of a raw part from B_0 into MP_1.

2. If no product is available in B_1, the stage 2 kanban waits in K_2 until a newly finished product arrives in B_1.

Customer demand information is transferred upstream by kanban signals. If at a stage 'i' a finished product is not available in B_i, no kanban is transferred upstream, and the demand information is temporarily held together with the kanban card in K_i. The idea is that demand is transmitted upstream from stage i only when a finished product is released downstream from the stage. Traditional Kanban Control System (TKCS) depends on only one parameter per stage, namely k_i, i = 1, … N, which helps transfer finished parts downstream and sends demand signals upstream.

In Traditional Kanban Control System (TKCS), the kanban and demand are considered as one entity, except in the very first demand Queue D, where the initial customer demands are stored. The transfer of a finished part from B_i to MP_{i+1} is synchronized with the transfer of a kanban, together with a demand, from K_{i+1} (or D if i = N) to K_i.

The advantage of a kanban is that it acts as a form of feedback, resulting in coordination between stages. Another advantage is that it sets a limit on WIP levels at each stage. However, demand signal blockage can sometimes occur, since demand can only flow upstream if the downstream demand is satisfied. In addition, the kanban provides no instantaneous transmission of demand information to all production stages, neither does it set a limit on the Work–In–Process (WIP) for the entire production line. Lastly, the system doesn't respond well to long-term demand fluctuations.

2.5.2. CONWIP

CONWIP stands for CONstant Work–In–Process. This system was first proposed by Spearman, Woodruff, and Hopp (1990). Figure 2.5 shows a two stage production line controlled by CONWIP. This is a pull system as it limits Work–In–Process (WIP) via cards similar to kanbans (W. J. Hopp & Spearman, 2004). The number of CONWIP cards represents the total WIP allowed. When the preset WIP is reached, no new parts can be released into the system until finished parts have been discharged. CONWIP can also be seen as a single kanban cell encompassing all stages (Boonlertvanich, 2005). That is, a single-stage Traditional Kanban Control System (TKCS) is equivalent to a single-stage CONWIP system. CONWIP control is executed only at the entry of the manufacturing system.

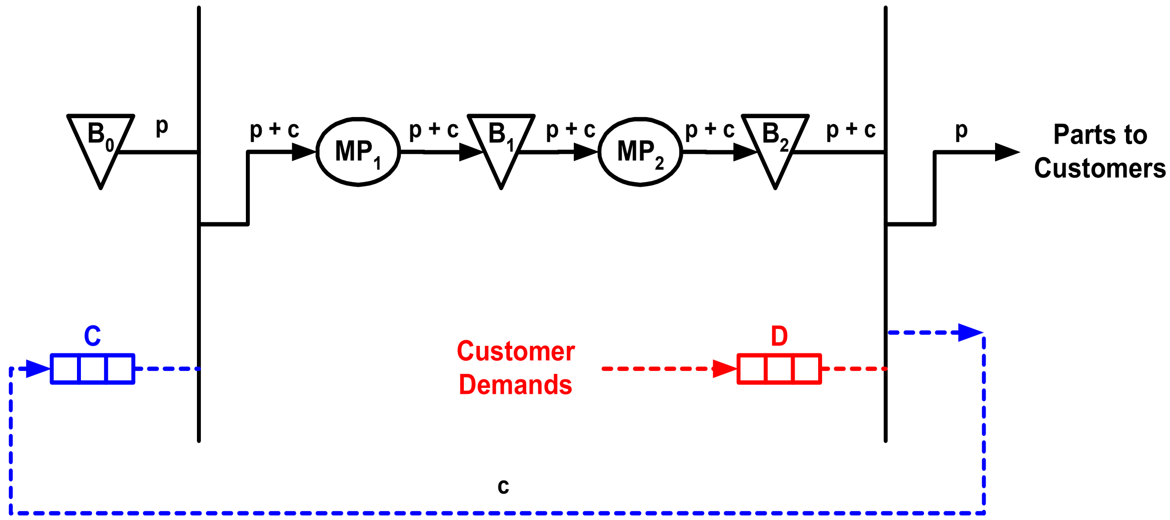

Figure 2.5: A two-stage production line controlled by CONWIP system (Spearman, Woodruff & Hopp, 1990)

MP_1 and MP_2 represent manufacturing stages 1 and 2 respectively. Buffer B_0 is the component buffer, and is assumed to contain infinite numbers of components. Queue B_i, i = 1, 2, is the output buffer of stage i. MP_i contains total number of products that have been released into stage I; Queue D contains the demand, and Queue C contains CONWIP cards/signals. When the system is in its initial state, before the arrival of any demand, Buffer B_1 is empty. Buffer B_2 contains C finished products attached with the CONWIP cards. A CONWIP system operates as follows:

When a customer demand arrives, the system requests the release of a finished product from B_2 to the customer. At this time there are two possibilities:

1. If a product is available in B_2 (which is the case initially), it is released immediately to the customer, and the CONWIP card is detached from the product and transferred to Queue C.
2. Otherwise, the demand is backordered and waits in D until a new product from the upstream stage arrives.

The advantage of CONWIP over kanban system is that instant feedback is provided between the first and last stage. This allows shop floor managers to dictate part sequence and schedule top-priority jobs earlier at the initial stage. In the kanban context, on the other hand, demand information is sent to the first stage only after passing through many intermediate stages. By then, this late information may not be useful to top-line managers. Another advantage of CONWIP is that all stages other than the last operate the same way as a push, hence parts move downstream without any blockage. CONWIP cards provide feedback on the actual performance of the entire production line. These cards are simpler, as there is only one type of card; kanban needs to set many cards, one for each stage. CONWIP also sets a limit on WIP levels for the entire production line.

CONWIP depends on one parameter for the entire system, namely the amount of CONWIP, C. There is no demand signal transfer between intermediate stages. Hence, if a stage fails, its downstream parts will be gradually flushed out of the system by demands. But in the upstream stage, demands continually trigger the release of new component parts into the system. Only when all finished parts from previous stages have accumulated in front of the failed machine will the release of new jobs to the system stop. Also, demand information blockage can occur, since there is no instantaneous transmission of demand signals. Lastly, CONWIP doesn’t respond well to long-term demand fluctuations.

2.6. Hybrid Pull Systems

In this section, Generalized Kanban (GKCS), CONWIP Kanban (CK) and Extended CONWIP Kanban (ECK) systems will be discussed. Extended Kanban Control System (EKCS) is discussed in Chapter 3. A major point of difference is that CONWIP Kanban (CK) and Extended CONWIP Kanban (ECK) have the CONWIP component, but Generalized Kanban Control (GKCS) and Extended Kanban Control System (EKCS) do not.

2.6.1. Generalized Kanban Control System (GKCS)

The Generalized Kanban Control System (GKCS) was first proposed by Buzacott (1989). Figure 2.6 shows a two-stage production line controlled by GKCS. It is a modified version of the Traditional Kanban Control System (TKCS). In the Generalized Kanban Control System (GKCS), each stage has k_i kanbans to authorize production. Initially, all kanbans of each stage, i.e. k_i, are stored in Queue K_i. Buffer B_i, i = 1, 2, has S_i, i = 1, 2, finished parts. Buffer B_0 is the component buffer and is assumed to contain infinite number of components. The demand of stage i is stored in two queues: Queue D_i contains purely demands, while Queue DK_i contains stage i kanbans that have been triggered by demand information from downstream. Kanban is now a separate entity from the demand. Generalized Kanban Control System (GKCS) depends on two parameters per stage: the number of kanbans in each stage, k_i and the base stock level of that stage, S_i.

GKCS operates as follows:

When a customer demand arrives at the system, it is instantaneously split into two demands. The first demand joins Queue D_3 requesting the release of a finished product from B_2 to the customer. The second demand joins Queue D_2 requesting production at stage 2.

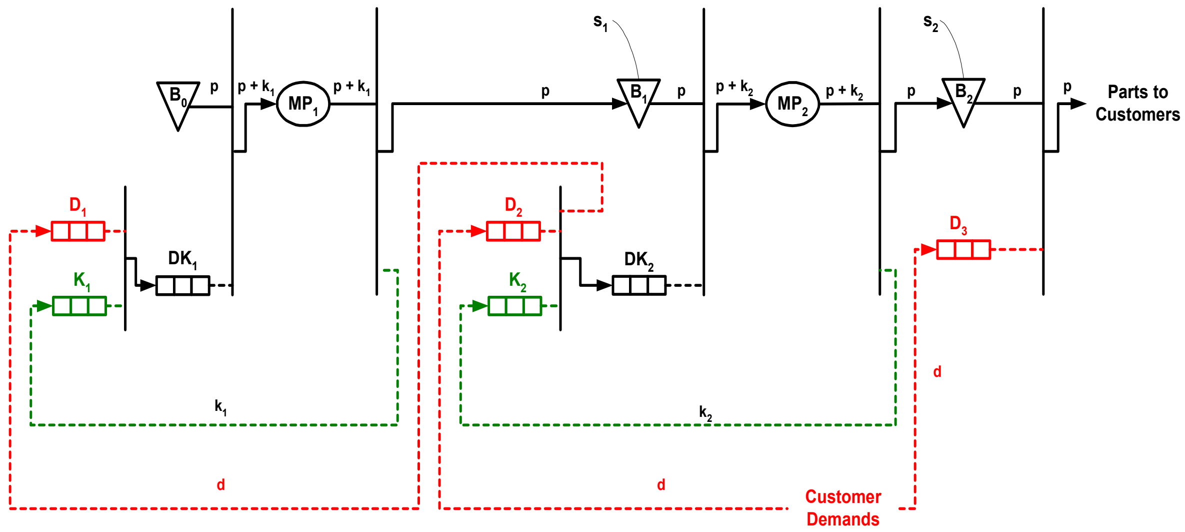

Figure 2.6: A two-stage production line controlled by Generalized Kanban Control System

(GKCS) (Buzacott, 1989)

When the first demand arrives at Queue D_3:

1. If a product is available in B_2 (which is initially the case), it is released to the customer;
2. Otherwise the demand is backordered in Queue D_3 and has to wait for a finished product to arrive in B_2.

When the second demand arrives at D_2:

1. If a stage 2 kanban is available in K_2 (which is initially the case), demand information is immediately transmitted upstream to D_1. Then a stage 2 kanban from K_2, attached together with the demand from Queue D_2, moves to Queue DK_2 to authorize production at stage 2.
   a. If a new product is available in B_1, it is instantaneously merged with a stage 2 kanban in DK_2 and the pair (both the product and the attached kanban) released into MP_2.
   b. Otherwise the kanban has to wait in Queue DK_2 for a finished product to arrive at B_1.
2. If no stage 2 kanban is available in K_2, the demand has to wait for a stage 2 kanban. This demand signal is stopped from being transmitted upstream.

Generalized Kanban Control System (GKCS) does set a limit on WIP levels in each stage. Kanbans and demand signals are coupled together, that is, the movement of demand signals are dependent on a kanban card. Both of them have to be present in Queue DK_i (an authorization queue) before parts can move downstream. Besides, an additional queue can only mean additional waiting time.

GKCS does suffer from demand information blockage. Demands can only flow upstream if there is a kanban in K_i. Also, no instantaneous transmission of demand information exists to all production stages. Lastly, GKCS doesn't respond well to long-term demand fluctuations. These factors, including its complicated structure, help explain why Generalized Kanban Control System (GKCS) has not become popular.

2.6.2. CONWIP Kanban (CK)

CONWIP Kanban (CK) was first proposed by Bonvik, Couch, and Gershwin (1997). Figure 2.7 shows a two-stage production line controlled by CK. It was proposed to leverage the advantages of both the CONWIP and the Traditional Kanban Control System (TKCS). Buffer B_0 is the components buffer, and is assumed to contain infinite number of components. Buffer B_i, where i = 1, 2, is the output buffer of stage i, where each product is attached a stage i kanban and a CONWIP card. Queue K_i contains stage i kanbans and queue C contains CONWIP cards. The kanban card movements are shown by the short-dash lines, while CONWIP card movements are shown by the long-dash lines. In its initial state, Buffer B_2 contains k_2 finished products, each product having a stage 2 kanban and a CONWIP card attached to it. Buffer B_1 contains $(C - k_2)$ finished products, each product having a stage 1 kanban and a CONWIP card attached to it. There are $[k_1 - (C - k_2)]$ free kanbans in Queue K_1. All other queues are empty. The first assumption, to prevent B_1 from having a negative number of products, is that $k_2 < C$. The second assumption (Boonlertvanich, 2005) is that the sum of all kanbans is greater than or equal to the number of CONWIPs, i.e. $\sum_{i=1}^{N} k_i \geq C$. This is to prevent the system from choking if an over-surge of demand takes place.

CONWIP Kanban (CK) operates as follows:

When a customer demand arrives at the system, it joins Queue D, requesting the release of a finished product from B2 to the customer. At that time there are two possibilities:

1. If a product is available in B2 (which is initially the case), it is released to the customer after detaching the stage 2 kanban and the CONWIP cards attached to it. This kanban is transferred upstream to K2 instantly, carrying with it a demand signal for the production of a new stage 2 part. The CONWIP card is then transferred to Queue C to authorize release of components.
2. If no product is available in B2, the demand is backordered and waits in Queue D until a new part arrives in B2.

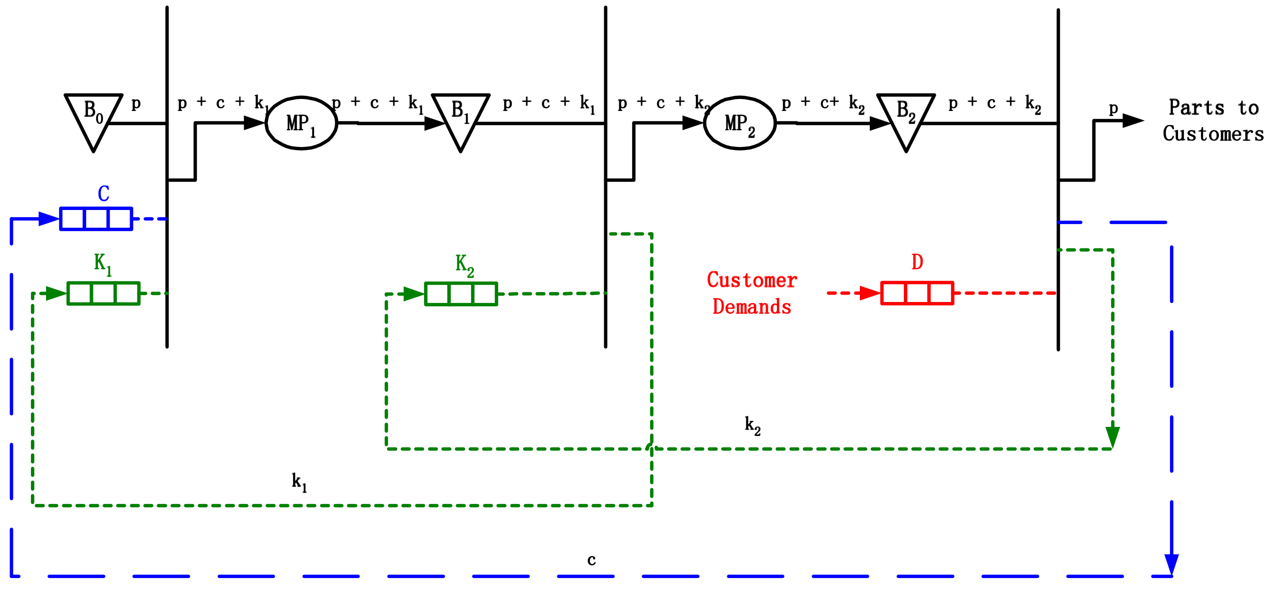

Figure 2.7: A two-stage production line controlled by CONWIP Kanban system (Bonvik et al., 1997)

As soon as a kanban card arrives in K_2, it authorizes the production of a new product in stage 2. At this time one of two things might happen:

1. If a product, with a stage 1 kanban and a CONWIP card attached is available in B_1, the stage 1 kanban is instantaneously detached and a stage 2 kanban immediately attached to it. At the same time, the pair (part and stage 2 kanban, and CONWIP card) is released to MP_2. The stage 1 kanban is transferred upstream to K_1 authorizing the release of component into stage 1.

2. If no product is available in B_1, stage 2 kanban waits in K_2 until a newly finished part arrives in B_1.

A special case occurs at the component stage. Only when a kanban card and a CONWIP card are present in Queue K_1 and Queue C respectively can components be released from B_0 into stage 1. Demand signals are transferred upstream by kanbans and transferred to the first stage by CONWIP cards. If at some stage a finished part is not available in B_i, i = 1, 2, no kanban is transferred upstream and the authorization for releasing a part downstream is temporarily stopped. It is resumed only when a part becomes available again in B_i, i = 1, 2. The advantage is that WIP is controlled for the entire production line as well as individual stages. This limits excessive inventory build-up in front of a machine if it fails. Since it has the CONWIP element, shop floor managers also get to dictate part number sequence and schedule priority jobs first, thereafter allowing the following stages to "pull" the parts downstream. With two kinds of cards, this system has lots of feedback. It depends on one parameter per stage, namely k_i, i = 1, …, N, and one additional parameter for the entire system, C.

However, having two types of cards also means more complications, since workers on the factory floor may have accidental mix ups. There is no instantaneous transmission of demand signals and the systems doesn't respond well to long-term demand fluctuations (Bonvik et al., 1997).

2.6.3. Extended CONWIP Kanban (ECK) Control System

Extended CONWIP Kanban (ECK) was first proposed by Boonlertvanich (2005). Figure 2.8 shows a two-stage production line controlled by ECK. It is the latest pull production system proposed to date, and is an improvement on the CONWIP Kanban (CK) system (2.6.2). It is a combination of Base Stock (BS), CONWIP, and Traditional Kanban Control System (TKCS), and is supposed to encapsulate the advantages of all three systems. Buffer B_0 is the raw materials buffer and is assumed to contain infinite quantity of components. Buffer B_i, where i = 1, 2, is the output buffer of stage i; each part in it is attached a CONWIP card. Queue K_i contains stage i kanbans, and Queue C contains CONWIP cards. Kanban card movements are shown by the short-dashed line, while CONWIP card movements are shown by the long-dashed line. This system depends on two parameters per stage, and an additional parameter for the entire system. They are the kanbans k_i, base stock level S_i and the total CONWIP limit C. C is the total number of CONWIP cards, representing the total WIP level allowed in the entire production line. k_i is used to limit the number of products in stage 'i', and S_i is the target number of products in the output buffer of that stage.

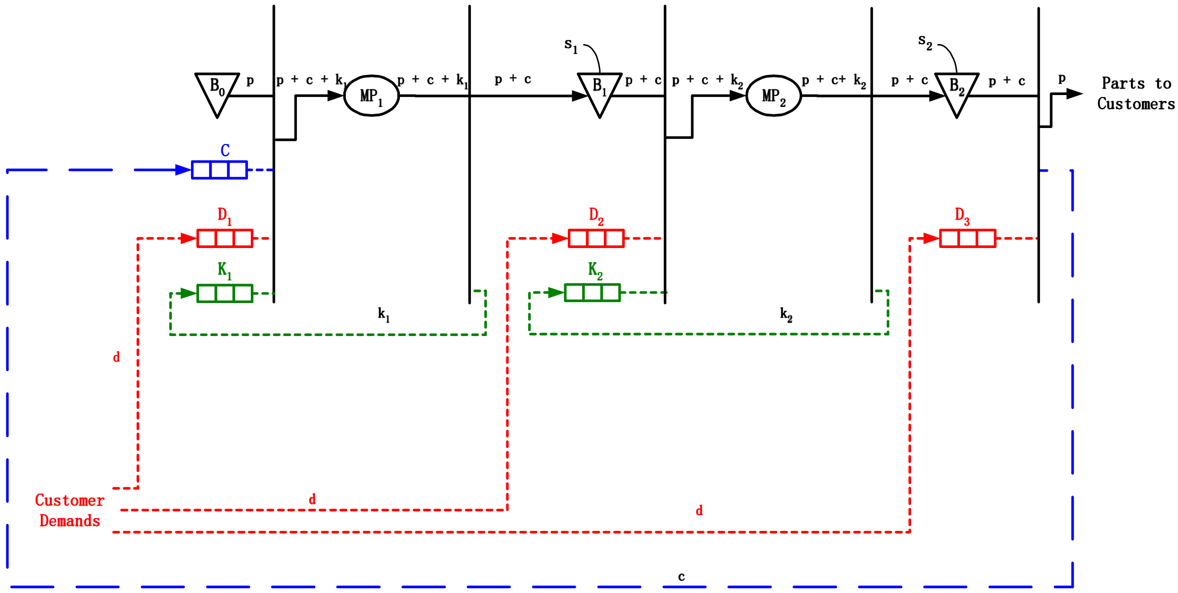

Figure 2.8: A two-stage production line controlled by Extended CONWIP Kanban (ECK) Control System (Boonlertvanich, 2005)

When the system is in its initial state, there are S_i finished products (with a CONWIP card attached to each product) stored in B_i. The remaining $C - \sum_{i=1}^{N} s_i$ CONWIP cards are stored in Queue C. There are k_i number of kanbans stored in Queue K_i. All other queues are empty. Production in the system is driven by customer demands. When a customer demand arrives at the system, the following happens:

1. The demand is split into N + 1 (=3) components, namely d_i, i = 1, 2, 3, and each component is immediately transferred to its respective Queue D_i.

2. At the very first synchronization station (the vertical line following B_0):

   a. If there is a kanban in K_1 and a CONWIP card in C (which is initially the case), a part is taken from B_0, and a kanban from K_1. A CONWIP card from C is attached to it. Then it proceeds to MP_1 for processing, and the demand in D_1 is satisfied. When the product exits MP_1, k_1 is detached and immediately transferred back to queue K_1. Thereafter, the product (with the attached CONWIP card only) proceeds to B_1 to be stored.

b. If K_1 has no kanban, or C has no CONWIP card, the demand is backlogged in D_1.

3. At the second synchronization station (the vertical line following B_1):

   a. If there is a kanban in K_2 and a product in B_1 (which is initially the case), a product is taken from B_1 and the kanban from K_2 is attached. Then it proceeds to MP_2 for processing, and the demand in D_2 is satisfied. When the product exits MP_2, K_2 is detached and immediately transferred back to queue K_2. Thereafter, the product (with the attached CONWIP card) proceeds to B_2 to be stored.

   b. If K_2 has no kanban or B_1 has no part, the demand is backlogged in D_2.

4. At the last synchronization station (the vertical line following B_2):

   a. If there is a product in B_2 (which is initially the case), it will be taken from B_2 to satisfy the demand in D_2. When the part exits the system, the CONWIP card is detached and immediately transferred back to C.

   b. If there is no product in B_2, then the demand is backlogged in D_3.

The kanban is freed earlier in ECK than in other systems, since its detached right after the part leaves a Manufacturing Process (MP). There is also an instantaneous transmission of demand. Since it has the CONWIP element, managers get to dictate part number sequence and schedule priority jobs; the following stages "pull" the parts downstream. ECK also sets a limit on WIP levels for the entire production line, while at the same time maintaining the WIP level for each stage. There are lots of feedbacks in this system. CONWIP cards feedback from the last to the first stage, while kanban cards coordinate every stage. Another advantage is that there is a target inventory level (the base stock) set at every stage's output buffer. However, the ECK is more complicated than more traditional systems not only because of its structure, but also because there are two types of cards to handle. Lastly, it doesn't respond well to long-term demand fluctuations (Boonlertvanich, 2005).

2.7. Other Pull Systems

In this section, a short introduction of other different types of Pull Systems will be presented. Since they are of a different class of kanban systems and not directly related to this research, they will only be mentioned briefly.

The Flexible Kanban System (FKS) was first introduced by S. M. Gupta, Al-Turki, and Perry (1999). They introduce this new system to cope with uncertainties and planned/unplanned interruptions. They demonstrate FKS's superiority by conducting four case examples which covered various uncertainties. After comparing the FKS's performance with the traditional JIT system, they claim that in all the cases considered the performance of their FKS was superior.

The Adaptive Kanban System (AKS) was first introduced by Tardif and Maaseidvaag (2001). They introduce a new adaptive kanban-type pull control mechanism which determines when to release or reorder raw parts based on customer demands. They claim that their system differs from the traditional kanban system in that the number of kanban cards is allowed to change with respect to the inventory and backorder levels. However, the number of cards in the system remains limited, restricting the amount of Work-In-Process (WIP) in the system. They show that their adaptive system can outperform the traditional kanban pull control mechanism and is easy to implement.

The Reactive Kanban System (RKS) was proposed by Katsuhiko Takahashi, Morikawa, and Nakamura (2004). Their paper proposed a reactive Just-In-Time (JIT) ordering system for multi-stage production systems with unstable changes in demand. They propose a reactive controller of the buffer size which can detect unstable changes in the mean and variance of demand. It uses exponentially weighted moving average charts for detection. They place numerous detection points at each inventory buffer to detect these unstable changes. The performance of their RKS is finally analysed using simulation experiments.

Jou Lin, Frank Chen, and Min Chen (2013) propose a Knowledge Kanban (KK) system to enhance knowledge flow for a virtual Research and Development (R&D) process. The idea is to employ the kanban philosophy into R&D firms for quicker and easier access to knowledge. They claim that their proposed system helps employees of these R&D firms reduce the cycle time of their work. First, they create a Virtual Enterprise (VE); then they design a KK model to custom fit it. Finally, in their study, they claim that KK system is an effective tool to facilitate knowledge creation, storage, transmission and sharing for R&D firms.

The latest E-Kanban paper is documented by Al-Hawari and Aqlan (2012). They develop a software application for an E-Kanban inventory control system, developed to track Work-In-Process (WIP) inventory and finished goods in an aluminium factory. They claim that after the current paper system is replaced with their E-Kanban system, data entry errors are minimized. Furthermore, their results show that manufacturing lead time and WIP is reduced by an average of 88% and 50%, respectively. They also build an accountability measure into their system to identify errors. Their system can generate reports about order information, aiding managers to make decisions based on real-time information.

Aghajani, Keramati, and Javadi (2012) study a cellular manufacturing system controlled by Kanban. In their model, they includ the possibility of defective items produced, and rework is carried out. They use a mixed-integer nonlinear programming (MINLP) model to minimize total cost. Thereafter, their total cost model is used to determine the optimal number of kanbans and batch size. They also use Particle Swarm Optimization (PSO) and Simulated Annealing (SA) algorithms to reduce the computational time for solving large MINLPs. They show that both PSO and SA result in a near optimal solution but the PSO algorithm gives a better performance than the SA method.

Al-Tahat, Dalalah, and Barghash (2012) study how to synchronize the flow of materials in a kanban controlled serial production line. Their production line is described as a queuing network; they then make use of a Dynamic Programming (DP) algorithm to solve it by decomposing it into several numbers of single-stage sub-production lines. A performance measure is then developed to determine and compare production parameters. Thereafter, they validate their results using a discrete event simulator called Pro-Model. They discover that their performance measure has a very small error. Thus, they claim that their method is effective in synchronizing inventory with customer demands.

2.8. Summary of Push-Pull Production Systems

This chapter introduces different Push-Pull systems found in the kanban literature. They are classified in two main categories: Pull, Push and Hybrid Pull. Base Stock (BS) falls under the Push category. CONstant Work–In–Process (CONWIP) and Traditional Kanban Control System (TKCS) fall under the Pull category, whereas CONWIP Kanban (CK), Extended CONWIP Kanban (ECK), Generalized Kanban Control System (GKCS) and Extended Kanban Control System (EKCS) fall under the Hybrid Pull category. The operating procedures, schematics, advantages and disadvantages for each system are presented, except for Extended Kanban Control System (EKCS), which is discussed in detail in Chapter 3

Altogether, there are seven prominent Push-Pull mechanisms in the kanban literature. Table 2.1 summarizes the characteristics of the various systems excluding CONWIP, while Table 2.2 summarizes the characteristics of systems including CONWIP.

Table 2. 1: Characteristics of Push-Pull Systems excluding CONWIP

Characteristic	Base Stock (BS)	Traditional Kanban Control System (TKCS)	Generalized Kanban Control System (GKCS)	Extended Kanban Control System (EKCS)
Sets a target inventory level (base stock) in the output buffer	Yes	No	Yes	Yes
Instantaneous transmission of demand	Yes	No	No	Yes
Feedback using kanbans?	No	Yes	Yes	Yes
Is WIP controlled at every stage?	No	Yes	Yes	Yes
Is WIP controlled for the entire production line?	No	No	No	No
Is the kanban separate from demand signal?	N.A.	No, except in the first demand queue	Yes	Yes
Are kanbans released immediately after Manufacturing Process (MP)?	N.A.	No	Yes	No

Table 2. 2: Characteristics of Hybrid Pull Systems with CONWIP

Characteristic	CONstant Work – In Process (CONWIP)	CONWIP Kanban (CK)	Extended CONWIP Kanban (ECK)
Sets a target inventory level (base stock) in the output buffer	No	No	Yes
Instantaneous transmission of demands	No	No	Yes
Any feedback system?	Yes, using CONWIP cards	Yes, using both Kanbans and CONWIP cards	Yes, using both Kanbans and CONWIP cards
Is WIP controlled at every stage?	No	Yes	Yes
Is WIP controlled for the entire production line?	Yes	Yes	Yes
Is CONWIP card separate from demand signal?	No	No	Yes
Is kanban separate from demand signal?	N.A.	No	Yes
Are kanbans released immediately after Manufacturing Process (MP)?	N.A.	No	Yes

In Chapter 3, the operation, gaps involved in current research, and a method to optimize the EKCS are presented. The two main research gaps of EKCS are insufficient study of its performance and a lack of optimization concepts. The most notable contribution of chapter 3 is the method that can help optimize EKCS.

CHAPTER 3

THE EXTENDED KANBAN CONTROL SYSTEM

The Extended Kanban Control System (EKCS), first proposed by Dallery (2000), has great potential to outperform both BS and TKCS. The objective of this research is to investigate EKCS's performance vis-à-vis other control systems, and to showcase where it could make factory shop floors leaner and more efficient. Section 3.1 describes EKCS's operation; section 3.2 highlights its research gaps, and section 3.3 proposes a method to optimize EKCS.

3.1 EKCS: Operations

Figure 3.1 shows a two-stage production line controlled by EKCS. Buffer B_i is the output buffer of stage i and contains stage i finished parts and kanbans. Buffer B_0 is the components buffer and is assumed to contain infinite number of components. Queue D_i contains demands for stage i parts; Queue D_3 is the customer demands buffer (in a 2 stage KCS) and Queue K_i contains stage 'i' kanbans. EKCS depends on two parameters per stage: number of kanbans, k_i, and base stock level, S_i. In the initial state, B_i contains S_i finished parts with stage 'i' kanbans attached to them, and Queue K_i contains (k_i - S_i) free stage 'i' kanbans. All the other queues are empty.

EKCS operates as follows:

1. When a customer demand arrives at the system, it is instantaneously split into N + 1 demands (equal to three in the two-stage system).
2. The first demand joins Queue D_3, requesting the release of a finished product from buffer B_2 to the customer.

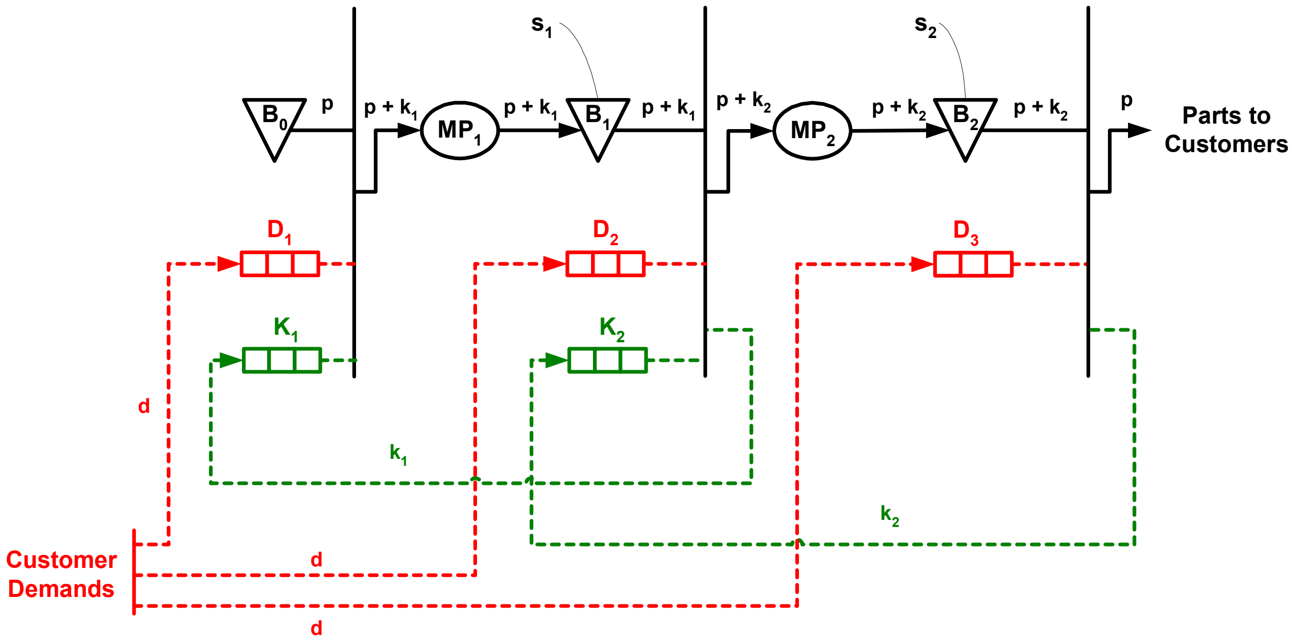

Figure 3. 1: A two-stage production line controlled by EKCS (Dallery, 2000)

a. If a product is available in B_2, it is released to the customer after detaching the stage 2 kanban. This kanban is then transferred upstream to queue K_2.

b. Otherwise, the demand is backordered.

3. The other N (=2) demands join the input demand Queue D_i of each stage i, i = 1, 2.

a. If a product is present, attached with a stage (i-1) kanban in B_{i-1} and a stage i kanban in Queue K_i, the stage (i-1) kanban is immediately detached from the product and transferred upstream to K_{i-1}. At the same time, the stage i kanban is removed from queue K_i and attached to the product. The pair is then released for processing into MP_i.

b. If there is either no product in B_{i-1} or no kanban in K_i, the demand is backordered and has to wait in queue D_i.

In EKCS, there is an instantaneous transmission of demand. When a demand arrives at the system, it is immediately broadcast to every stage in the system. This implies that each stage in the system knows immediately the need for production of a new part in order to replenish the finished-product buffer. Another advantage of EKCS is the decoupling of kanbans and demand signals. This means that a demand signal moves independently of a kanban, and can be released earlier to upstream stages. In addition, EKCS has the Base Stock (BS) element; each stage of production sets a target level of inventory in its output buffer.

But how does the BS element affect EKCS specifically? Firstly, referring to Figure 3.1, BS affects EKCS most with the parameter, s_1, which is the initial base stock level for Output Buffer B_1. However, setting this level will clash with its TKCS element, which is also trying to control the WIP by having an optimal k_1, the number of kanbans. Hence to have a distinctive separation, the Base Stock (BS) element will calculate the optimal number of base stock, s_1*, for the EKCS; whereas the TKCS element will calculate the optimal number of un-dispatched kanbans, k_1*, in the kanban queue K_1. The total initial WIP level will then be set to (s_1* + k_1*). Without the BS element, EKCS becomes the TKCS, where the total WIP is simply k_1*.

Secondly, referring to Figure 3.1 again, Queue D_1, supplied by the BS element, is where the major difference between EKCS and TKCS sets in. Without this Queue D_1, EKCS would have fallen back to TKCS. But with Queue D_1, TKCS now has become EKCS. But how does this affect EKCS? Now with both Queues D_1 and K_1 together, WIP entry into MP_1 is limited. That is, previously without Queue D_1, once a kanban arrives at K_1, the part gets shipped into MP_1 immediately. Now, with queue D_1, it ensures that a part is taken out of B_0 only if a real demand is present.

A disadvantage of EKCS is that kanbans are freed later in EKCS. A kanban is detached only after it proceeds out of the output buffer. Lastly, the EKCS doesn't respond well to long-term demand fluctuations (Dallery, 2000).

3.2 Research Gaps

Advantages of EKCS over well-established control systems such as BS and TKCS are not well addressed in the literature. Only Boonlertvanich (2005) has discussed EKCS' superior performance, albeit qualitatively. Thus, to date the performance of EKCS has not been analysed quantitatively against KPIs such as:

- Fill rate, or number of backorders;
- Average inventory (the more inventory held in the system, the less "lean" it is); and

- Average waiting time (turn-around-time, TAT) for a customer.

The next two sections 3.2.1 and 3.2.2 present relevant literature that identifies the research gaps, and justify the need for this research.

3.2.1 Research Gap 1: Lack of proof of superior performance of EKCS

Karaesmen and Dallery (2000) stated that "although the two parameters per stage mechanisms such as generalized or extended kanban offer potential improvements over single parameter mechanisms, it is not obvious how these improvements translate into cost savings and whether or not it is worth investing in a more complex mechanism." To test this, they use an optimal control framework to study BS, TKCS and GKCS. Figure 3.2 shows the model of a two-stage production line, as modelled by Karaesmen and Dallery (2000). Demands arrive according to a Poisson process with rate λ. The circled numbers 1 and 2 represent MPs for stages 1 and 2, respectively, and they both have exponentially distributed service times with rate μ_i (i=1,2). X_1 denotes the number of parts in the output buffer of stage 1 and the part that is currently in production in the second MP; X_2 denotes the number of parts in the output buffer of stage 2.

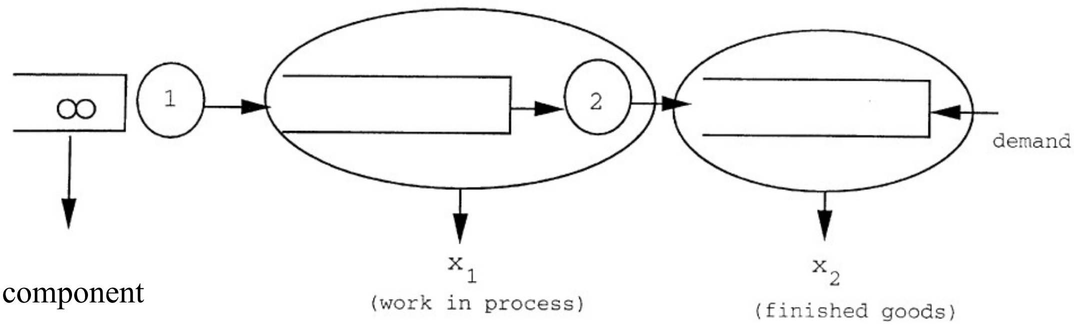

Figure 3. 2: The two-stage production line used by Karaesmen and Dallery (2000)

A major deficiency of this model is that it does not use the conventional approach to modelling a two-stage production line. Normally, X_i refers to the number of parts held in stage 'i'. For example, X_1 would denote WIP of the first MP and first output buffer, while X_2 would denote WIP of the second MP and second output buffer. Their approach makes it difficult to account for individual inventory levels in separate output buffers.

Karaesmen and Dallery (2000) use a theoretical method called “optimal switching curves” (Veatch & Wein, 1994) to determine optimal parameters such as base stock, S*, and kanban number, K* for individual stages. Also, Karaesmen and Dallery (2000) stop short of proving the superiority of EKCS, claiming that EKCS is a special case of GKCS, and propose that a comparison of GKCS to BS and TKCS is sufficient to show EKCS’ superiority. However, the scenarios (of comparison) are not clearly highlighted, and the performance comparison does not use standard KPIs.

Korugan and Çadırcı (2008) offered the latest study of pull policies’ performance to date, comparing the four most common pull systems: BS, TKCS, GKCS and EKCS. They use Markov Chains to model the four policies and minimize cost based on the control parameters of each control mechanism. They concluded that hybrid pull systems such as GKCS and EKCS demonstrate better performance than simple systems such as BS and TKCS. However, they do not show how EKCS outperforms the rest based on KPIs, such as fill rate and average customer waiting time. Also, they model a non-standard tandem process line. They use a remanufacturing process in addition to MP, which makes their analysis more complex and not readily applicable to general manufacturing systems.

Khuller (2006) uses simulation to compare two kanban systems in different environments: TKCS and Extended Information Kanban Control System (EiKCS). However, instead of using the standard EKCS, he uses a modified EKCS with its Base Stock level set to the maximum WIP capacity. Hence, even though he obtains positive results for EiKCS, his comparison fails to establish the superiority of traditional EKCS.

Deokar (2004) also uses simulation to compare the TKCS, GKCS and EKCS. However, she does not analyse them in a Single Product (SP) scenario, and assumes, instead, that all systems handle Multiple Products (MP). This comparison is not clearly defined, as she does not specify whether kanbans are dedicated or shared. Finally, the paper fails to clarify exactly how and why EKCS is superior to TKCS and GKCS.

The above literature review shows that there has been insufficient analysis on EKCS’s capabilities. A clearly defined and properly conducted investigation to explore the superiority of EKCS is thus needed to establish its advantages, especially for shop floor managers.

3.2.2 Research Gap 2: A need to optimize the EKCS

Dallery (2000), in their seminal paper, proposing the EKCS, mention the need for an optimization model. They claim that this system can only work if its parameters, base stock, S*, and kanban number, K*, are obtained optimally. The only researcher who attempts to optimize this system is Selvaraj (2009). He models a three-stage EKCS using ARENA and varies the customer arrival rate. Then, he obtains the optimal kanban number, K* for each scenario, through trial-and-error. This method requires tedious simulation for each scenario. Furthermore, the author does not explore ways to obtain the optimal base stock, S*, which is a critical component of the EKCS.

3.3 EKCS optimization

Existing literature lacks performance studies of EKCS. It is not yet proven whether EKCS outperforms traditional systems such as BS and TKCS. However, before a fair comparison can be made, EKCS first needs to be optimized.

Optimizing BS and TKCS is relatively easy, as they both rely on only one parameter that defines the system (either optimal base stock level, S*, or number of kanban, K*), EKCS, on the other hand, has two parameters. Optimizing two parameters simultaneously increases the complexity significantly.

This section proposes a model for determining the optimal base stock, S*, and number of kanban, K*, for a SS/SP/EKCS. It is based on Expected Total Cost (ETC), which comprises total holding and shortage costs. Optimal base stock, S*, and number of kanban, K* are obtained by minimizing the ETC. This algorithm is coded in MATLAB (Appendix I1) and tested on a sample system.

3.3.1 Expected Total Cost for Single Stage, Single Product, Extended Kanban Control System (SS/SP/EKCS)

Equation (3.1) shows the Expected Total Cost (ETC (S, K)) for a SS/SP/EKCS, which is the sum of expected inventory holding and shortage costs.

$$\begin{aligned} ETC(S,K) &= \text{ Inventory Holding Cost + Shortage Cost} \\ &= C_h\left(I\left[S,K\right]\right) + C_s\left(E\left\{t_{stockout}\right\}\right) \end{aligned} \tag{3.1}$$

Where

S: Base Stock level, or long run average number of finished parts in output buffer, B_1,

K: Number of un-dispatched kanbans in queue, K_1,

C_h: Holding cost per unit per unit of time,

C_s: Shortage cost per unit per unit of time,

I [S, K]: Expected Inventory Level, dependent on two parameters, S and K, and

$E\{t_{stockout}\}$: Expected duration of a stock out in output buffer, B_1.

The Manufacturing Process (MP) rate is assumed to be exponentially distributed, whereas the demand arrival rate follows the Poisson process.

3.3.2 Expected Inventory Level

Expected Inventory Level, I[S, K], represents the expected number of parts in Manufacturing Process (MP) as well as in the output buffer, over the long run. Figure 3.3 shows the location of I[S, K] and E [K], the expected number of undispatched kanbans in kanban queue, K_1, in a SS/SP/EKCS.

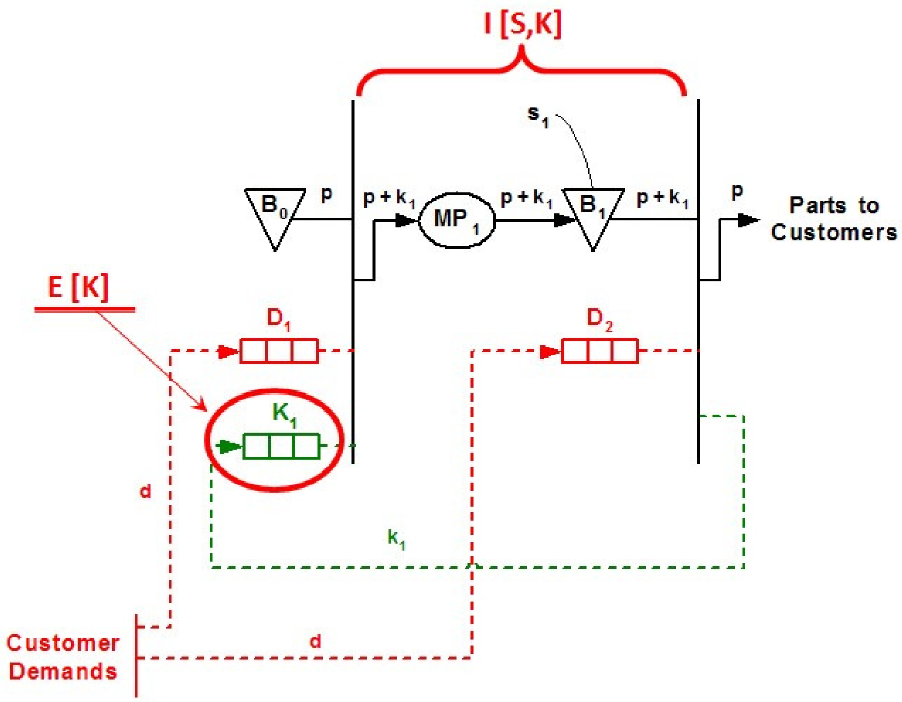

Figure 3. 3: Location of expected inventory level, I[S,K], and expected undispatched kanbans, E[K], in a SS/SP/EKCS

In EKCS, kanbans can only be present in three places: the undispatched kanban queue, K; in MP; or attached to stock in the output buffer, B. Thus, Expected Inventory Level, I[S, K] can be defined as the difference between total WIP (S+K) and expected number of undispatched kanbans in the queue, K_1, E[K]. This is represented in Equation 3.2.

$$I[S, K] = (S+K) - E[K_1] \quad (3.2)$$

Where

S+K: is the total work-in-process in EKCS,

I[S, K]: is the expected inventory level, and

$$E[K_1] \approx K - \frac{\rho(K+S)}{K+S+1-\rho(K+S)} + \frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]}$$ (Proof in Appendix A).

3.3.3 Expected Stock Out Time, E{$t_{stockout}$}

E{$t_{stockout}$} represents the expected stock out time in the output buffer, B_1, of EKCS. Since shortage cost, C_s, is defined to be per unit time, C_s is multiplied with the average amount of time, E{$t_{stockout}$}, to compute the total shortage cost. This method has been used in Wang and Hsu-Pin (1991) and Nori and Sarker (1998). E {$t_{stockout}$} is denoted by π_0, which is defined in Markov Chains as the long-run average probability of time spent in state 0 stock out mode. Figure 3.3 shows the location of E{$t_{stockout}$}. The derivation can be found in Appendix B.

$$\begin{aligned} E\{t_{stockout}\} &= \pi_0 \\ &= \left(\frac{\rho - 1}{\rho^{S+K+1} - 1}\right) \end{aligned} \tag{3.3}$$

Where

$$\rho = \frac{\lambda}{\mu}$$

μ is the MP rate, while λ is the demand arrival rate, both distributed exponentially.

3.3.4 Optimization of Expected Total Cost, ETC (S, K)

From Equations (3.1), (3.2) and (3.3), ETC (S, K) can be written as

$$ETC(S,K) = C_h\left(S + \frac{\rho(K+S)}{K+S+1-\rho(K+S)} + \frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]}\right) + C_s\left(\frac{\rho-1}{\rho^{S+K+1}-1}\right) \tag{3.4}$$

ETC is convex in the total number of kanbans, S+K. As S+K increases, resulting in more kanbans in the system, more WIP is allowed into the system. With more WIP, inventory increases and total holding cost increases. However, since (S+K) is in the denominator of Equation (3.4), total shortage cost decreases as well, as more inventory implies fewer stock outs. If (S+K) decreases, fewer kanbans restrict WIP into the system; total holding cost decreases while shortage cost increases. So the question becomes finding the optimal level of S and K to minimize the ETC.

Since S and K are discrete variables, minimization of ETC cannot be performed using differentiation. An alternate method is to carry out exhaustive enumeration over different values of S and K. However, such a search would be tedious in the absence of well-defined bounds for S and K. In this research, an incremental method is used to ascertain ranges for S and K. These are then used to bound the search, quickly obtaining the optimum base stock and kanban number, (S*, K*). The basic queuing theory assumption of $\rho \neq 1$ still holds.

3.3.5 Optimal Base Stock (S*) Range

Since ETC (S*, K) is the minimum possible ETC with S*, Equations (3.5) and (3.6) hold.

$$ETC\left[S^*-1,\ K\right]-ETC\left[S^*,\ K\right]\geq 0 \tag{3.5}$$

$$ETC\left[S^*+1,\ K\right]-ETC\left[S^*,\ K\right]\geq 0 \tag{3.6}$$

Where

$$ETC\left[S-1,\ K\right]=C_h\left(S-1+K-E[K]\right)+C_s\left(\frac{\rho-1}{\rho^{S+K}-1}\right) \tag{3.7}$$

$$ETC\left[S,\ K\right]=C_h\left(S+K-E[K]\right)+C_s\left(\frac{\rho-1}{\rho^{S+K+1}-1}\right) \tag{3.8}$$

$$ETC\left[S+1,\ K\right]=C_h\left(S+1+K-E[K]\right)+C_s\left(\frac{\rho-1}{\rho^{S+K+2}-1}\right) \tag{3.9}$$

Solving for Equation (3.5) leads to

$$\frac{\rho^{S+K}}{\left(\rho^{S+K}-1\right)\left(\rho^{S+K+1}-1\right)}\geq\frac{C_h}{C_s\left(\rho-1\right)^2} \tag{3.10}$$

Solving for Equation (3.6) leads to

$$\frac{\rho^{S+K+1}}{\left(\rho^{S+K+2}-1\right)\left(\rho^{S+K+1}-1\right)} \le \frac{C_h}{C_s\left(\rho-1\right)^2} \tag{3.11}$$

Thus, the range of S* is

$$\frac{\rho^{S^*+K+1}}{\left(\rho^{S^*+K+2}-1\right)\left(\rho^{S^*+K+1}-1\right)} \le \frac{C_h}{C_s\left(\rho-1\right)^2} \le \frac{\rho^{S^*+K}}{\left(\rho^{S^*+K}-1\right)\left(\rho^{S^*+K+1}-1\right)} \tag{3.12}$$

3.3.6 Optimal Number of Kanban (K*)

Since ETC (S, K*) is the minimum possible Expected Total Cost with an optimal kanban number, K*, Equations (3.13) and (3.14) hold.

$$ETC\left[S,\ K^*-1\right]-ETC\left[S,\ K^*\right]\ge 0 \tag{3.13}$$

$$ETC\left[S,\ K^*+1\right]-ETC\left[S,\ K^*\right]\ge 0 \tag{3.14}$$

Where $$ETC\left[S,\ K-1\right]=C_h\left(S+K-1-E[K-1]\right)+C_s\left(\frac{\rho-1}{\rho^{S+K}-1}\right) \tag{3.15}$$

$$ETC\left[S,\ K\right]=C_h\left(S+K-E[K]\right)+C_s\left(\frac{\rho-1}{\rho^{S+K+1}-1}\right) \tag{3.16}$$

$$ETC\left[S,\ K+1\right]=C_h\left(S+K+1-E[K+1]\right)+C_s\left(\frac{\rho-1}{\rho^{S+K+2}-1}\right) \tag{3.17}$$

Solving for Equation (3.13) leads to

$$\frac{\rho^{S+K}}{\left(\rho^{S+K}-1\right)\left(\rho^{S+K+1}-1\right)} \ge \frac{C_h}{2C_s\left(\rho-1\right)^2} \tag{3.18}$$

Solving for Equation (3.14) leads to

$$\frac{\rho^{S+K+1}}{\left(\rho^{S+K+2}-1\right)\left(\rho^{S+K+1}-1\right)} \le \frac{C_h}{2C_s\left(\rho-1\right)^2} \tag{3.19}$$

Hence the range of optimal kanbans K* is

$$\frac{\rho^{S+K^*+1}}{\left(\rho^{S+K^*+2}-1\right)\left(\rho^{S+K^*+1}-1\right)} \leq \frac{C_h}{2C_s\left(\rho-1\right)^2} \leq \frac{\rho^{S+K^*}}{\left(\rho^{S+K^*}-1\right)\left(\rho^{S+K^*+1}-1\right)}$$

(3.20)

Let $y = \rho^{S+K^*}$ and $C = \dfrac{C_h}{2C_s\left(\rho-1\right)^2}$

Then

$$\frac{\rho y}{\left(\rho^2 y-1\right)\left(\rho y-1\right)} \leq C \tag{3.21}$$

And

$$\frac{y}{\left(y-1\right)\left(\rho y-1\right)} \geq C \tag{3.22}$$

From (3.21), since 0 < p <1 and S+K* ≥ 0

$$\frac{\rho y}{\left(\rho^2 y-1\right)\left(\rho y-1\right)} \leq C$$

$$\rho y \leq C\left(\rho^2 y-1\right)\left(\rho y-1\right) \tag{3.23}$$

$$C\rho^3 y^2 - \left(C\rho^2 + C\rho + \rho\right)y + C \geq 0$$

From (2), likewise,

$$\frac{y}{\left(y-1\right)\left(\rho y-1\right)} \geq C$$

$$y \geq C\left(y-1\right)\left(\rho y-1\right)$$

$$0 \geq C\left[\rho y^2 - \rho y - y + 1\right] - y \tag{3.24}$$

$$0 \geq C\rho y^2 - C\rho y - Cy + C - y$$

$$0 \geq C\rho y^2 - y\left[C\rho y + C + 1\right] + C$$

3.3.7 Numerical example

Once the bounds on the parameters have been established, an exhaustive search over both the ranges can be performed to obtain the optimal S* and K*. An example from Wang and Hsu-Pin (1991) and Nori and Sarker (1998) is used below. A MATLAB program (Appendix I1) is written to help solve for S* and K*.

Given μ = 50 units per time, and λ = 40 units per time

$$\rho = \frac{40}{50} = 0.8$$

C_s, = \$200 / time, and C_h, = \$20 / unit / time

The range of optimal base stock, S*, becomes

$$\frac{0.8^{S^*+K+1}}{\left(0.8^{S^*+K+2}-1\right)\left(0.8^{S^*+K+1}-1\right)} \leq \frac{20}{200\left(0.8-1\right)^2} \leq \frac{0.8^{S^*+K}}{\left(0.8^{S^*+K}-1\right)\left(0.8^{S^*+K+1}-1\right)}$$

Using MATLAB to solve for this range leads to

$$1 \leq (S^*+K) \leq 3$$

The range of optimal kanban number, K*, becomes

$$\frac{0.8^{S+K^*+1}}{\left(0.8^{S+K^*+2}-1\right)\left(0.8^{S+K^*+1}-1\right)} \leq \frac{20}{2\left(200\left(0.8-1\right)^2\right)} \leq \frac{0.8^{S+K^*}}{\left(0.8^{S+K^*}-1\right)\left(0.8^{S+K^*+1}-1\right)}$$

Using MATLAB to solve for this range

$$1 \leq (S+K^*) \leq 3$$

Since both S* and K* lie in the same range, $1 \leq (S^*+K^*) \leq 3$.

Likewise, using the closed form solution from Equation (3.23)

$$(2.5)(0.8)^3 y^2 - \left[(2.5)(0.8)^2 + (2.5)(0.8) + 0.8\right] y + 2.5 \geq 0$$

$$1.28y^2 - [1.6 + 2 + 0.8] y + 2.5 \geq 0$$

$$1.28y^2 - 4.4y + 2.5 \geq 0$$

$$(y - 2.72)(y - 0.72) \geq 0$$

Therefore

$$S + K^* \geq 4.48$$
$$or$$
$$S + K^* \geq 1.47$$

Using the closed form solution from Equation (3.24)

$$0 \geq (2.5)(0.8) y^2 - y\left[(2.5)(0.8) + 2.5 + 1\right] + 2.5$$

$$0 \geq 2y^2 - 5.5y + 2.5$$

$$0 \geq (y - 2.18)(y - 0.58)$$

Therefore

$$S + K^* \leq -3.5 \quad (NA)$$
$$or$$
$$S + K^* \leq 2.48$$

The range becomes 1.47 ≤ S+K* ≤ 2.48. However, since K is an integer, the range is 1 ≤ S+K* ≤ 3.This answer is exactly the same as the one gotten earlier. This range is used in the MATLAB program and after an exhaustive search, the optimal S* and K* are obtained:

Table 3. 1: Enumeration over range of S & K for SS/SP/EKCS

S	K	S+K	ETC(S,K)
0	1	1	$98.89

0	2	2	\$72.46
0	3	3	\$64.69
1	0	1	\$108.89
1	1	2	\$82.46
1	2	3	\$74.69
2	0	2	\$92.46
2	1	3	\$84.69
3	0	3	\$94.69

The optimal base stock, S* is 0, optimal kanban number, K* is 3 and lowest ETC(S*, K*) is found to be \$64.69. A plot of ETC vs. base stock and kanban number is shown in Figure 3.5. As can be seen, the lowest ETC is exactly at the point S* = 0 and K* = 3.

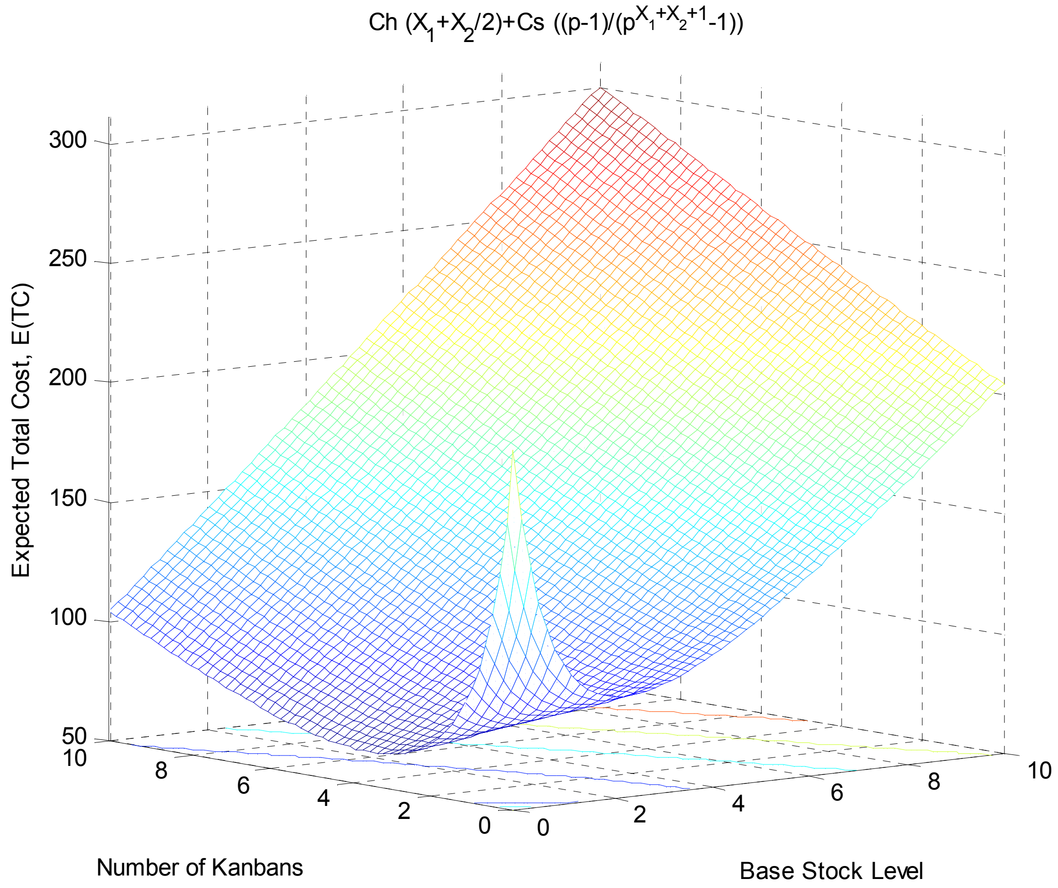

Figure 3. 4: Expected Total Cost (ETC) vs. base stock, S, and number of kanban, K

3.4 Summary

This chapter presented the operation of EKCS, research gaps in evaluating its performance and a model for optimizing the system. The first research gap is inadequate studies of EKCS' performance, which is addressed in the next chapter. The second research gap is the need to optimize the EKCS system by computing its optimal base stock level and number of kanbans. The optimization procedure presented in this chapter is based on minimizing the expected total cost; an exhaustive search is performed in the bounded region of optimal base stock, S*, and optimal kanban number, K*, to obtain the lowest cost. The ETC is comprised of total holding and shortage costs, which can be computed once the system is modelled as a Markov Chain. The algorithm is coded in MATLAB (Appendix I1) and tested on an example.

The next chapter presents a simulation performance comparison of three SS–SP control systems: BS, TKCS and EKCS. Performance comparisons in terms of

KPIs are done to compare the performance of EKCS against BS and TKCS over a range of system parameters.

CHAPTER 4

PERFORMANCE COMPARISON OF SINGLE STAGE, SINGLE PRODUCT KANBAN CONTROL SYSTEMS (SS/SP/KCS)

This chapter presents a performance comparison of single-stage, single-product kanban control systems (SS/SP/KCS), namely EKCS, TKCS and BS. Firstly, optimization models for BS and TKCS are developed. These optimization models are then used to find the optimal base stock, S*, and optimal number of kanban K*, for the systems. Then, three scenarios with different simulation parameters are set up to compare the performance of the systems. Simulation assumptions are listed in Section 4.2.1. Finally, the simulation results and insights gained are presented in Section 4.3.

A Monte Carlo simulation is necessary in this research as it acts as a common platform for all three systems, the BS, TKCS and EKCS, to be compared fairly. It is not possible to use their closed form solution to do a performance comparison as each of them uses a different set of parameters which makes the result biased. The reason for the usage of different parameters (in different KCS) can be found in Ang (2014). Hence simulation is conducted for the reason of a fair comparison of the three systems, using a standard cost function, called Actual Total Cost (ATC).

In comparison, the closed form solution uses the Expected Total Cost (ETC), which consists of:

- BS: Holding and backorder cost
- TKCS: Holding and shortage cost
- EKCS: Holding and shortage cost

In simulation, performance of BS, TKCS and EKCS is measured by Actual Total Cost (ATC) which consists of Total Backorder Cost, Total Shortage Cost and Total Holding Cost (Equation 4.1). Kindly see Section 4.2 for their definition. Since the three systems are compared using the same ATC, and the parameters are obtained from ARENA, it becomes a fairer way to compare their performance.

4.1 Optimization of Single Stage, Single Product Base Stock and Traditional Kanban Control Systems

Chapter 3 presented a model for obtaining optimal base stock, S*, and optimal number of kanbans, K*, in a SS/SP/EKCS. This section briefly discusses models to optimize BS and TKCS, both of which also assume Single Stage and Single Product (SS/SP) for consistency.

The most popular method to optimize BS was proposed by Zipkin (2000), later simplified by Wallace J. Hopp and Spearman (2008). It is based on Expected Total Cost (ETC), comprised of total holding cost and backorder cost. Optimization of each of component leads to a Cumulative Distribution Function (CDF), G(D), of demand during replenishment lead time, which is shown to be equal to the ratio of the individual backorder cost (C_b \$ per unit) and the sum of holding (C_h \$ per unit) and backorder costs. With the assumption that $C_b \geq C_h$ and G(S*) is Poisson distributed, POISSONINV ([C_b / (C_b + C_h)], D) is used in MATLAB to obtain S*. In this research, a MATLAB program has been written to obtain S* for BS, following Zipkin (2000) and Wallace J. Hopp and Spearman (2008) approach (Appendix I2).

Many authors have proposed different techniques, mostly based on Markov Chains, to find K* for TKCS. One proposed by Nori and Sarker (1998) is presented, which considers an ETC of total holding and shortage costs. The model is based on Markov Chains, and the state space is fixed at the output buffer B_1. Demand arrivals follow a Poisson process, and exponential processing times are assumed at the MP. In all Markov Chain-based methods, the most tedious and difficult part is obtaining the steady state probabilities of different states; this typically requires many cross substitutions, arising from simultaneous equations. However, Nori and Sarker (1998) cleverly devise a coefficient matrix, S, using standard techniques in stochastic processes (Ramakumar, 1993) from the rate of departures matrix, R. Then they use induction to generalize equations to obtain an expression for the ETC. In order to speed up the search process, they ascertain bounds for K. The algorithm proposed by Nori and Sarker (1998) to obtain, K* for TKCS has been coded in MATLAB for comparison purposes.

4.2 Simulation Parameters

Simulation experiments are conducted under three scenarios (Table 4.1): low, medium and high backorder and shortage costs, C_b and C_s. Their ratio to the holding cost, C_h, is kept constant. Manufacturing Process (MP) rate is also constant, but the demand arrival rate is varied. The results of the simulation runs are listed in Appendix C.

Table 4.1: Simulation parameters for SS/SP/KCS

	Scenario 1 (low)	Scenario 2 (medium, 20X)	Scenario 3 (high, 200X)
Holding Cost, C_h (per unit per day)	\$10	\$10	\$10
Backorder Cost, C_b (per unit)	\$20	\$200	\$2,000
Shortage Cost, C_s (per day)	\$20	\$200	\$2,000
Manufacturing Process, MP, rate (per day)	20		
Demand arrival rates (units per day)	10, 12, 14, 16, 18		

Before each scenario is simulated, MATLAB is used to obtain optimal base stock, S* and kanban, K* for the three systems. Based on these optimal values, operation of SS/SP/EKCS, TKCS and BS is simulated in ARENA simulation software. Finally, results are tabulated (Appendix C) and compared in terms of Actual Total Cost (ATC).

In this experiment, Key Performance Indicator (KPI) for each KCS is "Actual Total Cost" (ATC). All other KPIs are translated into ATC. For example, a KPI like fill rate can be indirectly represented by total backorder cost (since number of backorders is simply number of demands "unfilled"), while a KPI like average inventory level can be represented by the total holding cost. In order to obtain ATC for each scenario, the total backorder, shortage and holding costs are added together. Thus,

$$\text{Actual Total Cost (ATC)} = \text{Total Backorder Cost} + \text{Total Shortage Cost} + \text{Total Holding Cost} \quad (4.1)$$

Where

- Total Shortage Cost = C_s x Average Customer Waiting Time in Customer Demand Queue (obtained from Arena)
- Total Backorder Cost = C_b x Total Number of Backordered Customers in Demand Queue (obtained from Arena)
- Total Holding Cost = C_h x Average Stock Held in Output Buffer (obtained from Arena; for a simulated one year period)

4.2.1 Simulation Assumptions

Assumptions made in modelling SS/SP/ KCS are:

- All systems produce only a single product.
- One card Kanban system is adopted.
- The system produces no defective parts.
- All systems have a single stage containing only one MP.
- Each MP contains only one machine/server.
- Machine setup times are zero.
- No machine failures occur.
- Each machine can only process one part per unit time.
- Parts are transported with negligible transfer time.
- Demand signals and kanbans flow instantaneously.
- Parts follow a First In First Out (FIFO) dispatching policy at all machines and buffers.
- Input material buffers have an infinite supply of component parts.
- Demand arrivals follow a Poisson process.

- All processing times at MPs are assumed to be exponentially distributed.
- Each replication is run for one year.
- Each simulation run is replicated 10 times.
- The warm up period for each replication is three months.

The ARENA snapshots for SS/SP/KCS are shown in Appendix J1. These simulation models (Figure J1 to J3) were developed based on their respective KCS schematics. Thus, SS/SP/BS (Figure J1) corresponds to Figure 2.4; SS/SP/TKCS (Figure J2) corresponds to Figure 2.5 and SS/SP/EKCS (Figure J3) corresponds to Figure 3.1. These models were simulated only after their respective optimal parameters were found. For example in BS case, the optimal base stock, S*, was found using the method discussed in 4.2, coded in MATLAB and simulations run using the parameters listed above (Table 4.1). The MATLAB code for optimizing a SS/SP/BS is listed in Appendix I2. These steps are repeated for SS/SP/TKCS and EKCS. The simulation results are discussed in the next section.

4.3 Discussion of Simulation Results

Figures 4.1 to 4.3 show the ATC of each system with varying demand arrival rates. They provide a graphical summary of simulation results listed in Appendix C. The figures show that EKCS and TKCS outperform BS significantly by achieving a lower ATC. However, the most interesting insight is that the performance of EKCS does not differ much from that of TKCS.

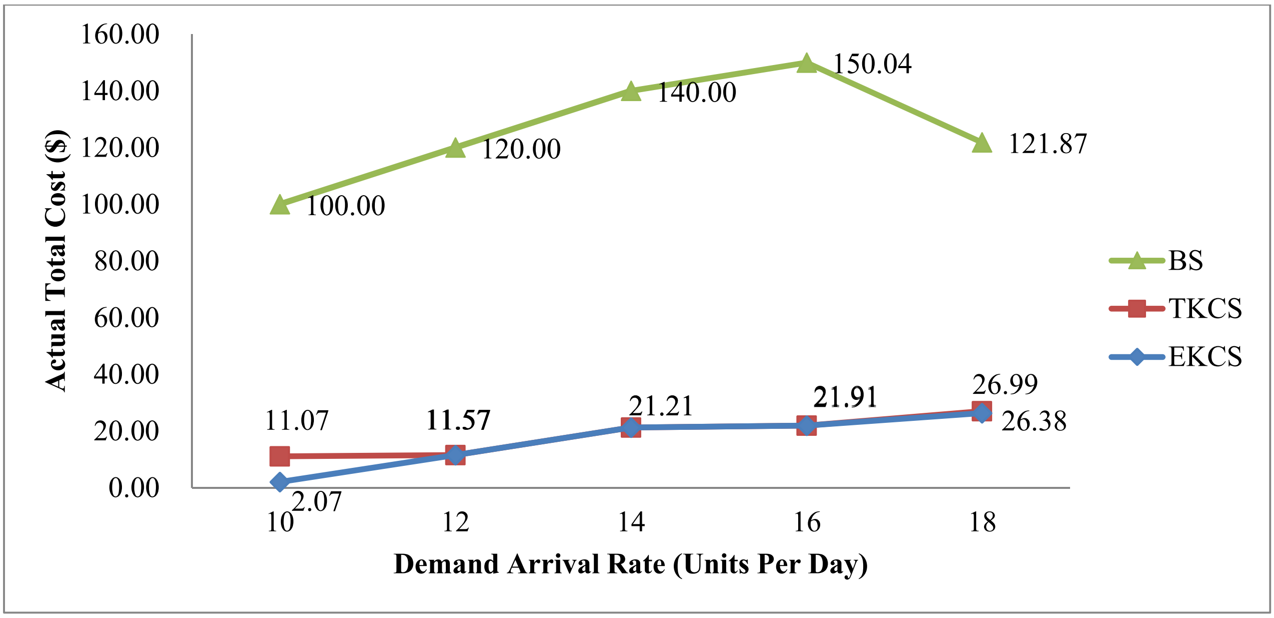

Figure 4. 1: Comparison of SS/SP/EKCS, TKCS and BS in low backorder and shortage costs scenario

Referring to Figures 4.1 to 4.3, the most prominent difference is between BS and the other two systems. BS incurs the highest cost, followed by TKCS and EKCS. BS, by definition, keeps a pre-specified level of stock, thereby incurring higher inventory costs. Because BS follows a "push" production strategy, whilst EKCS and TKCS follow a "pull", BS will produce stock according to the demand arrival rate; the higher the arrival rate, the more stock it will produce. In fact, the popular optimization algorithm for BS (proposed by Zipkin (2000) does not take into account the Manufacturing Process (MP) rate, as the idea is to stock up and prevent a stock-out situation.

On the other hand, TKCS and EKCS follow a lean philosophy. They produce only when needed and keep inventory low. Their optimization methods incorporate MP rates to obtain the utilization rate. The utilization rate represents the level of congestion in the system, and determines the optimal number of kanbans using Markov Chains. Since the number of kanbans defines WIP or "congestion" level, controlling it ultimately determines the inventory level.

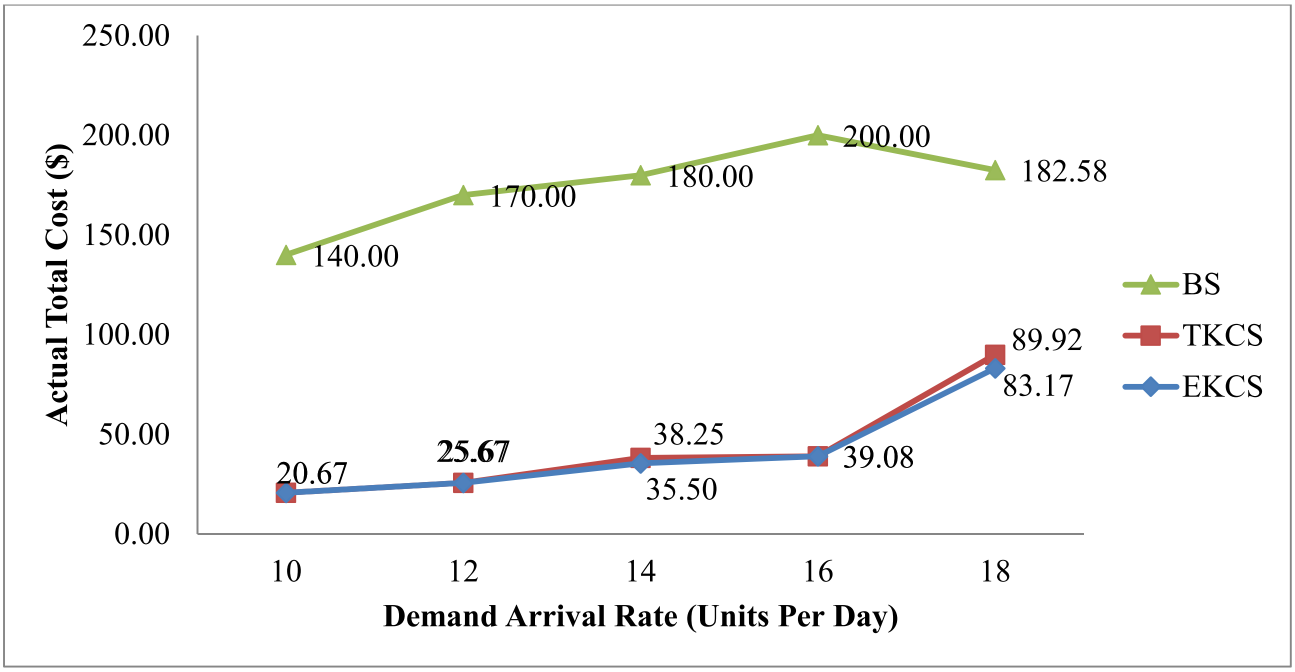

Figure 4. 2: Comparison of SS/SP/EKCS, TKCS and BS in medium backorder-and-shortage-costs scenario

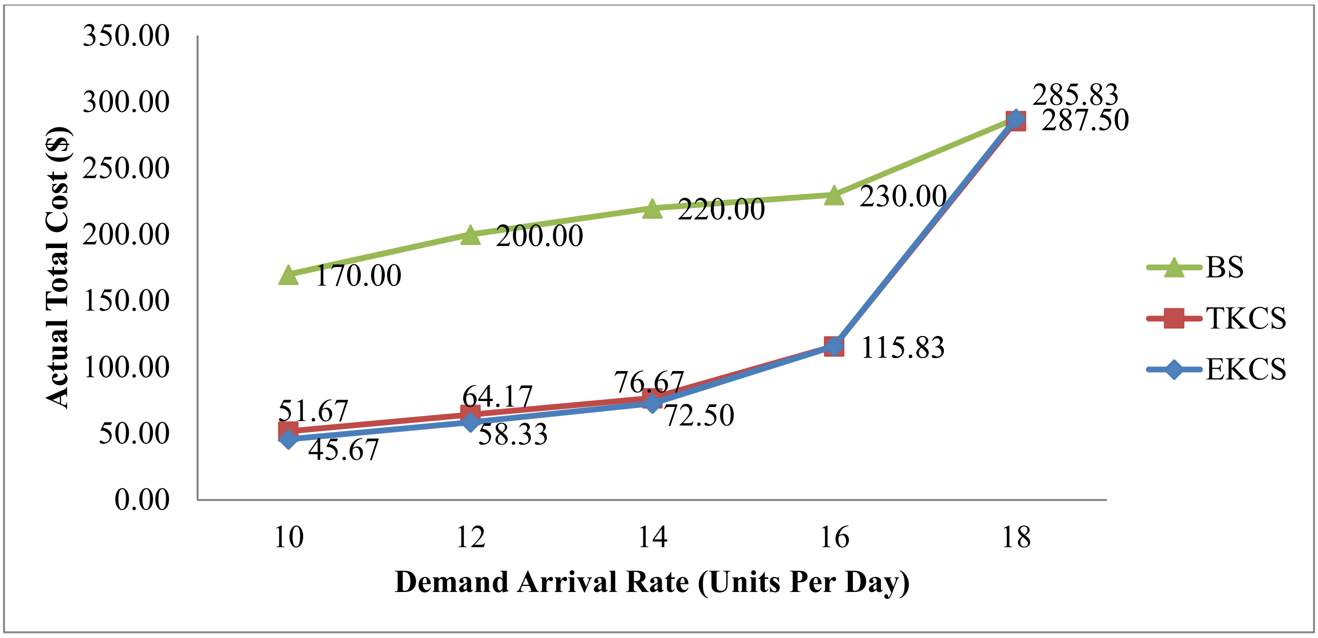

Figure 4. 3: Comparison of SS/SP/EKCS, TKCS and BS in high backorder-and-shortage-costs scenario

4.3.1 EKCS and TKCS performance similarity

Looking at Figures 4.1 to 4.3, it can be noted that performance results of EKCS and TKCS differ very little. In fact, EKCS seems to imitate TKCS in almost all scenarios. On the surface, they look vastly different, with EKCS having something that TKCS doesn't, namely instantaneous transmission of demand. However, deeper analysis reveals that they aren't as different as they seem to be.

Referring to Tables C1, C3, C7, C9, C13 and C15 (in Appendix C), and looking at average inventory levels, both seem to hold almost equal amount of inventory; hence, their average backorders and customer waiting times are almost equal. This leads to their Actual Total Costs (ATC) being very close. They hold comparable inventory levels because their optimal number of dispatched kanbans is always the same. Dallery (2000) also notes that by setting the number of kanbans for TKCS the same as the base stock level of EKCS, EKCS ***becomes*** and behaves like TKCS.

4.3.2 Dispatched kanbans between EKCS and TKCS

The number of kanbans calculated by optimization algorithms for EKCS and TKCS are almost equal. The number of dispatched kanbans in EKCS represents its "base stock" level, just as kanbans in TKCS represent the average inventory level. Logically, the more stock a system has, the higher its inventory level, but with lower backorders and customer waiting time, and vice versa. Hence, adding one more dispatched kanban is equivalent to increasing the base stock, and increasing holding cost, and thus ATC.

On the other hand, taking away one kanban lowers base stock by one, lowering holding cost, but incurring longer customer waiting time, thereby (possibly) increasing ATC again. This illustrates the need to balance holding costs and backorder/shortage costs. The optimization algorithms used in this research seek the lowest ATC in all scenarios to compute optimal base stock, S* and/or number of kanbans, K*.

4.3.3 Undispatched kanban queue in EKCS: some comments

Proposers of EKCS have claimed that it is *"leaner"* than TKCS, as it uses its undispatched kanban queue to lower inventory, yet achieves optimal WIP. However, this research has shown that that does not result in EKCS outperforming TKCS; rather, its performance gets worse. This may be due to the following reasons:

1. By reducing the number of kanbans and placing them in the undispatched queue, average on-hand inventory level is reduced, leading to higher backorder and shortage costs and ultimately, higher ATC, outweighing the benefits of lower stock. The proposers of EKCS had the idea of reducing stock by having the undispatched kanban queue locked away and

used only when needed. But the moment EKCS has base stock, optimal EKCS *becomes* optimal TKCS, and optimal dispatched kanbans in EKCS make it analogous to TKCS.

2. Base stock in EKCS makes the undispatched kanbans redundant. Referring to Figure 4.4, the only time the undispatched kanbans are allowed into the MP are in the event of demand arrivals. But demand arrivals are always accompanied by kanbans being passed from downstream stages, which are then placed behind these undispatched kanbans. Those kanbans in front of queue K_1 get attached to component parts, and are sent into MP (since there is already demand arrival in queue D_1). This brings the undispatched kanbans back to the original number. Looking back, the initial proposed role of undispatched kanbans in EKCS was to allow more WIP into MP. But it turns out that absolutely no benefit results, since the bottleneck is the MP rate! In other words, more WIP can be allowed into MP for EKCS, but MP can still only process one part per unit time.
3. Instantaneous transmission of demands in EKCS does not deliver any benefits, as kanbans in queue K_1 already fulfil this role. This is not immediately clear. But upon closer examination of Figures 4.4 and 4.5, the same thing can be observed for both systems: once a demand arrives, and there is a part in the output buffer, a component part is instantly sent into MP. Thus, with or without queue D_1, EKCS behaves identically to TKCS. Of course, this is under the assumption of negligible kanban transfer time. Even if this assumption was relaxed and kanban transfer time was taken into account, it would still affect both systems the same way.

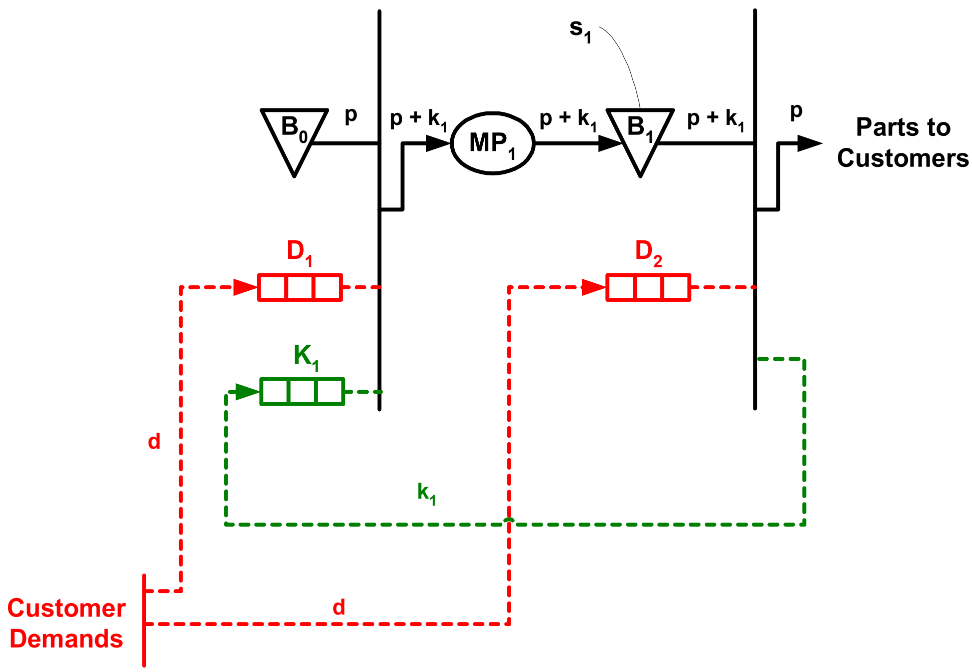

Figure 4. 4: SS/ SP/ EKCS

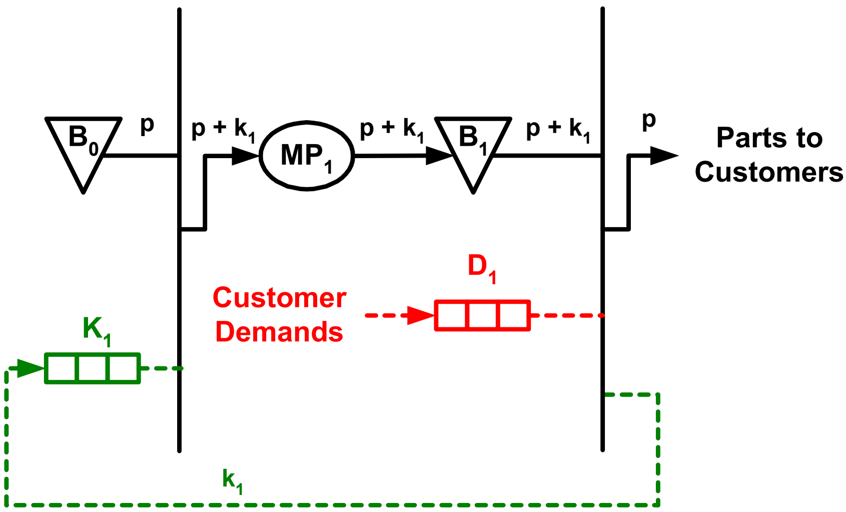

Figure 4. 5: SS/ SP/TKCS

4.3.4 Study of Multiple Stage, Single Product Kanban Control Systems (MS/SP/KCS)

TKCS (Figure 4.5) features instant demand transmission to all stages upon a demand arrival, since its kanbans are transmitted to their individual stage MPs immediately (assuming a part is in the output buffer), even though demand arrives only at the final stage. This makes the role of instantaneous transmission of demands in EKCS redundant. Likewise for EKCS (Figure 4.6), undispatched kanbans in queues K_1 and K_2 are unnecessary. If base stock exists in B_1 and B_2, any demand arrivals will immediately transmit previously attached kanbans upstream, joining the undispatched kanban queues K_1 and K_2. Those kanbans placed in front are then attached to component parts and sent into MP. In the end, the undispatched kanbans are still rendered unnecessary.

In chapter six, SS/MP/EKCS is studied, as the working mechanism is expected to be different.

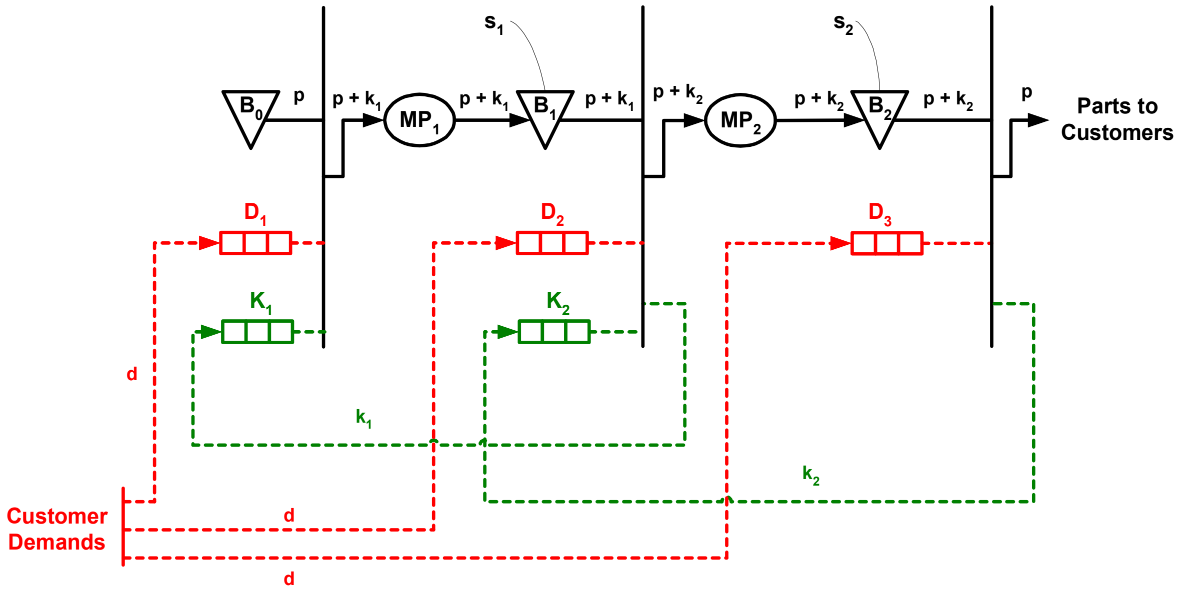

Figure 4. 6: Two Stage, Single Product EKCS

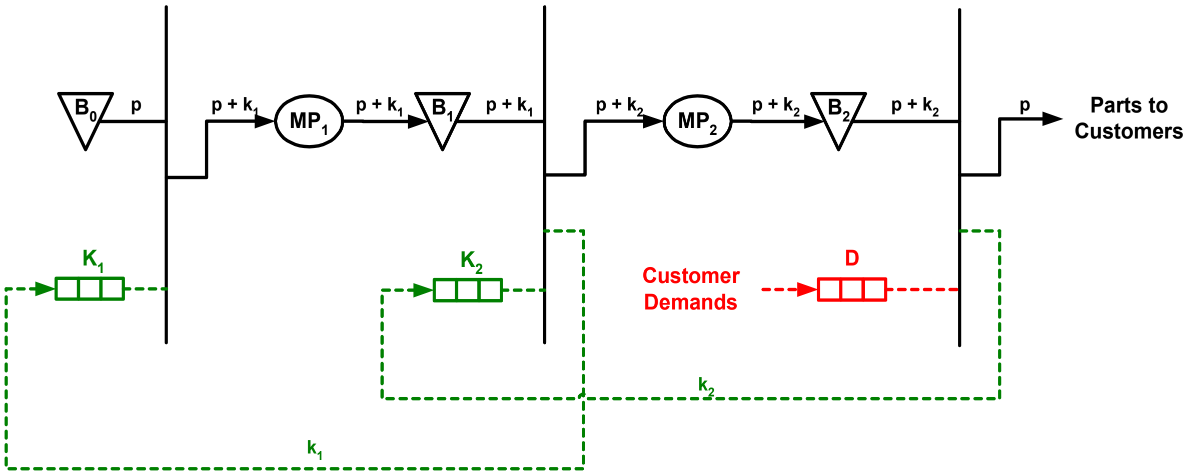

Figure 4. 7: Two Stage, Single Product TKCS

4.3.5 Comparison of TKCS and EKCS in low utilization scenario

This section analyses scenarios under which EKCS may outperform TKCS. Further investigation (Figures 4.8(a) to (c),) shows that below 50% utilization rate, for low backorder, C_b, and shortage cost, C_s, EKCS outperforms TKCS. However, medium and high C_b and C_s scenarios show negligible difference. As can be seen in Figure 4.8, the cost difference between EKCS and TKCS is quite significant and worthy of further study.

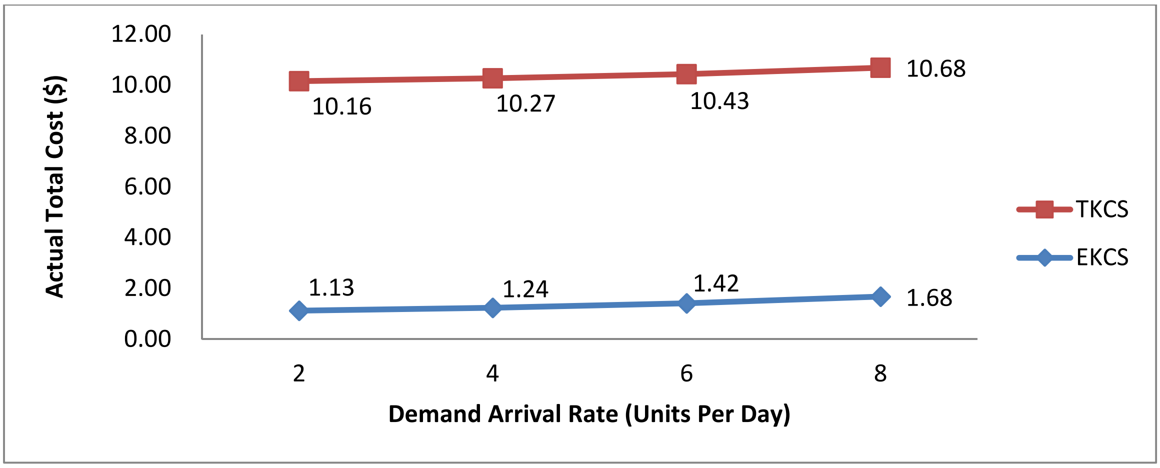

Figure 4. 8 (a): Comparison of EKCS and TKCS (low utilization rates and low backorder and shortage costs)

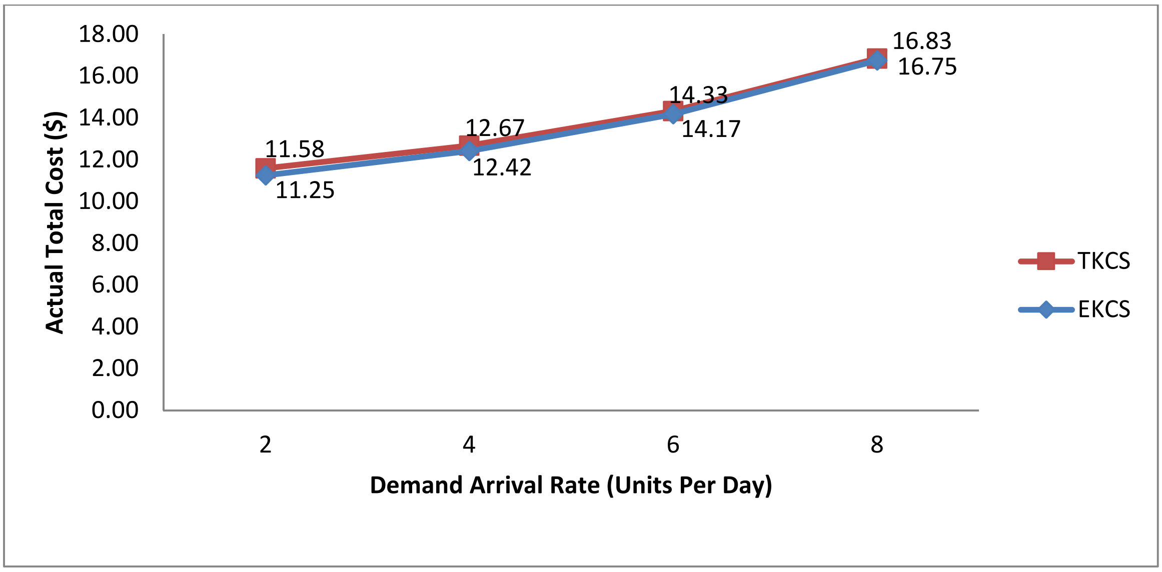

Figure 4. 8(b): Comparison of EKCS and TKCS (low utilization rates and medium backorder and shortage costs)

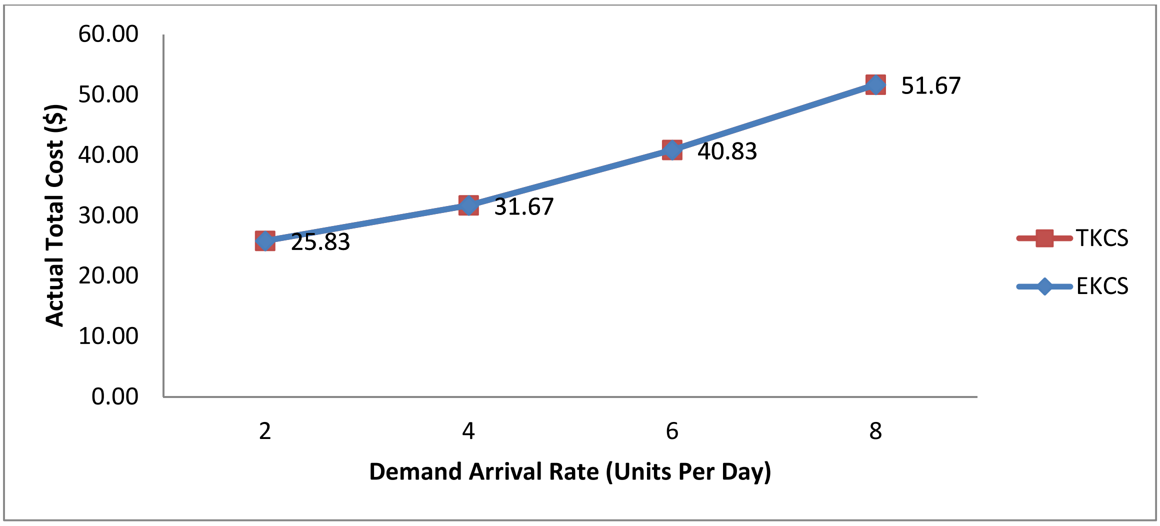

Figure 4. 8(c): Comparison of EKCS and TKCS (low utilization rates and high backorder and shortage costs)

4.3.6 EKCS and TKCS: Some comments

The reason for the significant cost difference in Figure 4.8(a) is that EKCS has zero holding cost. EKCS can afford not to hold stock in its output buffer as:

1. The utilization rate is very low—below 50%. This means that, most of the time, MP is idle and whenever demand arrives, it can produce at a fast rate to meet the demand. In contrast, Figure 4.1 shows that the gap starts to close at above 10 demand arrivals per day (or above 50% utilization), as MP gets increasingly congested. Stock is now needed to prevent backorders and to shorten customer waiting time. But the moment EKCS has base stock, it starts to behave like optimal TKCS.
2. The ratio of holding cost, C_h to backorder, C_b, and shortage cost, C_s, is low. This implies that the cost incurred in holding stock in EKCS is comparable to the cost for backorders and shortages. So in this scenario, having lower stock proves to be less costly, and making customers wait leads to the same cost penalty as holding stock. TKCS, however, is forced to hold same amount of stock as the number of kanbans. Hence, even though EKCS and TKCS can both have same number of kanbans in their system–each having only one (Table C19 and Table C21)-TKCS attaches kanbans to real stock, whereas EKCS holds kanbans in the undispatched queue, which is converted into finished product only upon arrival of a demand.

The most important insight here is that the extra demand queue D_l (Figure 4.4) in

EKCS prevents undispatched kanbans from entering MP when there is no demand, thereby making EKCS a system with truly no inventory.

4.3.7 EKCS and TKCS: medium and high backorder, C_b, and shortage Costs, C_s, scenario

In medium to high C_b and C_s scenarios (Figure 4.8 (b) and (c)), the gap between EKCS and TKCS is negligible. The key factor is whether EKCS holds base stock. In medium C_b and C_s scenario (C_s =20xC_h), EKCS does not hold base stock (Table C23), thus increasing shortage costs. This leads to higher costs for EKCS. EKCS, in this case, is still capable of remaining *stockless* and achieving a slightly lower cost than TKCS.

In a high C_b and C_s scenario (C_s=200xC_h), there is no difference in cost between EKCS and TKCS, as EKCS now has to hold base stock. It cannot be stockless anymore as the penalty for shortage is too high, leading to the identical Actual Total Cost (ATC) curves in Figure 4.8(c).

4.3.8 EKCS' performance: Final comments

If EKCS holds base stock, the undispatched kanbans become ineffective. But when EKCS does not hold base stock, the question becomes: would the undispatched kanbans still be useful? Once EKCS is stockless, the maximum number of undispatched kanbans required is only one, as the MP can only produce one part at a time. Increasing the number of undispatched kanbans allows more WIP into MP, but that only makes it more congested. In other words, even if more than one demand arrives simultaneously, the system only really needs one undispatched kanban, as the sole undispatched kanban can be sent back instantly to upstream queue K_1 and then into MP for processing. This is also what would happen in a system with multiple undispatched kanbans. The conclusion then is that EKCS outperforms TKCS only when it contains no base stock and during low-demand arrival rate and low-backorder (C_b) and shortage cost (C_s) scenarios. Also, the optimal number of undispatched kanbans in the system is one, assuming negligible kanban transfer times.

4.4 Summary and Conclusions

This chapter presents a performance comparison of single-stage, single-product kanban control systems (SS/SP/KCS), namely EKCS, TKCS and BS. Firstly, optimization models for BS and TKCS are described. These optimization models are used to find the optimal base stock, S*, and optimal number of kanban K* for the respective systems. Then, three scenarios with different simulation parameters are set up to compare the individual KCS performance. The simulation results in this chapter show that BS incurs the highest cost in all cost scenarios, while EKCS is found effective only under very special cases. Also, TKCS is still a very powerful system. The only time EKCS outperforms TKCS is when demand arrival rate is low, and backorder, C_b, and shortage costs, C_s are low as well, since under those circumstances it does not need to hold stock. The most important insight made is that EKCS *behaves* like TKCS once it contains base stock (or dispatched kanbans). This chapter also supports the superiority of the pure kanban system, the TKCS, over BS.

BS was developed in the 1960s, while TKCS was developed in the 1970s and EKCS in the 2000s. Naturally, TKCS outperforms BS, as lean production seems to work best for mass-produced products, such as cars, which is where these systems are predominantly implemented. In fact, many publications have described TKCS as the "Just–In–Time (JIT)" revolution that made Toyota the biggest car manufacturer in the world (Womack et al., 2007). In this chapter, it is shown that BS always incurs the highest cost, as it stocks a higher level of inventory, disregarding the MP processing rate and putting emphasis only on demand arrival rate. All in all, the results clearly illustrate that BS is an inferior control system when compared to pull-type control systems.

To summarize, the main findings of this chapter are:

1. EKCS outperforms TKCS only when the demand rate is low (<50% utilization rate) and backorder, C_b, and shortage costs, C_s are low.
2. If EKCS has stock in its output buffer, it behaves exactly like TKCS. Their performance becomes the same, as the optimal number of dispatched kanbans is the same.
3. If EKCS has stock in its output buffer, its undispatched kanbans become ineffective, and the number of kanbans equal the base stock.
4. The role of extra demand queues for instantaneous transmission in EKCS (queues D_1 and D_2 in Figure 4.9) is ineffective, as TKCS also has this functionality but without the additional queues. In other words, assuming negligible kanban transfer times, TKCS' kanban queues also act to instantaneously transmit a demand signal.
5. Extra demand queues are useful only when EKCS has no stock in the output buffer. Extra demand queues help lock up undispatched kanbans, which makes EKCS truly stockless.
6. It has been shown that MS/SP/EKCS behaves similar to MS/SP/TKCS, assuming

negligible kanban transfer times; the optimal number of undispatched kanbans in such a case is one.

The next chapter presents SS/MP/KCS, systems that operate and behave very differently from SP/KCS.

CHAPTER 5

MULTIPLE PRODUCT KANBAN CONTROL SYSTEMS (MP/KCS)

In Chapter 4, a performance comparison of SS/SP/KCS was presented. Simulation results showed that EKCS outperforms TKCS and BS, but only under specific conditions, such as low utilization rates and low backorder and shortage cost. TKCS still performs very well, whereas BS is the worst performer. This chapter follows up on that comparison and presents a study of multi-product systems, examining the operating mechanics of the following systems are presented:

- Single Stage, Multiple Product Base Stock (SS/MP/BS);
- Single Stage, Multiple Product Dedicated Traditional Kanban Control System (SS/MP/De-TKCS)[1];
- Single Stage, Multiple Product, Shared Extended Kanban Control System (SS/MP/Sh-EKCS); and
- Single Stage, Multiple Product, Dedicated Extended Kanban Control System (SS/MP/De-EKCS);

For the sake of simplicity, in this chapter, only a two-product case is considered.

5.1 Dedicated versus shared kanbans

The difference between dedicated and shared kanbans, as the names suggest, is that dedicated kanbans are assigned only to a specific product type, for example kanban 1 for product type 1 and kanban 2 for product type 2. In other words, dedicated kanbans are *type specific*, whereas shared kanbans can be used for any type of product; that is, they can authorize any product type into the MP, depending upon the sequencing rule.

[1] B. Baynat, Buzacott, and Dallery (2002) proposed two kinds of MP/TKCS. They are dedicated and shared TKCS. However, they have been proven to be equivalent and it is not worthwhile to consider the shared case. Hence, in this research, only the SS/MP/De-TKCS is considered. In further discussions 'De' is dropped and the system is simply termed as SS/MP/TKCS.

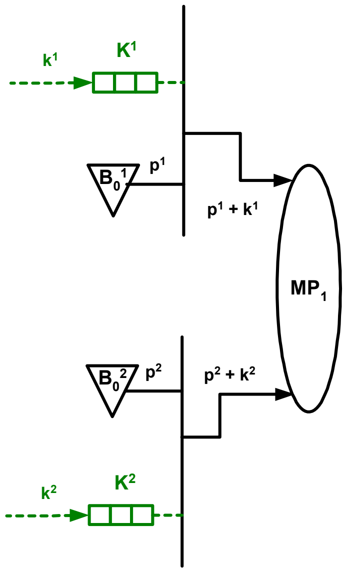

Figure 5. 1: Dedicated Kanban queue (B. Baynat et al., 2002)

In Figure 5.1, a product must be present in buffer $B_0{}^1$, and a kanban for product type 1, k^1, must be present before they are allowed to proceed into the MP. Notice that there are two different queues, K^1 and K^2, each representing a kanban queue for the specific product type. Figure 5.2, in contrast, has only one kanban queue (shared); as long as there is a product in either $B_0{}^1$ or $B_0{}^2$, and a shared kanban in kanban queue K, the product and kanban are authorized for production. It is assumed that input buffer $B_0{}^j$ contains infinite components of type j.

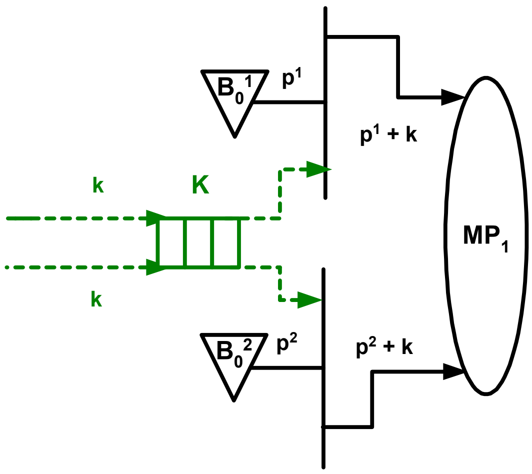

Figure 5. 2: Shared Kanban queue (B. Baynat et al., 2002)

In this chapter, the control systems considered have a single stage and process two product types. Each MP consists of only one server or machine, and its processing rate is different for different product types. Product type 1 has priority in processing, so that if demand for both product types occurs at the same time, product type 1 is allowed into the MP first.

5.2 Single Stage, Multiple Product Base Stock (SS/MP/BS) System

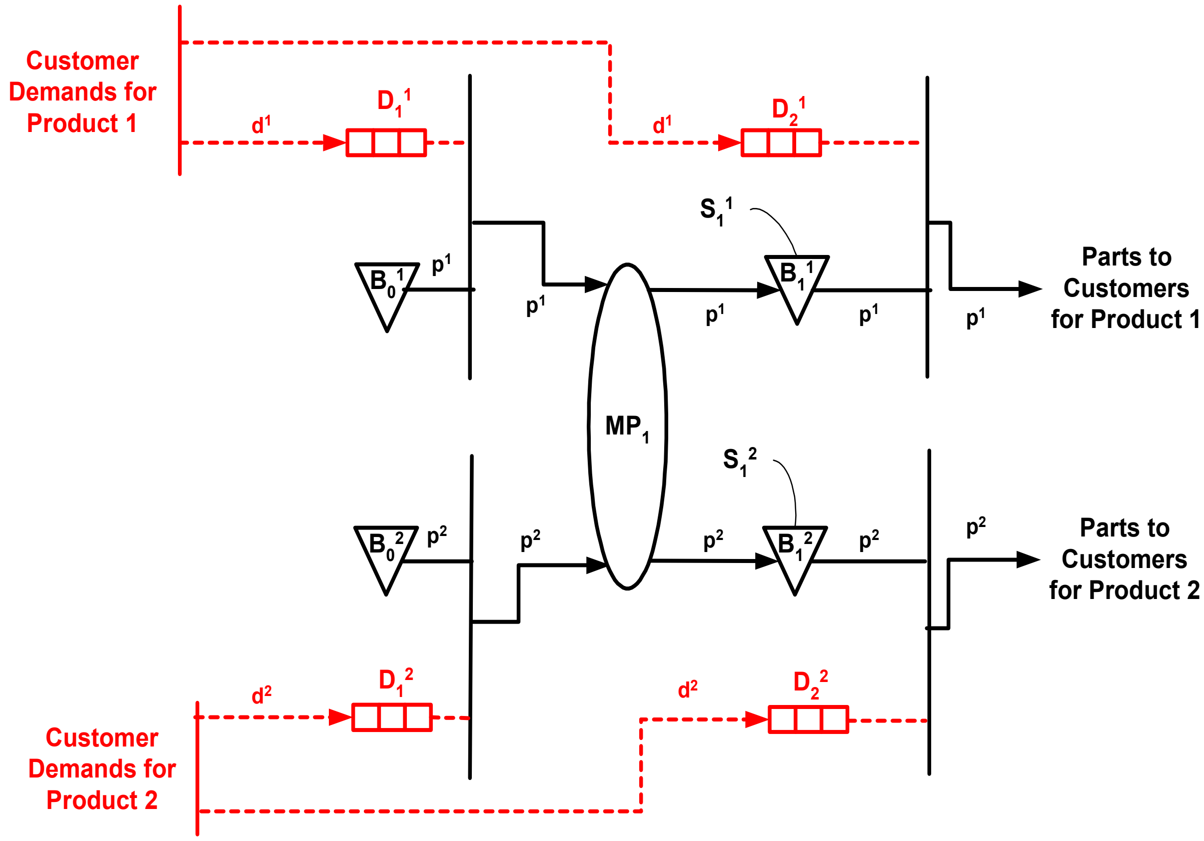

Figure 5. 3: Single Stage, Multiple Product Base Stock (SS/MP/BS) System (B. Baynat et al., 2002)

Figure 5.3 shows a SS/MP/BS system for two product types. In the initial state of the system, B_0^1 and B_0^2 have infinite component parts; B_1^1 and B_1^2 have base stock levels S_1^1 and S_1^2, respectively; and all other queues are empty. When a customer demand arrives, it is instantaneously transmitted to the two demand queues, D_1^j and D_2^j, depending on the product type j.

The MP/ BS system operates as follows. When a customer demand, for either part type j, arrives at the system, it is duplicated and immediately transmitted to its respective queues D_i^j. The last one joins Queue D_2^j, requesting the release of a finished product from B_1^j to the customer. At this time, there are two possibilities:

a. If a part is available in B_1^j, it is released immediately to the downstream stage, and MP produces one part to top up the base stock S_1^j in B_1^j.

b. If no part is available in B_1^j, the demand is backordered and waits in

Queue $D_1^{\ j}$ until a new part is completed at the MP.

5.2.1 SS/MP/BS optimization

The SS/MP/BS system is considered a dedicated one, as product types have separate component types and use their dedicated buffers. In other words, it is a combination of two dedicated single product systems, as both demands and products are type-specific. For this reason, the single product optimization methods can be used in the multiple product systems, except that during calculation of optimal base stock levels, S_1* and S_2*, the average MP rate is used. In a multiple-product base stock system, each item's optimal base stock can be computed individually (Zipkin, 2000).

Thus, Wallace J. Hopp and Spearman (2008) method can be used to calculate individual optimal base stock levels, S_1* and S_2* in the multi-product BS system. The required inputs are the individual product's demand arrival rates λ_1 and λ_2, and holding and backorder costs, C_{h1} , C_{h2}, C_{b1} and C_{b2}.

5.3 Single Stage, Multiple Product, Traditional Kanban Control System (SS/MP/TKCS)

In Figure 5.4, kanban movement is shown by the dashed green lines. Buffer $B_0^{\ j}$ is the raw materials buffer, and is assumed to contain infinite component parts. When the system is in its initial state, $B_i^{\ j}$ contains $K_i^{\ j}$ finished parts, each part having a stage i kanban attached to it. All other queues are empty. SS/MP/TKCS operates as follows. When a customer demand for part type j arrives at the system, it joins Queue $D^{\ j}$, requesting the release of a finished product from $B_1^{\ j}$ to the customer. At that time, there are two possibilities:

a. If a part is available in $B_1^{\ j}$ (which is initially the case), it is released to the customer after detaching its kanban. This kanban is transferred upstream to queue $K^{\ j}$, carrying with it a demand signal for the production of a new finished part. As soon as a kanban signal arrives in $K^{\ j}$, it authorizes the production of a new part from component buffer $B_0^{\ j}$
b. If no part is available in $B_1^{\ j}$, demand is backordered and waits in queue D^j until a new part is completed, and arrives in $B_1^{\ j}$. The newly finished part is released to the customer and the detached kanban is transferred to queue $K^{\ j}$ at the same time.

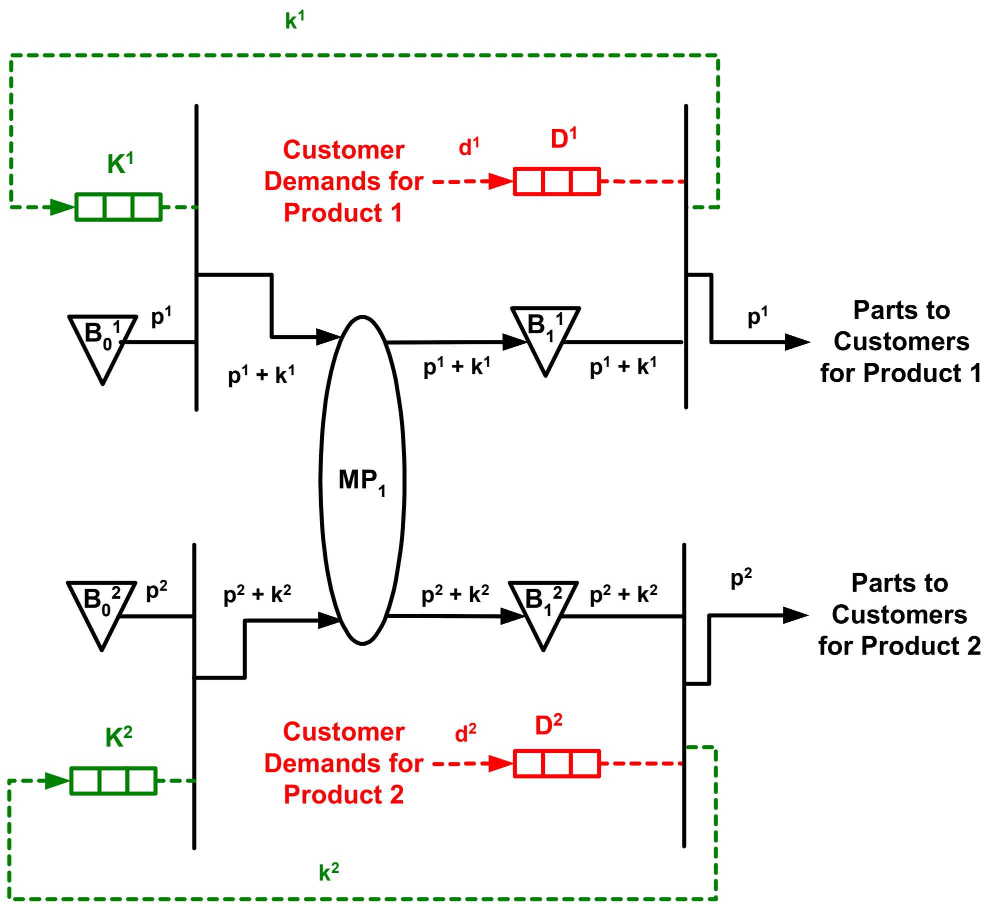

Figure 5. 4: Single Stage, Multiple Product, Traditional Kanban Control System (SS/MP/TKCS)

5.3.1 SS/MP/TKCS Optimization

Askin, Mitwasi, and Goldberg (1993) optimized the SS/MP/TKCS using Markov Chains. In this research, their technique is modified for improved computational efficiency (Appendix D).

$$\underset{k}{\text{Minimize}}\, ETC(k) = \frac{C_s\left(\lambda^k\right) + C_h\left[\sum_{x=1}^{k} x\left(\frac{k!}{(k-x)!}\lambda^{k-x}\rho^x\right)\right]}{\sum_{x=0}^{k}\frac{k!}{(k-x)!}\lambda^{k-x}\rho^x} \tag{5.1}$$

Equation 5.1 shows the equation for the Expected Total Cost (ETC) for a MP/TKCS; its derivation can be found in Appendix D. This method is a heuristic algorithm which gives a near optimal solution. The near-minimum cost is found using exhaustive enumeration. ETC is computed for the feasible range of K until ETC (K+1) - ETC (K) >=0.

5.3.1.1 Manufacturing Example

In this section, a manufacturing example is used to validate the comparison between different MP/KCS. This example comes from the practice of automobile platform sharing. According to Diem & Kimberly (2001), an automobile platform can be defined as “a vehicle’s primary load-bearing assembly. It determines a vehicle’s basic size and links the driveline and suspension components.” The benefits of sharing an automobile platform between different models include potential savings from sharing parts while still preserving marque identity. For instance, the platform shared between Peugeot and Citroen helped save €0.4 billion in one project, while using a common platform for Volkswagen has helped cut costs by €1 billion and speed up development of niche vehicles, all in the year 2000.

This example refers specifically to the Volkswagen small passenger car platform, platform A. This platform produced a series of mini-car models such as VW Golf, Audi A3, and Audi TT in 2000 (Diem & Kimberley, 2001).

Figure 5. 5:Audi TT (Diem & Kimberley, 2001)

Figure 5. 6: VW golf (Diem & Kimberley, 2001)

To model this scenario as a single stage, two-product TKCS (Figure 5.4), Production of platform A can be considered as the single-stage Manufacturing Process (MP), while the two product types can refer to VW Golf (Figure 5.5) and Audi TT (Figure 5.6) models. Product type 1 is assumed to be the expensive product – the Audi TT – while product type 2 is assumed to be the cheaper product – the VW Golf. The demand arrival rate for product type 1, λ_1, is assumed to be five products per month, while that for product type 2, the inexpensive model, λ_2, is assumed to be 12. This assumption is based on the Volkswagen platform production in 2000, where the production for VW Golf was far greater than the Audi TT (Diem & Kimberley, 2001). This means that the sales forecast for VW Golf were far greater than for Audi TT, since it was a more economical car. Hence, it can be assumed that the demand arrival rate for the VW Golf is greater than for the Audi TT.

Next, the Manufacturing Process (MP) rate for product type 1 customers, μ_1, is 18 products per month, while the MP rate for product type 2 customers, μ_2, is 20 products per month. These values of λ_1, λ_2, μ_1 μ_2, are chosen in order for the standard queuing theory assumption to hold. A more detailed explanation for why these values were selected can be found in Section 5.6.1.

Following this, the holding cost for product type 1, C_{h1}, is assumed to be \$10 per product per month, while the holding cost for product type 2, C_{h2}, is assumed to be \$5 per product per month. Holding cost for product 1, the Audi TT, is twice that of product 2, the VW Golf, because the control electronics used in the Audit TT's engine and transmission system are better and more expensive than those in the VW Golf (Diem & Kimberley, 2001). Thus, since these software electronic components become obsolete quickly, the opportunity cost for not selling an Audi TT immediately after production would most likely be higher than for the VW Golf.

The shortage cost for product type 1, C_{s1}, is assumed to be \$100 per unit per month, and for product type 2, C_{s2}, \$20 per unit per month. Shortage cost for the Audi TT is five times that of the VW Golf because the Audi TT's production quantity is very small compared to the VW Golf (Diem & Kimberley, 2001). This means that it's less likely for the VW Golf to experience a stock-out than Audi TT. This also means that if customers want to purchase an Audi TT but gets turned away due to a stock-out situation, they could have many other cars of the same platform to choose from (Diem & Kimberley, 2001).

Finally, these inputs are summarized in Table 5.1 below, and an algorithm to optimize SS/MP/TKCS is coded in MATLAB (Appendix I4), and it is tested with these inputs.

Table 5. 1: Input parameters for a test SS/MP/TKCS

Product Type 1				Product Type 2			
λ_1 (units/time)	μ_1 (units/time)	C_{h1} (\$/unit/time)	C_{s1} (\$/unit/time)	λ_2 (units/time)	μ_2 (units/time)	C_{h2} (\$/unit/time)	C_{s2} (\$/unit/time)
5	18	10	100	12	20	5	20

Using the input parameters from Table 5.1, and searching over a range of 20 for the optimal number of kanbans for product 1, K_1* the resulting costs (output from MATLAB) are plotted in Figure 5.7 and Figure 5.8.

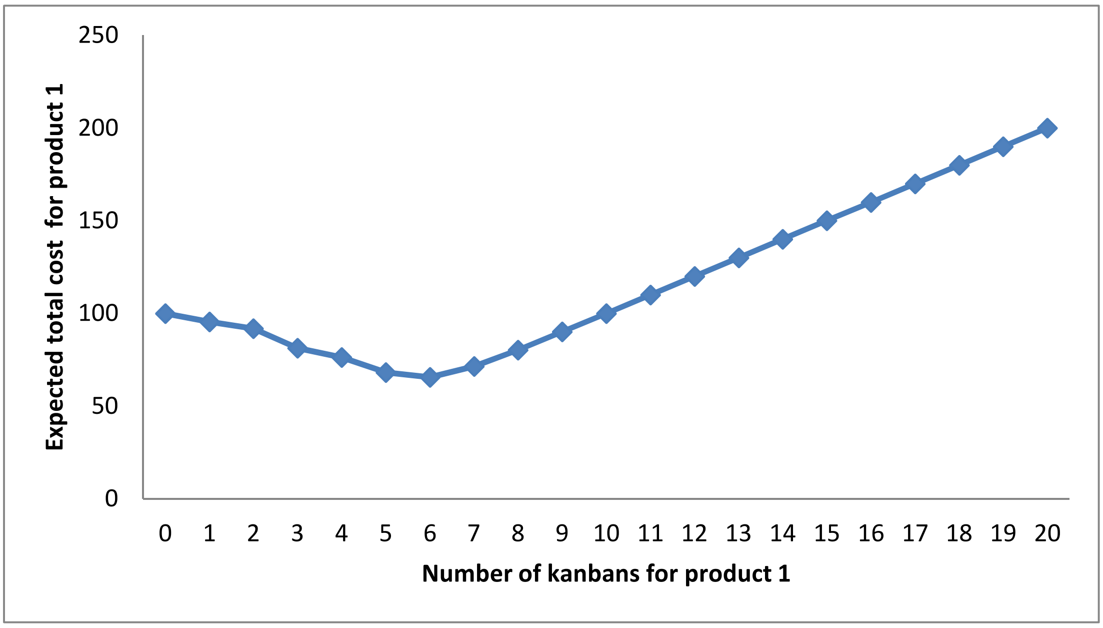

Figure 5. 7: Expected Total Cost (ETC1) for SS/MP/TKCS

Figure 5.7 shows that the optimal K1* is six because the corresponding Expected Total Cost (ETC1) is $65.64. Likewise, using a search range of 20 for the optimal number of kanbans for product 2, K_2* the resulting costs are plotted in Figure 5.8.

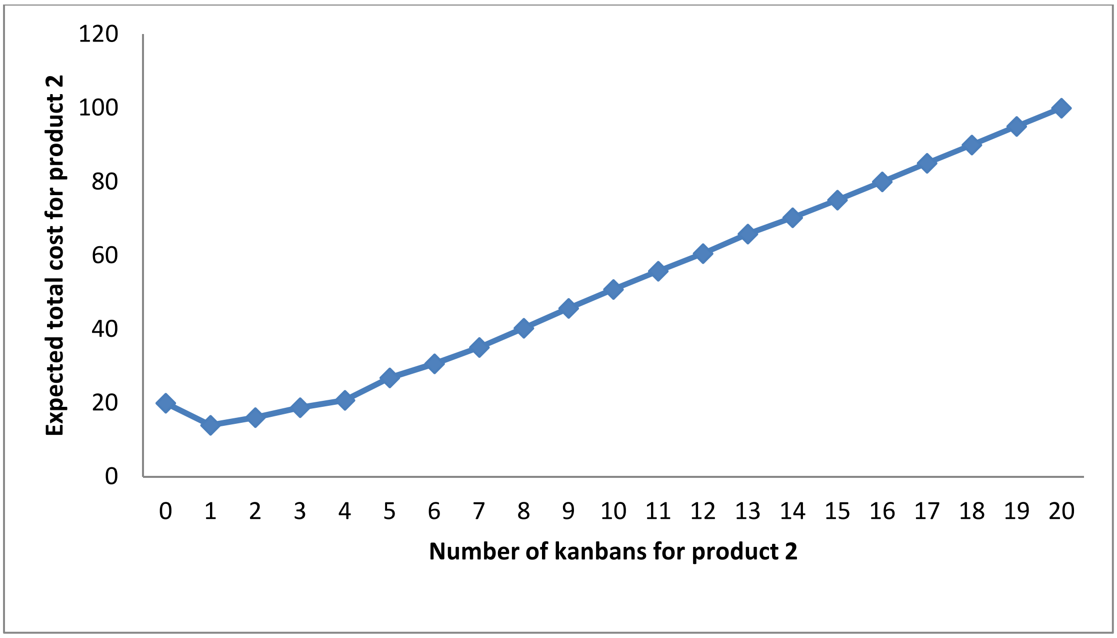

Figure 5. 8: Expected Total Cost (ETC2) for SS/MP/TKCS

Figure 5.8 shows that the optimal K2* is one since the corresponding Expected Total Cost (ETC2) is \$14. Note that the values shown in Figures 5.7 and 5.8 are as expected. Referring to equation 5.1, for example, if $K_1 = 0$ (meaning there are no product 1 kanbans in the system), the equation reduces to a simple shortage cost per unit for product 1, C_{s1}, which in this example is \$100. Looking at Figure 5.7, this also holds true. If there is no product 1 units to satisfy the demand the penalty/shortage cost would be \$100. Same holds for product 2. Referring to equation 5.1, for example, if $K_2 = 0$ (meaning there are no product 2 kanbans in the system), the equation reduces to a simple shortage cost per unit for product 2, C_{s2}, which in this example is \$20. Looking at Figure 5.8, this also holds true.

In summary, this example shows the computation of optimal K1* and K2*. Henceforth, these input parameters, λ_1, λ_2, μ_1 μ_2, C_{h1}, C_{h2}, C_{s1}, C_{s2}, are used in the simulation models of Chapter 6.

5.4 Single Stage, Multiple Product, Dedicated Extended Kanban Control System (SS/MP/De-EKCS)

Figure 5.9 shows the process flow of a MP/De-EKCS; Buffer $B_1{}^j$ is the output buffer for product type j and contains dedicated kanbans attached to finished products. Buffer $B_0{}^j$ is the raw materials buffer and is assumed to contain infinite number of components. Queue $D_2{}^j$ contains incoming customer demands and queue $K_1{}^j$ contains undispatched kanbans. De-EKCS depends on two parameters per stage: number of undispatched kanbans, K^j, and base stock level S^j. Initially, output buffer $B_1{}^j$ contains S^j finished parts having dedicated kanbans attached to them and queue $K_1{}^j$ contains K^j undispatched kanbans. All the other queues are empty.

The De-EKCS operates as follows:

1. When a customer demand for product type j arrives at the system, it is instantaneously transmitted to the two demand queues. If a part is available in output buffer $B_1{}^j$, it is released to the customer after the dedicated kanban is detached. The kanban is then transferred to undispatched kanban queue $K_1{}^j$; else the demand is backordered.
2. The other demand signal that joins demand queue D_1 signals the undispatched kanban in queue $K_1{}^j$ to be attached to a component from component buffer $B_0{}^j$. This part is then released into the MP. However, if there are no undispatched

kanbans available, the demand signal will wait.

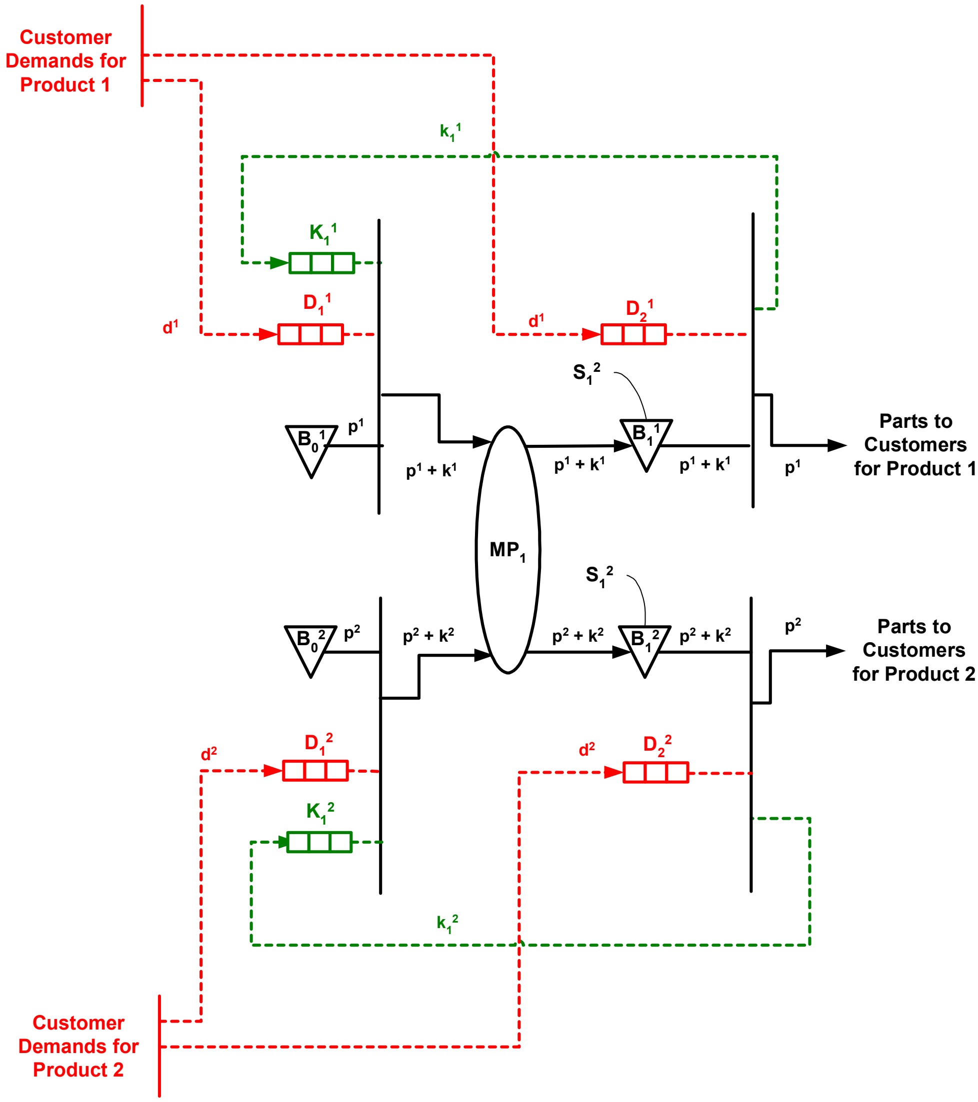

Figure 5. 9: Single Stage, Multiple Product, Dedicated Extended Kanban Control System (SS/MP/De-EKCS)

5.4.1 SS/MP/De-EKCS optimization

This section details a method to determine the optimal number of base stock, S*, and kanbans, K*, for each of the products in a SS/MP/De-EKCS. The method proposed is based on the Markov Chains technique–similar to the method used to optimize the MP/TKCS in Section 5.3.1.

5.4.1.1 Optimal base stock, S_j* and number of kanban, K_j*

The ETC equation for a SS/MP/De-EKCS can be written as (its derivation can be found in Appendix E):

$$ETC(S,K) = C_h\left(I[S,K]\right) + C_s\left(E\{t_{stockout}\}\right)$$

$$= C_h\left(S + \frac{K}{2}\right) + C_s\left(\frac{\lambda^{S+K}}{\sum_{x=0}^{S+K} \frac{(S+K)!}{(S+K-x)!} \lambda^{S+K-x} \rho^{x}}\right) \tag{5.2}$$

To optimize (5.2) with respect to S and K exhaustive enumeration is used, as with MP/TKCS. The main difference, however, is that optimizing MP/TKCS is easier since it only involves a single parameter K. But in MP/De-EKCS, two parameters are involved: S and K.

Equation (5.2) has two parts:

$$\text{Total holding cost} = C_h\left(S + \frac{K}{2}\right) \tag{5.3}$$

and

$$\text{Total shortage cost} = C_s\left(\frac{\lambda^{S+K}}{\sum_{x=0}^{S+K} \frac{(S+K)!}{(S+K-x)!} \lambda^{S+K-x} \rho^{x}}\right) \tag{5.4}$$

Total holding cost equation, (5.3), is an increasing function of S and K. However, an upper bound for S and K exists for its total shortage cost, equation (5.4). Therefore, the strategy would be to use the incremental search technique or exhaustive enumeration. Firstly, S+K is treated as the total WIP, and is incremented as a whole to obtain the lowest total shortage cost. Then, after obtaining the optimal S+K, each combination is searched individually of both S and K that add up to the optimal S+K. Finally, the optimal S* and K* is the pair that results in the lowest ETC for the MP/De/EKCS.

5.4.1.2 Manufacturing Example

An example helps to gauge the efficacy of SS/MP/De-EKCS optimization algorithm, the example discussed above in section 5.3.1.1 is used. The scenario is of two product type demand arrivals, namely product type 1 and 2. Product type 1 is assumed to be an expensive item while product type 2 is assumed to be an inexpensive item.

The main difference between this example and the one described in section 5.3.1.1 is the output. Referring to Figure 5.10 and 5.11, they are simple 2 dimensional, two axis graphs which show the ETC versus the number of kanbans for each product type. This is as expected since MP/TKCS takes into account only 1 parameter: the number of kanbans, K. But now that the system is the MP/De-TKCS, there are two parameters: number of kanbans, K, and number of base stock, S. The end result from MATLAB should give a 3 dimensional, three axis plot that shows the ETC on the Z axis, versus the number of kanbans, K and the base stock number, S on the X and Y axis. From the plot, a local optimal point could be derived. However, in this case, in order to specifically locate a discrete optimal K* and S* for this example, the MATLAB code (Appendix I5) used an incremental search technique or exhaustive enumeration. In other words, exhaustive enumeration was used first, followed by a 3D plot for confirmation.

The optimal base stock for product 1, S1* is 0 while the optimal number of kanbans for product 1, K1* is 2. The resulting lowest Expected Total Cost for product 1, ETC(S1*,K1*) is $15.69. A plot of ETC vs. base stock and kanban number is shown in Figure 5.8. As can be seen, the lowest ETC is at the point S1* = 0 and K1* = 2. Also, referring to Table 5.3, when S1 and K1* are 0 (that is there is neither base stock nor kanbans in the system), the ETC(S1,K1) is $100.

The optimal base stock for product 2, S2* is 0 while the optimal number of kanbans for product 2, K2* is 2. The resulting lowest Expected Total Cost for product 2, ETC(S2*,K2*) is $8.67. A plot of ETC vs. base stock and kanban number is shown in Figure 5.9. As can be seen, the lowest ETC is at the point S2* = 0 and K2* = 2.

Table 5. 2: Enumeration over S & K for SS/MP/De-EKCS – Product 1

S_1	K_1	S_1+K_1	$ETC(S_1,K_1)$
0	0	0	$100.00
0	1	1	$26.74
0	2	2	$15.69
0	3	3	$16.56
0	4	4	$20.43
1	0	1	$31.74
1	1	2	$20.69
1	2	3	$21.56
1	3	4	$25.43
2	0	2	$25.69
2	1	3	$26.56
2	2	4	$30.43
3	0	3	$31.56
3	1	4	$35.43
4	0	4	$40.43

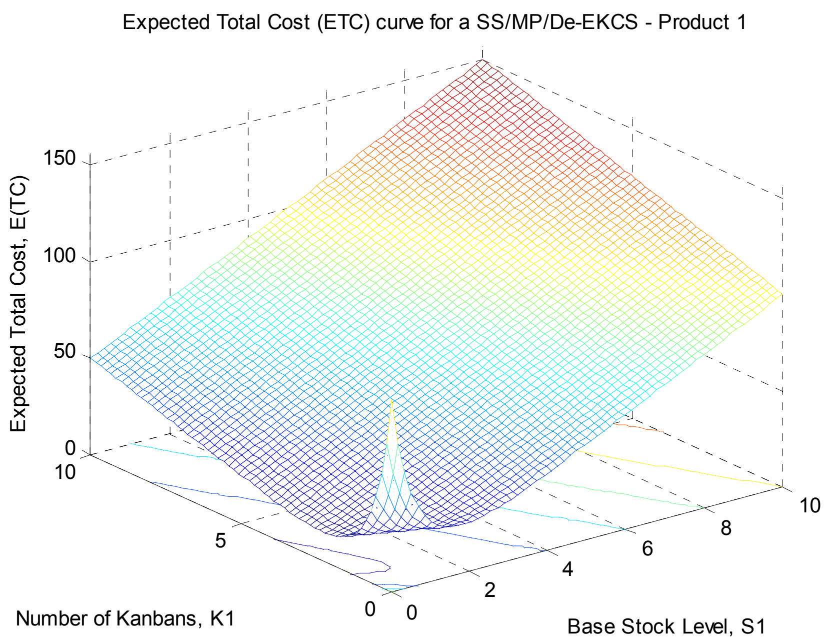

Figure 5. 10: Expected Total Cost (ETC) curve for a SS/MP/De-EKCS for Product 1

Table 5. 3:Enumeration over S & K for SS/MP/De-EKCS – Product 2

S_2	K_2	S_2+K_2	ETC(S_2,K_2)
0	0	0	$20.00
0	1	1	$10.00
0	2	2	$8.67
0	3	3	$9.49
1	0	1	$12.50
1	1	2	$11.17
1	2	3	$11.99
2	0	2	$13.67
2	1	3	$14.49
3	0	3	$16.99

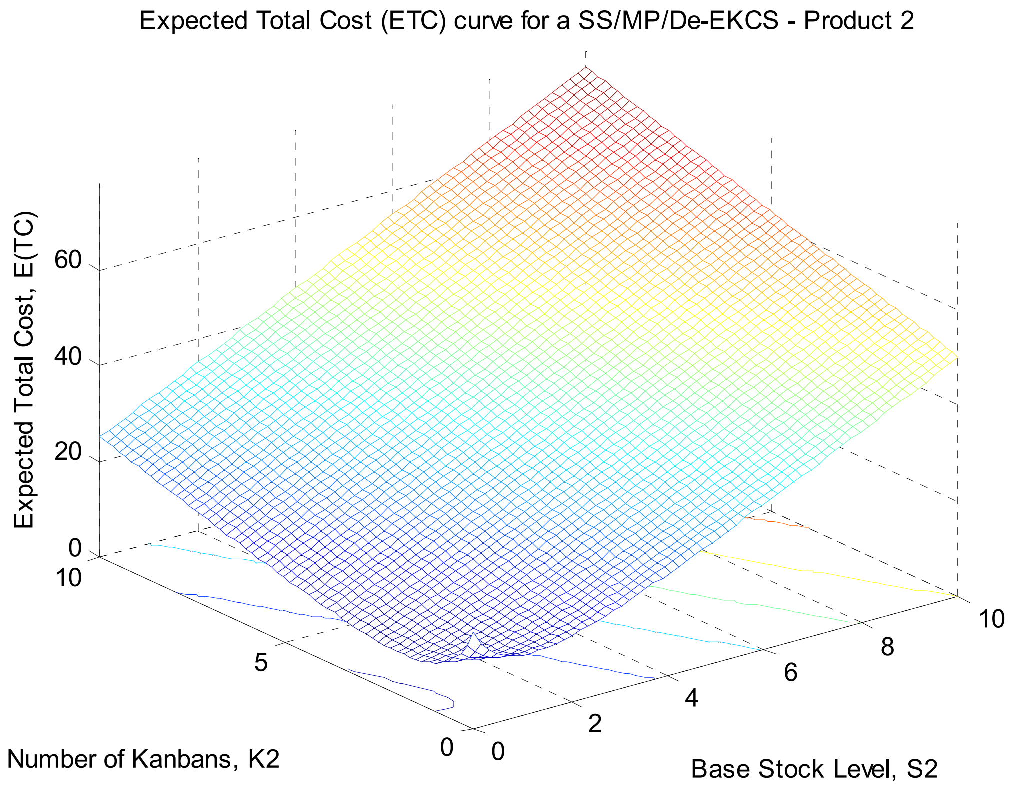

Figure 5. 11: Expected Total Cost (ETC) curve for a SS/MP/De-EKCS for Product 2

Note that the values shown in Figures 5.10 and 5.11 are as expected. Referring to equation 5.2, for example, if S_1 and $K_1 = 0$ (there is no product 1 in the system), the equation reduces to a simple shortage cost per unit for product 1, C_{s1}, which in this example is \$100. Looking at Figure 5.10, this also holds true (refer to its Z-axis – the ETC). This has to happen since there is no product 1 units that can satisfy its demands; hence the penalty/shortage cost of \$100. Likewise for product 2. Referring to equation 5.2 again, for example, if S_2 and $K_2 = 0$ (there is no product 2 in the system), the equation reduces to a simple shortage cost per unit for product 2, C_{s2}, which in this example is \$20. This can also be verified by looking at Figure 5.11.

In summary, this example gives an idea of what the optimal S_1*, K_1* and S_2*, K_2* should be for a specific case. Henceforth, these input parameters, λ_1, λ_2, μ_1 μ_2, C_{h1}, C_{h2}, C_{s1}, and Cs2, are varied and used in the simulation models of Chapter 6.

5.5 Single Stage, Multiple Product Shared, Extended Kanban Control System (SS/MP/Sh-EKCS)

In Figure 5.12, Buffer B_1^j is the output buffer for product type j and contains kanbans attached to finished parts. Buffer B_0^j is the raw materials buffer and is assumed to contain infinite number of components. Queue D_2^j contains incoming customer demand and queue K contains shared kanbans. Sh-EKCS depends on two parameters per stage: number of undispatched kanbans, K, and base stock level S^j. Initially, output buffer B_1^j contains S^j finished parts having kanbans attached to them and queue K contains K shared undispatched kanbans. All other queues are empty.

The Sh-EKCS operates as follows:

1. When a customer demand for product type j arrives at the system, it is instantaneously transmitted to the two demand queues. If a product is available in output buffer B_1^j, it is released to the customer after detaching the kanban. This kanban is then transferred to undispatched shared kanban queue K; else, the demand is backordered.
2. The other replicated demand that joins demand queue D_1^j signals the undispatched shared kanban in queue K to be attached to a component from component material buffer B_0^j. The component is then released into MP. However, if there are no undispatched shared kanbans present, this demand signal waits.

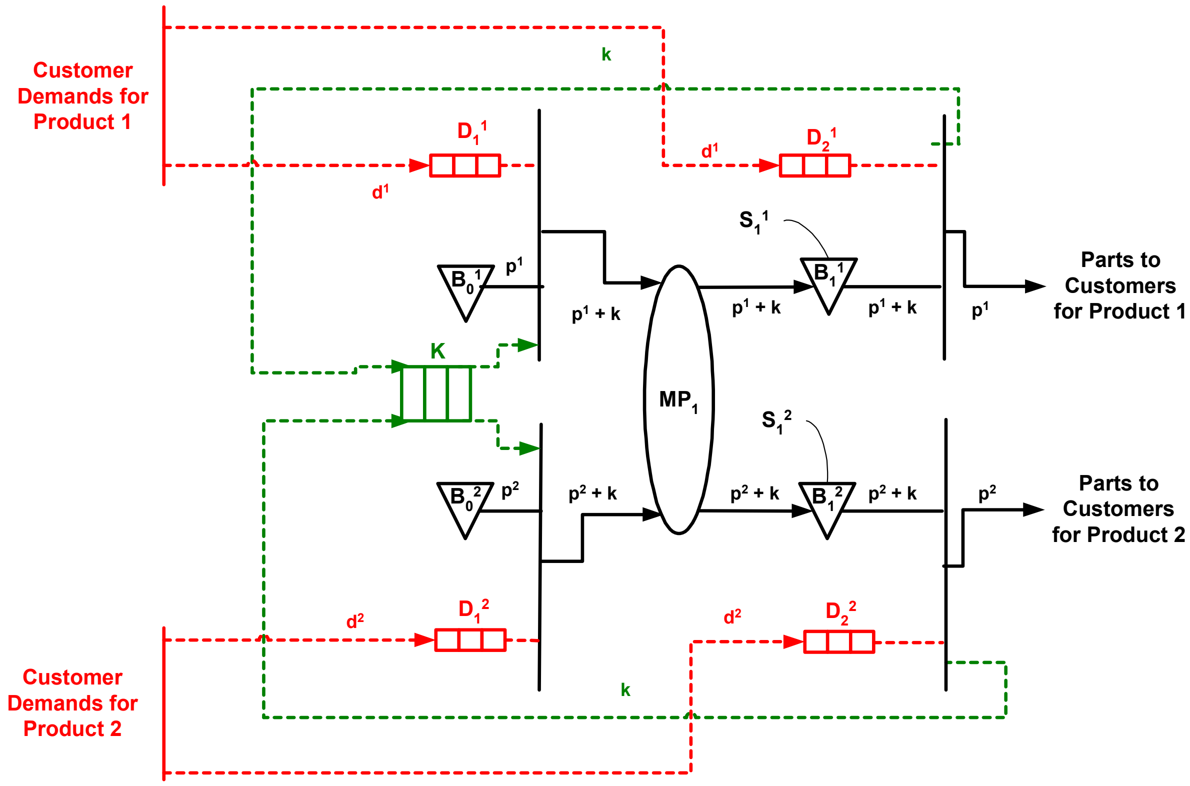

Figure 5. 12: Single Stage, Multiple Product, Shared, Extended Kanban Control System (SS/MP/Sh-EKCS) (In the figure, it should say: Product 1 (or 2) to customers)

5.5.1 SS/MP/Sh-EKCS optimization

Since kanbans are now shared in a SS/MP/Sh-EKCS, conventional techniques like ETC to obtain optimal S* and K* cannot be used. In dedicated kanban systems, holding cost, C_h and shortage costs, C_s can be computed separately for each product type, by minimizing ETC. However, in shared kanban systems, obtaining an optimal shared K* becomes difficult. In this section a new heuristic is developed to optimize the SS/MP/Sh-EKCS.

Considering MP/Sh-EKCS in its entirety, total WIP in the system is $S_1 + S_2 + K$, where K represents the number of shared un-dispatched kanbans in the kanban queue. Askin et al. (1993) and Ross (2007) discussed total number of jobs in process (or total WIP for all product types, i = 1, …, m) 'L' in an M/G/1 queue. The equation for L is shown in Figure 5.13: $L = S_1 + S_2 + K$. Now the utilization rate of each product type can be obtained as λ_i / μ. Next, L is apportioned according to the utilization rate; WIP (S+K) is assigned to a product type according to its utilization ratio. Note that $K_1$and K_2 here are not dedicated kanbans. They are actually the shared undispatched kanbans in the kanban queue K. However, they have the subscripts to represent the number of undispatched kanbans required to meet the demand arrival rate for the specific product type.

With the WIP for each product type obtained, ETC can then be used to obtain S* and K* for each product type. This is done to apportion base stock, S*, to each product type. ETC expression for Sh-EKCS is the same as De-EKCS (Appendix F). Finally, K_1* and K_2* are added to get the total shared K*. The flow chart for optimizing Sh/EKCS is shown in Figure 5.13.

Obtain the Total WIP, L, for the Sh/EKCS.

$L = S_1 + S_2 + K$

and

$$L = \frac{D^2/\mu^2}{\left(1 - D/\mu\right)} + \frac{D}{\mu}$$

↓

Obtain the utilization rate for each product type.

$$\rho_i = \frac{\lambda_i}{\mu}$$

↓

Portion the total WIP, L, according to the utilization rate for each product type, p_i

↓

For each product type, use ETC to obtain the optimal base stock, S*, and undispatched kanban number, K* to use.

That is,

$$ETC(S,K) = C_h\left(I[S,K]\right) + C_s\left(E\{t_{stockout}\}\right)$$

$$= C_h\left(S + \frac{K}{2}\right) + C_s\left(\frac{\lambda^{S+K}}{\sum_{x=0}^{S+K} \frac{(S+K)!}{(S+K-x)!}\lambda^{S+K-x}\rho^x}\right)$$

↓

After obtaining the optimal base stocks for both product types, S_1* and S_2*, sum up the individual K_1* and K_2* to get the optimal total shared kanbans, K*.

Figure 5. 13:Algorithm for optimizing the MP/Sh/EKC

5.5.1.1 Manufacturing Example

The heuristic presented above was coded in MATLAB to obtain optimal base stock, S*, and kanbans, K*, for a SS/MP/Sh-EKCS (Appendix I6). In order to test the effectiveness of the SS/MP/Sh-EKCS optimization algorithm, an example is used in this section. The example used here is described in section 5.4.1.2 above

Similar to section 5.4.1.2, the system taken into consideration is the MP/Sh-TKCS, there are two parameters: number of kanbans, K, and number of base stock, S. The end result from MATLAB should give a 3 dimensional, three axis plot that show the ETC on the Z axis, versus the number of kanbans, K and the base stock number, S on the X and Y axis. From the plot a local optimal point can be derived. However, in this case, in order to specifically locate a discrete optimal K* and S* for this example, the MATLAB code (Appendix I6) used an incremental search technique or exhaustive enumeration. In other words, exhaustive enumeration was used first, followed by a 3D plot for confirmation.

The optimal base stock for product 1, S1* is 0 while the optimal number of kanbans for product 1, K1* is 3. The resulting lowest Expected Total Cost for product 1, ETC(S1*, K1*) is $18.68. A plot of ETC vs. base stock and kanban number is shown in Figure 5.12. As can be seen, the lowest ETC is at the point S1* = 0 and K1* = 3. Also, referring to Table 5.6, when S1 and K1* is 0 (there is neither base stock nor kanbans in the system), the ETC(S1,K1) is $100.

The optimal base stock for product 2, S2* is 0 while the optimal number of kanbans for product 2, K2* is 2. The resulting lowest Expected Total Cost for product 2, ETC(S2*,K2*) is $9.47. A plot of ETC vs. base stock and kanban number is shown in Figure 5.13. As can be seen, the lowest ETC is at the point S2* = 0 and K2* = 2.

Table 5. 4:Enumeration over range of S & K for SS/MP/Sh-EKCS – Product 1

S_1	K_1	S_1+K_1	ETC(S_1,K_1)
0	0	0	$100.00
0	1	1	$33.00
0	2	2	$19.82
0	3	3	$18.68
0	4	4	$21.41
0	5	5	$25.55
1	0	1	$38.00

1	1	2	$24.82
1	2	3	$23.68
1	3	4	$26.41
1	4	5	$30.55
2	0	2	$29.82
2	1	3	$28.68
2	2	4	$31.41
2	3	5	$35.55
3	0	3	$33.68
3	1	4	$36.41
3	2	5	$40.55
4	0	4	$41.41
4	1	5	$45.55
5	0	5	$50.55

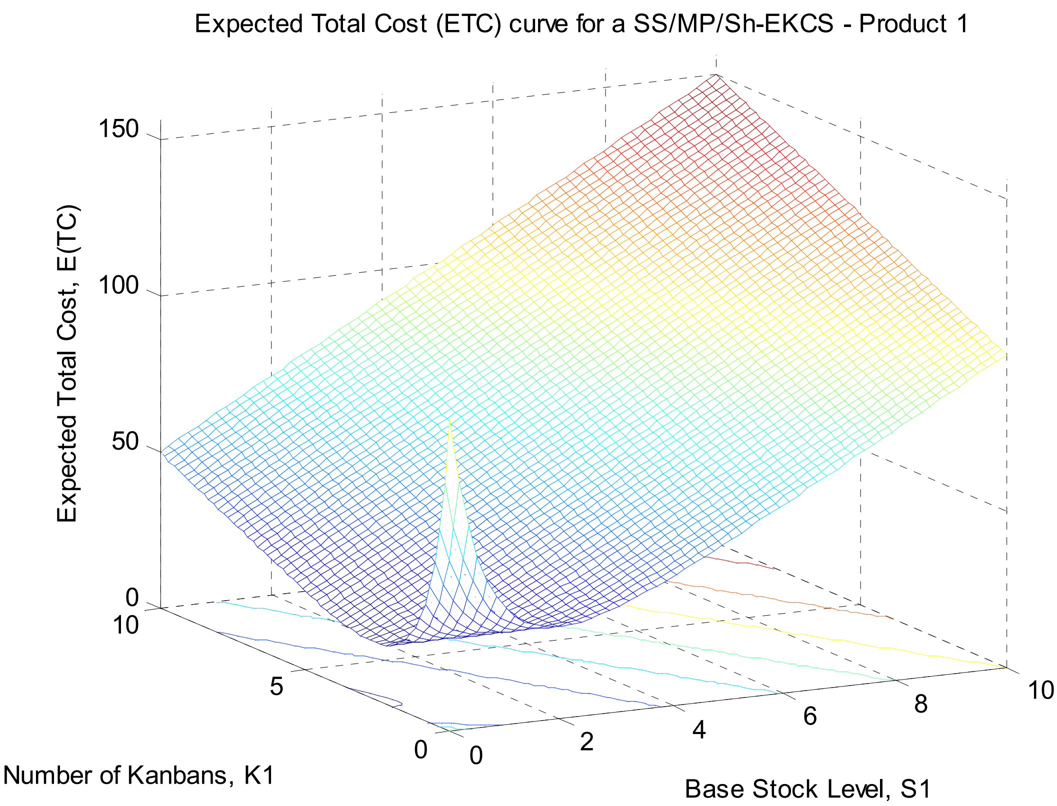

Figure 5. 14: Expected Total Cost (ETC) curve for a SS/MP/Sh-EKCS for Product 1

Table 5. 5: Enumeration over S & K for SS/MP/Sh-EKCS – Product 2

S_2	K_2	$S_2 + K_2$	$ETC(S_2,K_2)$
0	0	0	$20.00
0	1	1	$10.74
0	2	2	$9.47
0	3	3	$10.21
1	0	1	$13.24
1	1	2	$11.97
1	2	3	$12.71
2	0	2	$14.47
2	1	3	$15.21
3	0	3	$17.71

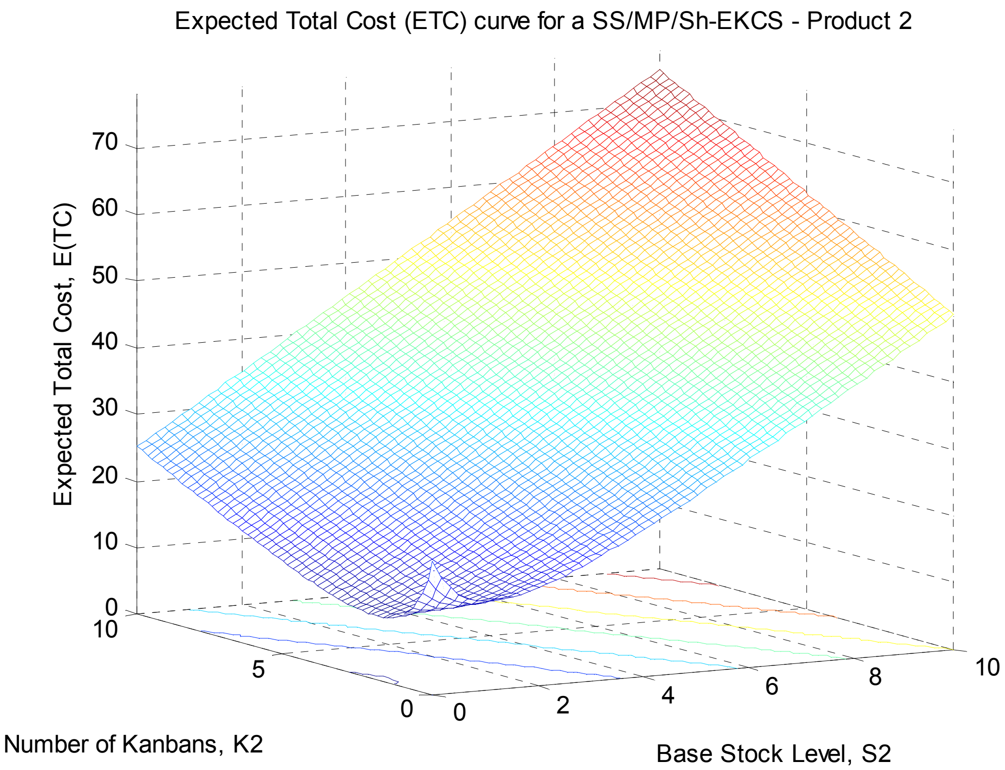

Figure 5. 15: Expected Total Cost (ETC) curve for a SS/MP/De-EKCS for Product 2

The values shown in Figures 5.14 and 5.15 are as expected. Referring to equation 5.2, similar to the equation in Figure 5.13, for example, if S_1 and $K_1 = 0$ (there is no product 1 in the system), the equation reduces to a simple shortage cost per unit for product 1, C_{s1}, which in this example is $100. Looking at Figure 5.14, this also holds true (refer to its Z-axis – the ETC), there will be no product 1 units that can satisfy its demands, hence the penalty/shortage cost of \$100. Referring to equation 5.2 again, if S_2 and $K_2 = 0$ (there is no product 2 in the system), the equation reduces to a simple shortage cost per unit for product 2, C_{s2}, which in this example is \$20. This can be verified by looking at Figure 5.15.

In summary, this example gives an idea of what the optimal S_1*, K_1* and S_2*, K_2* should be for a specific case. Henceforth, these input parameters, λ_1, λ_2, μ_1 μ_2, C_{h1}, C_{h2}, C_{s1}, and Cs2, are varied and used in the simulation models of Chapter 6.

5.6 Manufacturing Process (MP) for SS/MP/KCS

B. Baynat et al. (2002) first discussed the MP shown in Figures 5.1 to 5.12 in all proposed multi-product Kanban control systems. . No assumptions were made about either workstations of a stage (which could be a production line, a job shop or any flexible manufacturing system), or on the routing of parts. Their discussion of MP/KCS focused on external routing and not on internal processing of MPs. However, in the following it is important to discuss the internal mechanism of MP/KCS so that they can be compared using simulation.

5.6.1 M/M/1 with Priority Queues

Figure 5.16 shows what goes on inside the manufacturing process of an MP/KCS.

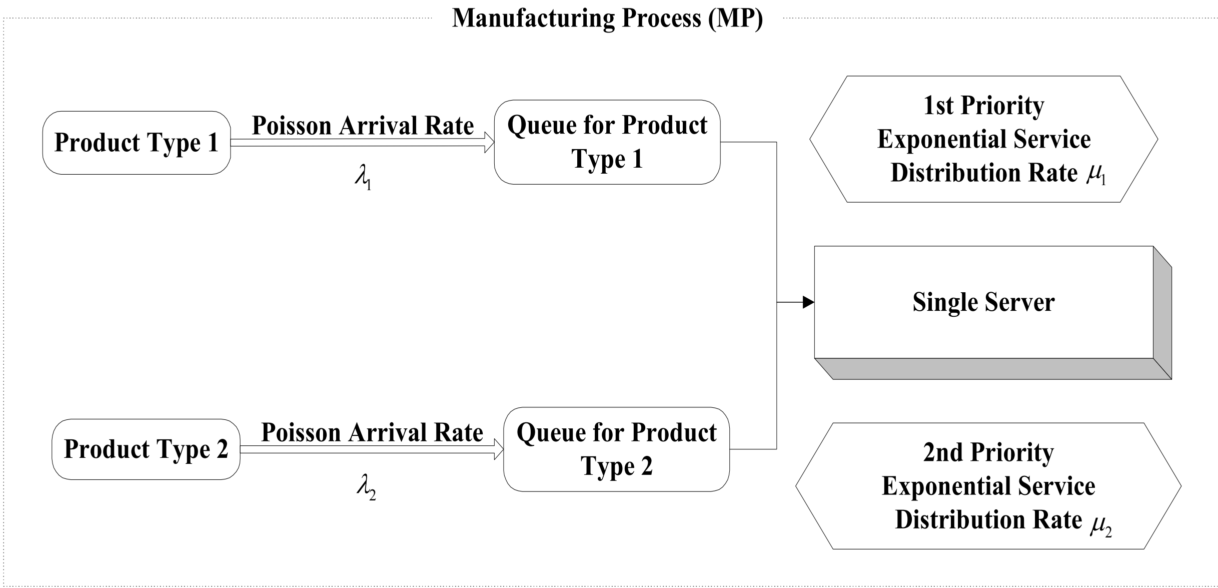

Figure 5. 16: Manufacturing Process (MP) as an M/M/1 queue with two priorities

The MP operates as follows:

- Type 1 and 2 demands have Poisson arrival rates of λ_1 and λ_2, respectively.

- Type 1 demand arrivals have higher priority.
- There is no pre-emption of service.
- Type 1 and 2 demands have exponential service distribution rates of μ_1 μ_2, respectively.
- $\lambda_1 < \lambda_2$: Demand arrival rate of type 2 is greater than type 1; else type 2 will never get processed.
- $\lambda_1 < \mu_1$: Standard queuing theory assumption is used; arrival rate of type 1 customers cannot be greater than the processing rate.
- $\lambda_2 < \mu_2$: Standard queuing theory assumption, arrival rate of Type 2 customers cannot be greater than its processing rate.
- $\lambda_2 < \mu_1$: Demand arrival rate for type 2 is smaller than processing rate for type 1. This assumption is needed to prevent system from getting choked with type 2 demands. Consider for example the reverse, $\mu_1 < \lambda_2$; each time a type 1 product enters, it has priority for processing. But it has a slower processing rate compared to type 2 arrivals, so that once it completes, the system would have a long queue of type 2 demands waiting and will be flooded with type 2 demands.
- $\lambda_1 < \mu_2$: Demand arrival rate for type 1 is smaller than MP rate for type 2. This assumption is similar to the one above and is needed to prevent the system from getting flooded with type 1 demands.

Since the law of queuing theory states that average demand arrival rate, λ_A, be lower than average Manufacturing Process (MP) rate, μ_A , that is $\lambda_A < \mu_A$; two possibilities exist:

i. $\lambda_1 < \lambda_2 < \mu_1 < \mu_2$

ii. $\lambda_1 < \lambda_2 < \mu_2 < \mu_1$

5.6.2 Average Manufacturing Process (MP) rate

The average MP rate is used in optimization models for MP/KCS. Figure 5.16 shows an M/M/1 system with priority queues of two product types. Ross (2007) used a General Service distribution time to compute the average arrival rate and distribution of arrivals as follows:

$$\lambda_A = \lambda_1 + \lambda_2 \tag{5.5}$$

$$G(x) = \frac{\lambda_1}{\lambda} G_1(x) + \frac{\lambda_2}{\lambda} G_2(x) \tag{5.6}$$

Equation (5.5) shows the average arrival rate of two Poisson arrival rates while Equation (5.6) shows average arrival distribution of two independent arrival distributions. This could either be the 1st or 2nd moment of Service Distribution i.e. $E[S_i]$ or $E[S_i^2]$ (where $E[S_i]$ is expected service time for customer i). Equations (5.5) and (5.6) hold since the combination of two independent Poisson Processes is itself a Poisson Process, whose rate is the sum of rates of the component processes. Also, the 1st and 2nd moments of the Exponential distribution are

$$E[S_i] = \frac{1}{\mu_i} \tag{5.7}$$

$$E[S_i^2] = \frac{2}{\mu_i^2} \tag{5.8}$$

Substituting Equations (5.7) and (5.8) into (5.6):

$$E[S_A] = \frac{\lambda_1}{\lambda} E[S_1] + \frac{\lambda_2}{\lambda} E[S_2] \tag{5.9}$$

Also

$$\frac{1}{\mu_A} = \frac{\lambda_1}{\lambda}\left[\frac{1}{\mu_1}\right] + \frac{\lambda_2}{\lambda}\left[\frac{1}{\mu_2}\right] \tag{5.10}$$

The validity of Equation (5.10) can be tested by a system with no type 2 customer arrivals, that is, $\lambda_2 = 0$. From Equation (5.5), $\lambda = \lambda_1$. Substituting this into Equation (5.10)

$$\begin{gathered} \frac{1}{\mu_A} = \frac{1}{\mu_1} \\ E[S_A] = E[S_1] \end{gathered} \tag{5.11}$$

This shows that without type 2 arrivals, the system behaves as a single product system, validating Equations (5.5) and (5.6). Continuing to obtain average MP rate:

$$\begin{aligned} \frac{1}{\mu_A} &= \frac{\lambda_1}{\lambda\mu_1} + \frac{\lambda_2}{\lambda\mu_2} \\ &= \frac{\lambda_1\mu_2 + \lambda_2\mu_1}{\lambda\mu_1\mu_2} \end{aligned} \tag{5.12}$$

And

$$\mu_A = \frac{\lambda \mu_1 \mu_2}{\lambda_1 \mu_2 + \lambda_2 \mu_1} \tag{5.13}$$

5.7 Summary

This chapter presents single-stage, multi-product kanban control systems: namely BS, De and Sh-TKCS and De and Sh-EKCS. Dedicated kanbans are assigned to a particular product type, whereas shared kanbans can be assigned to any product type–depending on which demand arrives first. Next, their optimization models are presented. These optimization models are compared numerically in the next chapter.

Then, manufacturing process for MP/KCS is defined. It consists of a single server having exponential processing times and allows two product types with Poisson process arrivals to enter with product type 1 having priority for service over product 2. Based on these assumptions, the average arrival and processing rates are defined and are used to optimize the MP/ KCS.

In the next chapter, MP/De and Sh-EKCS are investigated in detail

CHAPTER 6

PERFORMANCE COMPARISON OF MULTIPLE PRODUCT KANBAN CONTROL SYSTEMS (MP/KCS)

In Chapter 5, various MP/KCS systems were examined. Their operating schematics were drawn, and their operations explained. In addition, the various models to optimize these systems were presented. In this chapter, all SS/MP/KCS systems are simulated and compared, namely SS/MP/BS, TKCS, De- and Sh-EKCS. MATLAB programs (Appendix I) are used to obtain optimal parameters for the respective systems, such as optimal S* for MP/BS and optimal K* for MP/TKC, as well as optimal pairs of (S1*, K1*), (S2*, K2*) for De and Sh-EKCS. These optimal values are then applied into their simulation models. This chapter begins with the simulation parameters presented in section 6.1, followed by their simulation assumptions and snapshots in section 6.2. Section 6.3 presents the results and discussion. Section 6.4 validates the models using sensitivity analysis. Finally, conclusions and insights are presented in section 6.5.

6.1 Simulation parameters

Similarly to Chapter 5, a fictitious scenario is used here where demand arrivals take place for only two product types, namely product type 1 and 2. Product type 1 is assumed to be the expensive item, while product type 2 is an inexpensive item. For product 1, the holding cost, C_{h1} is \$10 per unit per day; the shortage cost, C_{s1} is \$100 per day; and the backorder cost, C_{b1} is \$100 per unit. For product 2, the holding cost, C_{h2} is \$5 per unit per day; the shortage cost, C_{s2} is \$20 per day; and the backorder cost, C_{b2} is \$20 per unit. This is summarized in Table 6. 1 below

Table 6. 1: Simulation parameters for SS/MP/KCS

Simulation Parameters	Product 1 Expensive Item	Product 2 Cheap Item
Holding cost (\$ per unit per day), C_h	10	5
Shortage cost (\$ per day), C_s	100	20
Backorder cost (\$ per unit), C_b	100	20

The key difference between these input parameters in Table 6.1 and Table 5.1

is the added backorder cost. This is due to the way the total costs are calculated. In Chapter 5, Expected Total Cost (ETC) equations were used, whereas here, Actual Total Costs (ATC) are used. As the names suggest, Expected Total Cost (ETC) equations are used to obtain an average or forecasted total cost value based on the scenario given. Minimizing this cost results in the optimal base stock level, S*, or optimal kanban number, K*, depending on which system is being studied. ATC is the sum of multiplying the actual outputs from the simulation models (like number of back orders, average inventory level, etc.) by their respective cost. That is,

Actual Total Cost (ATC) = (Average inventory level x C_{hi}) + (Number of backorders x C_{bi}) + (Expected shortage time x C_{si}), where i: product 1, 2.

ETC equations were discussed in Chapter 5. Put simply, ETC was used in Chapter 5 for optimization purpose, while ATC is used in this chapter for performance comparison purposes. Thus, even though the fictitious examples used in Chapters 5 and 6 are similar, there are also slight differences. In this chapter, ATC takes into account the backorder cost, while ETC does not. The optimization models used in Chapter 5 do not require backorder costs, however in order to reflect a more realistic actual scenario and to see which system performs the best, in this chapter they are incorporated into the total cost.

Moving on, referring to section 5.6.1, given that the MP is treated as an M/M/1 queue with two priorities, it was stated that two possibilities exist:

iii. $\lambda_1 < \lambda_2 < \mu_1 < \mu_2$

iv. $\lambda_1 < \lambda_2 < \mu_2 < \mu_1$

Therefore, in this chapter, two main scenarios are simulated: MP rate of product type 1 slower than the MP rate of product type 2, and vice versa. The average demand arrival rate, λ_A, and the average manufacturing process (MP) rate, μ_A, have to be calculated first (Appendix D); then, the percentage average utilization (λ_A/μ_A) is used to determine how "busy" the system is. Tables 6.3 and 6.4 show the simulation parameters for SS/MP/KCS for the two scenarios described. For simplification, only 50 to 90% utilization level is studied. This is because most of the time, only when the system being tested is in its "busiest" state can its peak performance be effectively compared against other systems.

Table 6. 2: Simulation parameters for SS/MP/KCS (MP rate of product 1 slower than product 2; 50 to 90% utilization)

Demand Arrival Rate for Product 1, λ_1	Demand Arrival Rate for Product 2, λ_2	Manufacturing Process (MP) Rate for Product 1, μ_1	Manufacturing Process (MP) Rate for Product 2, μ_2	Average Demand Arrival Rate, λ_A	Average Manufacturing Process (MP) Rate, μ_A	Percentage Average Utilization
(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(%)
1	2	3	8	3	5	0.58
2	3	6	9	5	8	0.67
3	4	9	10	7	10	0.73
4	5	10	11	9	11	0.85
5	6	11	12	11	12	0.95

Table 6. 3: Simulation parameters for SS/MP/KCS (MP rate of product 1 faster than product 2; 50 to 90% utilization)

Demand Arrival Rate for Product 1, λ_1	Demand Arrival Rate for Product 2, λ_2	Manufacturing Process (MP) Rate for Product 1, μ_1	Manufacturing Process (MP) Rate for Product 2, μ_2	Average Demand Arrival Rate, λ_A	Average Manufacturing Process (MP) Rate, μ_A	Percentage Average Utilization
(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(%)
1	2	8	5	3	6	0.53
2	3	9	7	5	8	0.65
3	4	10	9	7	9	0.74
4	5	11	10	9	10	0.86
5	6	12	11	11	11	0.96

6.2 Simulation assumptions and snapshots

Assumptions made in modelling SS/MP/KCS are:

- All the systems produce only two product types.
- Type 1 demand arrivals have higher priority; they are served first, before type 2 demand is served. There is no pre-emption of service. If a type 2 demand is being serviced while a type 1 demand arrives, the type 2 demand completes its processing before the type 1 demand is allowed to enter the MP.
- The one-card kanban system is adopted.
- The system does not produce defective parts.
- All systems consist of a single stage and one MP.
- Each MP contains only one machine or server.
- No setup times are required at the machine.
- Machine failures are ignored.
- Each machine can only process one part at a time.
- Transfer times for parts are negligible.
- Demand signals and kanbans flow instantaneously.
- Component buffers have an infinite supply of components.
- Demand arrivals follow a Poisson Process.
- Processing times at MPs are assumed to be exponentially distributed.
- Each replication is run for one year.
- Each simulation is replicated 10 times.
- The warm up period for each replication is three months.

In Appendix J2, Figure J4 to J7 show MP/KCS simulated in ARENA. The figures follow the schematics of the respective MP/KCS presented in chapter 5.

6.3 Simulation Results

Figure 6.1 and Figure 6.2 show the Actual Total Costs (ATC) of different Kanban Control Systems (KCS) being simulated; tables of results can be found in Appendix G. Figure 6.1 shows the case of MP rate of Product 1 *slower* than Product 2; Figure 6.2 shows the other case – MP rate of Product 1 *faster* than Product 2. Both cases showed similar results – BS incurred the highest ATC, followed by TKCS and De and Sh/EKCS. This is due to BS stocking up the most, hence incurring the highest total holding cost (refer to Table G7 and Table G28). The BS optimization method results in very high optimal base stock, S* values, which in return results in high inventory levels for both products.

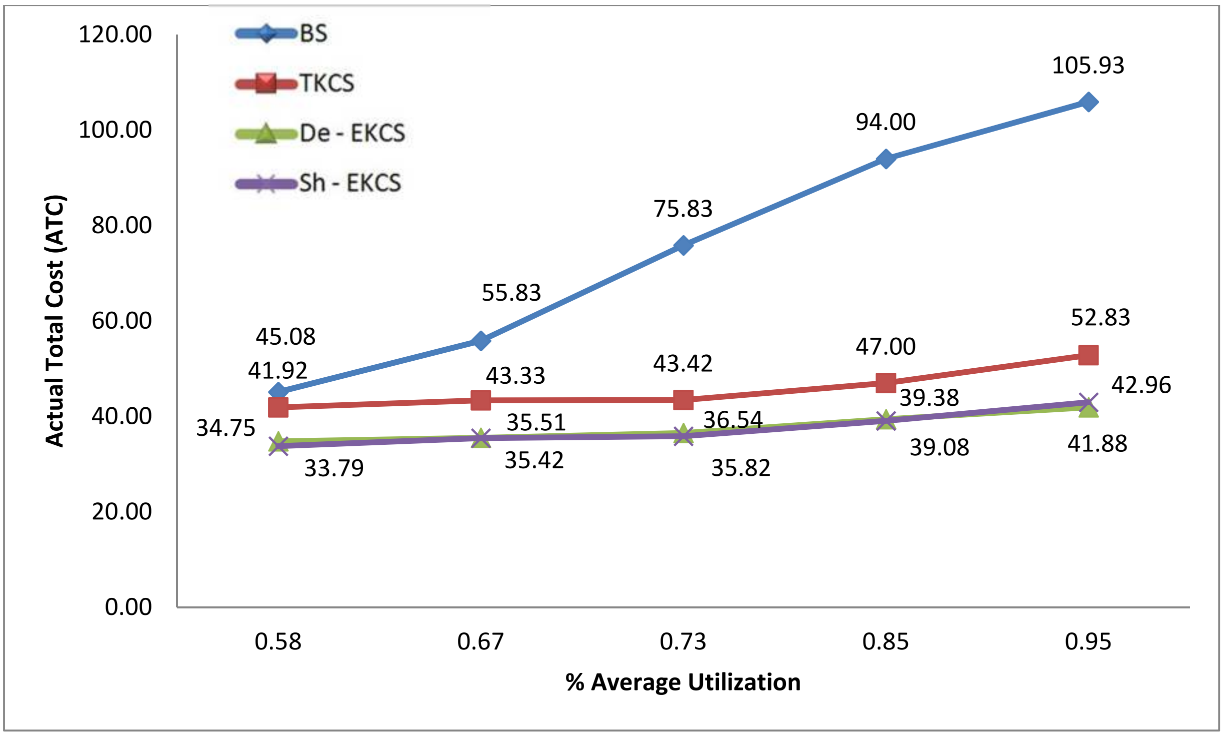

Figure 6. 1: Performance comparison of SS/MP/KCS (MP Rate of product 1 < product 2)

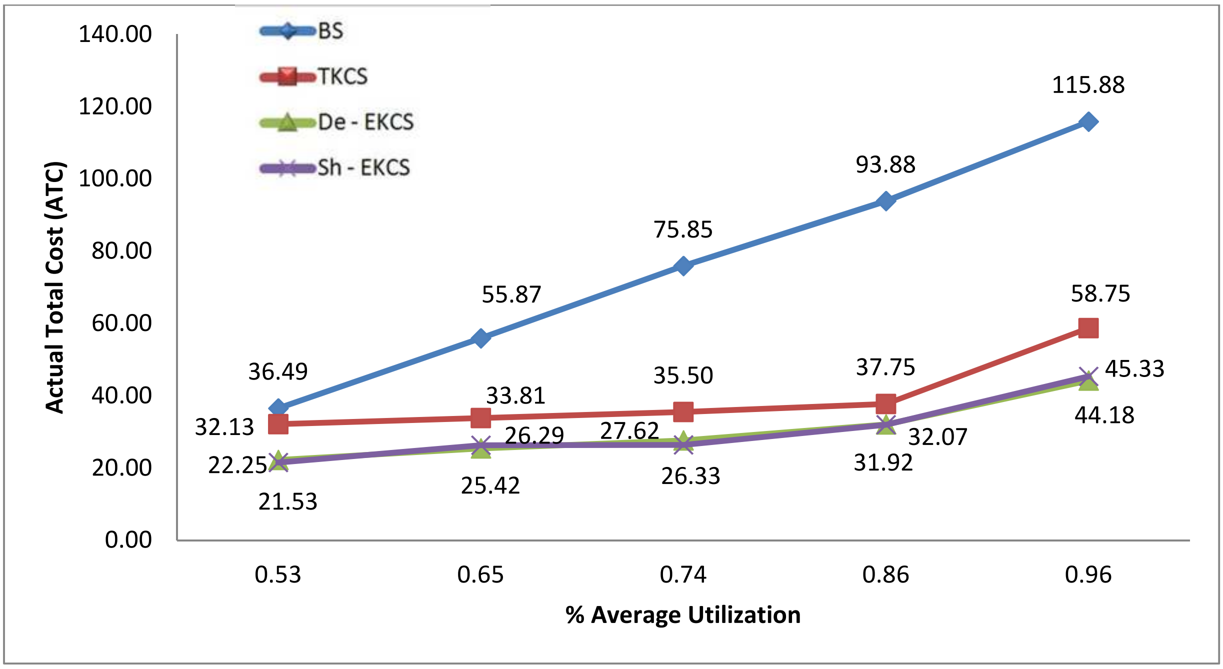

Figure 6. 2: Performance comparison of SS/MP/KCS (MP Rate of product > product 2)

Considering the performance of TKCS, De and Sh/EKCS, De and Sh/EKCS outperformed TKCS consistently by a gap of about \$8 to \$10 (for both Figures 6.1 and 6.2). This is due to TKCS consistently holding one 1 more unit of inventory than

De and Sh/EKCS. For the first case (MP rate of product 1 < product 2), comparing TKCS's average inventory level column in Table G8 to De and Sh/EKCS's (Tables G13 and Table G18, respectively), TKCS consistently held one more unit of inventory of product 1. Likewise, for the second case where MP rate of product 1 > product 2, comparing TKCS's average inventory level column of Table G29 to De and Sh/EKCS's (Table G34 and Table G38 respectively), the same thing is observed – TKCS consistently held 1 more unit of inventory of product 1 (except in the last row, where TKCS held 2 more).

Hence, the $8 to $10 gap seen in Figures 6.1 and 6.2 is due to TKCS holding 1 more stock of product 1 as compared to De and Sh – EKCS. Since product 1's holding cost is $10 per unit, this naturally accounts for the approximate $10 gap. So the next question is: why did TKCS holds 1 additional unit of inventory of product 1? The answer is that its optimal dispatched kanban number is generally 1 more than De and Sh/EKCS (Table G8, G13, G18, G29, G34 and G38). With 1 more stock, the average customer waiting time for TKCS product 1 was lesser than with De and Sh/EKCS. However, lowering the customer waiting time cannot reduce ATC as much as lowering the inventory (which is $10 per unit for product 1). Hence, although De and Sh/EKCS have slightly longer customer waiting times, their undispatched kanbans help to make up for the loss of WIP, thereby justifying the reduced stock.

Overall, the results showed that De and Sh/EKCS outperformed the other systems studied. This is very similar to the performance comparison done earlier on SS/SP/KCS, where performance of BS was the worst, followed by TKCS and EKCS. The interesting finding here is the De and Sh/EKCS performed very much alike, with ATCs close to each other. Hence, the question that naturally arises is whether the operating behaviour of De and Sh/EKCS are really the same. To ascertain their differences, further investigation is carried out, simulating De and Sh-EKCS from the 10 to 50% utilization range.

6.3.1 Comparing De and Sh/EKCS (10 to 50% Utilization Rates)

To determine whether differences exist between De and Sh/EKCS, further simulations are carried out, over the 10 to 50% utilization range. Simulation parameters are presented below in Table 6.4.

Table 6. 4: Simulation parameters for SS/MP/De and Sh-EKCS (comparing 10 to 50% utilization)

Demand Arrival Rate for Product 1, λ_1	Demand Arrival Rate for Product 2, λ_2	Manufacturing Process (MP) Rate for Product 1, μ_1	Manufacturing Process (MP) Rate for Product 2, μ_2	Average Demand Arrival Rate, λ_A	Average Manufacturing Process (MP) Rate, μ_A	Percentage Average Utilization
(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(%)
1	2	16	21	3	19	16
2	3	17	22	5	20	25
3	4	18	23	7	21	34
4	5	19	24	9	21	42
5	6	20	25	11	22	49

Figure 6.3 **c**ompares De and Sh/EKCS with 10 to 50% utilization rates. Tables of results can be found in Appendix H. This range of utilization was chosen to examine whether the two systems continue to perform similarly during times of low utilization. Interestingly, Figure 6.3 also showed that there is no significant difference between the two systems' performance.

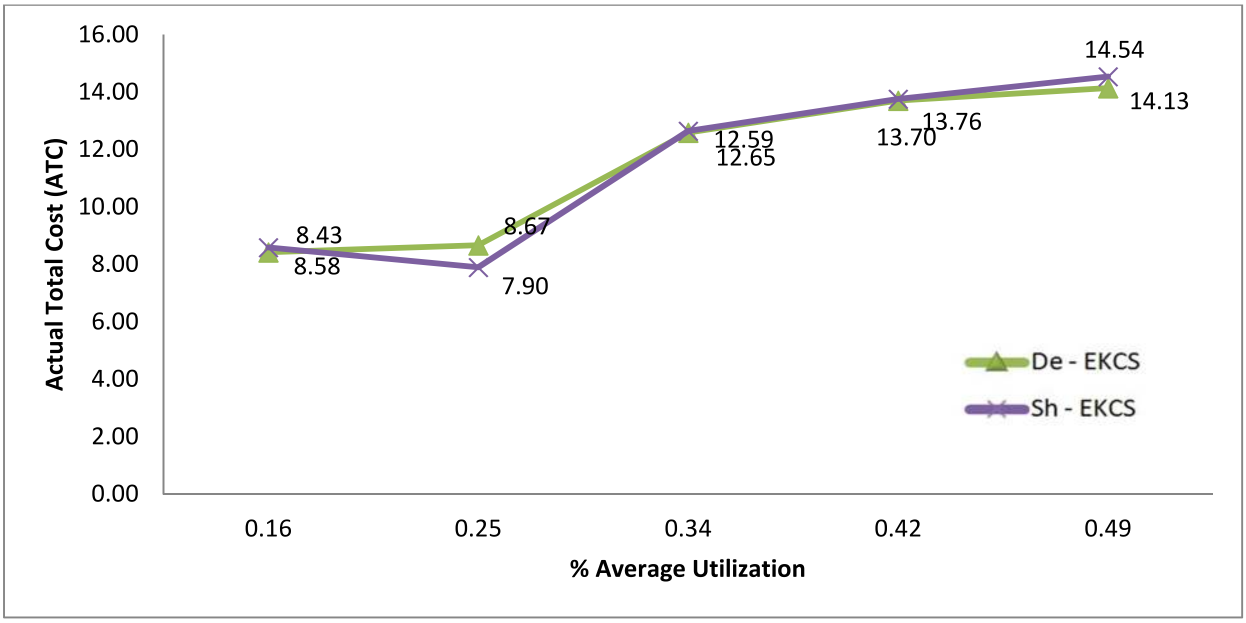

Figure 6. 3: Performance comparison of SS/MP/De and Sh/EKCS (10 to 50% Utilization)

6.4 Sensitivity Analysis

"Sensitivity analysis is the study of how the uncertainty in the output of a mathematical model or system (numerical or otherwise) can be apportioned to different sources of uncertainty in its inputs" (Saltelli, 2008). In this section, sensitivity analysis is conducted on the simulated results for SS/MP/KCS (for the specific scenario of product type 1's MP rate > product type 2 at 50 to 90% utilization rates). Appendix G3 shows the tables of results before sensitivity analysis was conducted while Appendix G2 shows the results afterwards. Overall, the results showed that the proposed optimal base stock, S*, and kanban number, K* were not too far off the simulated optimal values, with a few exceptions. This is a good sign since it demonstrates that the proposed optimization algorithms work well and are robust. An explanation for the few exceptions, which were quite far off from the optimal values, is also provided.

6.4.1 Steps involved in sensitivity analysis

In the sensitivity analysis, two key parameters are considered: number of Kanbans and base stock, and their effect on ATC observed. If, at any time in the sensitivity analysis, the ATC is found to be lower than the proposed optimal values, the lower values replace the optimal ones. Finally, a graph of different ATCs is plotted to show the gap between proposed and optimal results.

Figure 6.4 shows the result of sensitivity analysis conducted on MP/BS. Values are taken from Table G28 and Table G47. As observed, there was not much difference, since the BS optimization algorithm gives very high values of base stocks, S*, for both products.

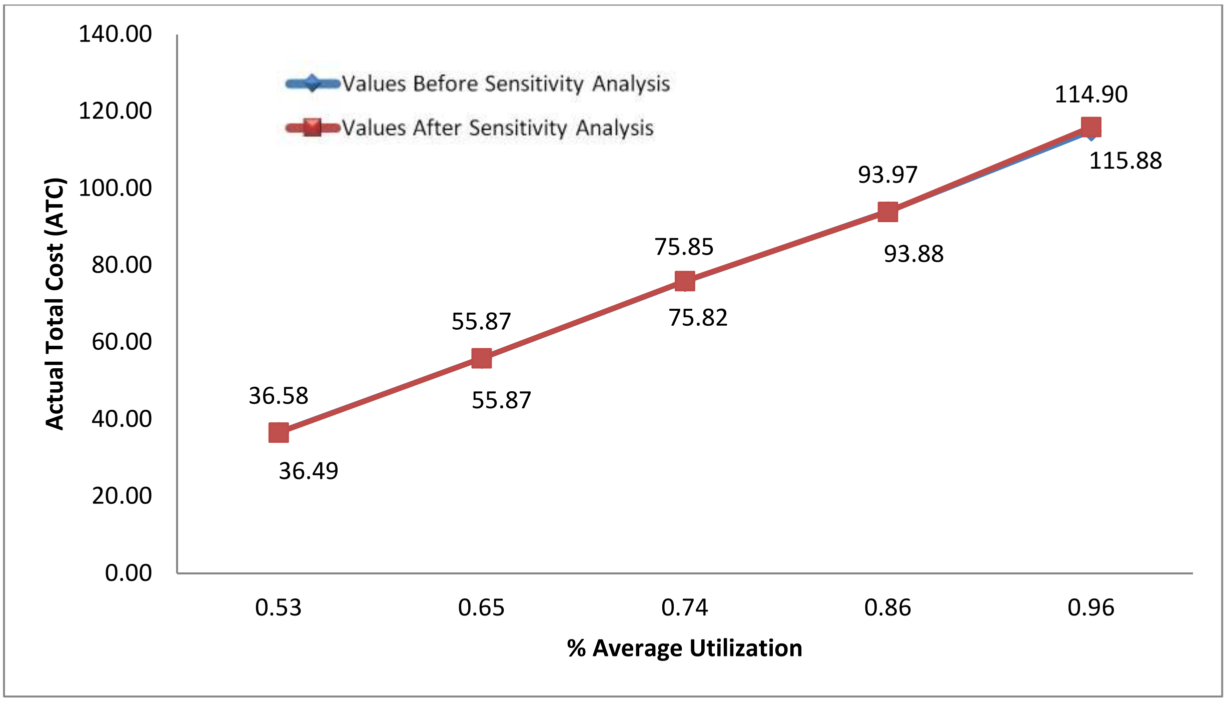

Figure 6. 4: Sensitivity analysis of SS/MP/BS for case of MP rate of product 1 > product 2 (50 to 90% utilization)

Figure 6.5 shows sensitivity analysis conducted on MP/TKCS. Values are taken from Tables G33 and G52. Note that significant differences occur at 53% and 96%, at both ends of the graph since backorders occur for each case (Table G48). The reason for the backorders is the lack of kanbans. Before sensitivity analysis, Table G48 shows optimal kanban numbers for product 1, K*, to be 1 (53%) and 3 (96%). After sensitivity analysis, Table G29 shows K* for product 1 to be 2 and 4 respectively. Since the backorder cost for product 1 is $100, it explains the huge jump in ATC in Figure 6.5.

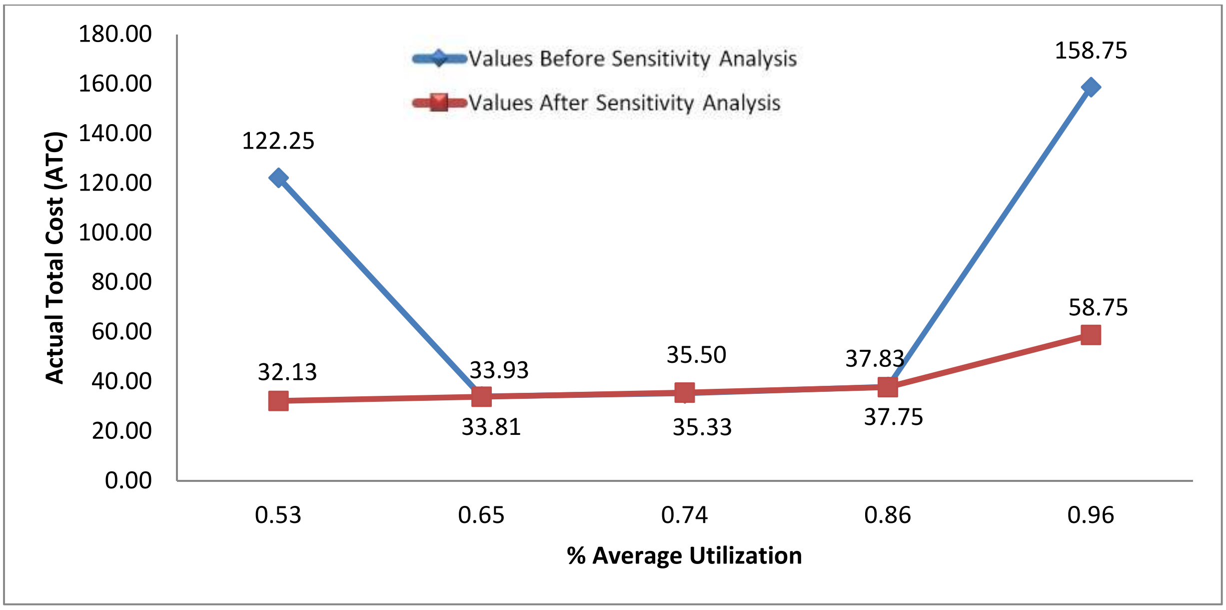

Figure 6. 5: Sensitivity analysis of SS/MP/TKCS for case of MP rate of product 1 > product 2 (50 to 90% utilization)

Figure 6.6 shows result of sensitivity analysis conducted on MP/De/EKCS. The values are taken from Tables G37 and G57**.** A major difference occurs at 96% since a backorder occurs for product 1 (Table G53). The reason for the backorder is the lack of dispatched kanbans (or base stock level). Before sensitivity analysis, Table G53 shows the optimal base stock number for product 1, S*, to be 2, while afterwards, it is 3. Since the backorder cost for product 1 is $100, it justifies the jump in ATC in Figure 6.6.

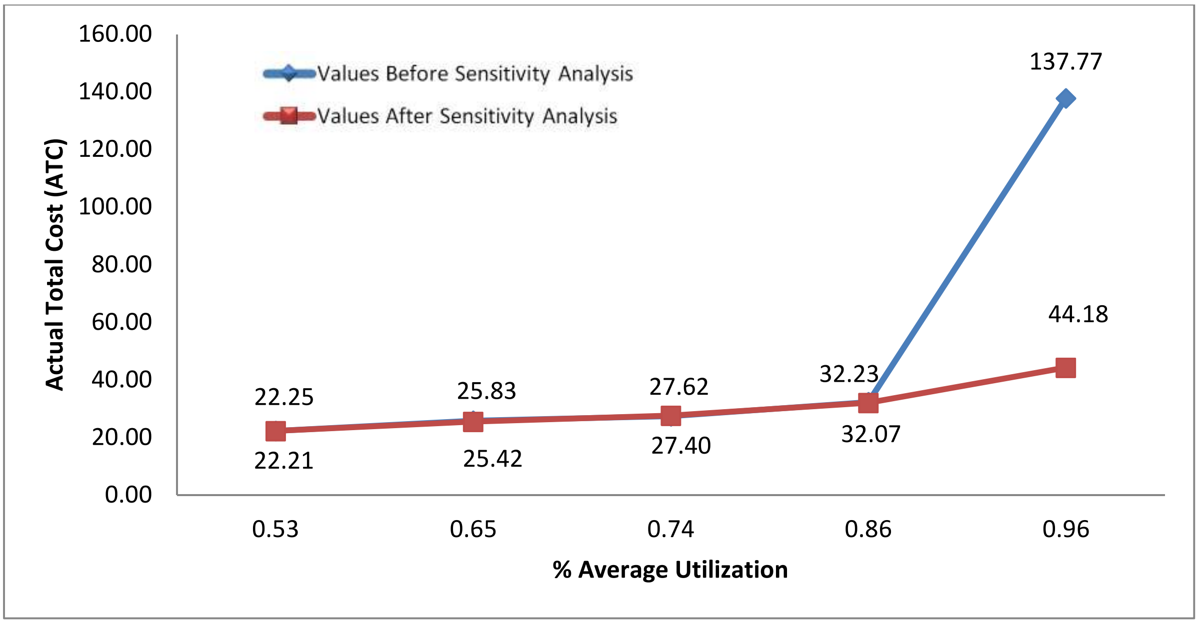

Figure 6. 6: Sensitivity analysis of SS/MP/De/EKCS for case of MP rate of Product 1 > Product 3 (50 to 90% utilization)

Figure 6.7 shows the results of sensitivity analysis conducted on MP/Sh/EKCS; the values were taken from Tables G42 and Table G62. A major difference is noticed at 96% since a backorder occurs for product 2 (Table G60). The reason for the backorder is the lack of dispatched kanbans (or base stock level). Before sensitivity analysis, Table G60 shows the optimal base stock number for product 1, S*, to be 3 (96%), whereas afterwards it is 4. Since the backorder cost for product 1 is $10, this justifies the jump in ATC in Figure 6.7.

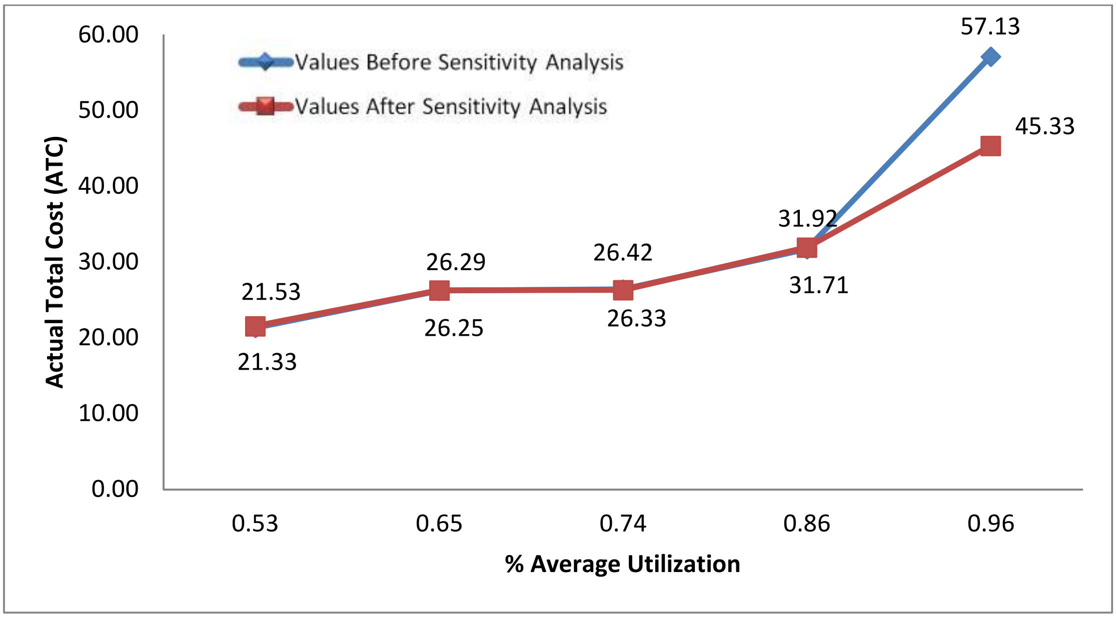

Figure 6. 7: Sensitivity analysis of SS/MP/Sh/EKCS for case of MP rate of product 1 > product 2 (50 to 90% utilization)

Overall, sensitivity analysis shows that the proposed optimal base stock, S*, and kanban number, K* generated from MATLAB are not far off from the true optimal values – with a few exceptions. This means that proposed optimization algorithms perform as expected.

6.5 Conclusion and Insights

In this chapter, the performance of the different types of SS/MP/KCS is compared. Firstly, MATLABMATLAB is used to obtain optimal parameters for the respective systems, such as optimal S* for MP/BS and optimal K* for MP/TKCS. Then, these optimal values are used in the simulation models of MP/KCS to compare their performance. Thereafter, sensitivity analysis is carried out to study the effect of two key parameters.

It is found that De and Sh/EKCS outperform the rest, with BS performing the worst, followed by TKCS. The most interesting finding is that De and Sh/EKCS actually perform the same. In other words, simulation has shown that their operating mechanisms are similar despite differences in schematics. But this was not clear when B. Baynat et al. (2002) first proposed the MP/EKCS. The main reason they behave alike is because of the priority assumption. Hence, for managers who wish to adopt the MP/EKCS, there is no need to consider shared kanbans. In order to demonstrate why De and Sh/EKCS are equivalent, a simple scenario is discussed below.

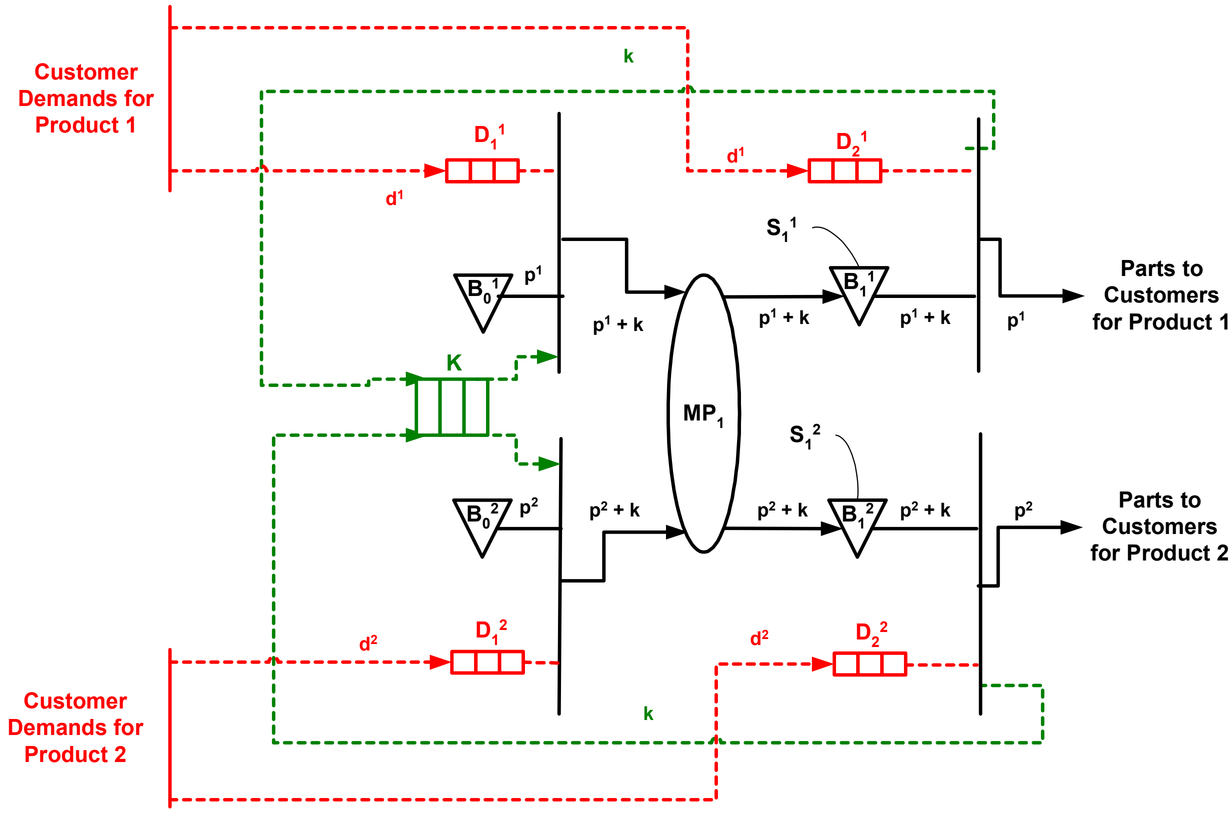

Figure 6. 8: Single Stage, Multiple Product, Shared Extended Kanban Control System (SS/MP/Sh/EKCS)

Referring to Figure 6.12, a shared kanban acts as though it has been "dedicated" when a customer arrives, because the MP can only process one customer at a time and the product with the highest priority goes first. This means that despite the shared kanbans being pooled together in the shared kanban queue, they each have been *pre-assigned* to a particular product. Consider the following scenario:

1. Assume that S_1^1 and S_1^2 each have 1 base stock (and hence attached kanban) in their respective output buffers B_1^1 and B_1^2. Also, assume that there are no shared kanbans in the kanban queue K (so that the effect of shared kanbans being *dedicated* can be seen).

2. Assume now that 1 customer demand per product appears at the same time. Instantaneously, each of them is satisfied with the products on hand while each of their kanban is detached and sent back to the shared kanban queue K. Take note the kanbans are no longer *dedicated*, but can be shared by any product demand that comes first.
3. However, upon entering queue K, these 2 kanbans are immediately attached to the respective component part, according to the previous customer demands that had just arrived, and sent into MP for processing.
4. Since the priority rule is such that product 1 has precedence over product 2, it is processed first. Hence, despite the fact that the two kanbans are *shared*, the priority rule set forth ensures that every customer demand arrival is accompanied by a *dedicated* kanban.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this research, different types of kanban systems are described and categorized according to their operating behaviour. Three significant pull systems, namely BS, TKCS and EKCS, are investigated, with EKCS being a hybrid of BS and TKCS. Previous studies only focussed on the qualitative differences among these systems. This research studies the quantitative performance differences of these systems and compares the performance of EKCS against BS and TKCS.

This study focuses on both single and multiple product kanban systems, assuming a single stage and a single server. MATLAB is used to model and optimize these systems and simulation (using ARENA version 12) with different initial parameter settings is used to validate the models. A performance comparison is done using KPIs such as fill rate, average inventory and average customer cycle time. The results show that under all scenarios, EKCS outperforms its predecessors, TKCS and BS.

There are four key contributions of this research. First, a method is proposed to determine the optimal number of base stock, S*, and number of kanbans, K*, in a SS/SP/EKCS. Second, a performance comparison is done using simulation to demonstrate that EKCS outperforms TKCS and BS in both single and multiple product scenarios. Third, the optimization method used for SS/SP/EKCS is generalized for use in both MP/De and Sh/EKCS. This is a novel contribution. Lastly, it is shown that De and Sh/EKCS are equivalent, and operate the same way even though their schematics are different.

Despite constant praise for the performance of EKCS in kanban literature, this thesis shows that it outperforms its predecessor, TKCS, by only a slight amount, while confirming the worth of TKCS, which has been in use for many decades. The worst-performing system is BS, as it stocks up too much inventory in almost all scenarios. Thus, this research shows that lean or pull production is still more effective than push. Factory floor managers still utilizing BS as their production control strategy should consider upgrading to TKCS, switching to EKCS only for special situations.

7.1 Insights from SS/SP/KCS Comparison

In SP/KCS comparisons, BS, TKCS and EKCS are tested using three KPIs: average customer waiting time, average inventory level and number of backorders. All three KPIs are combined into a total cost, Actual Total Cost (ATC). BS was shown to have the worst performance, incurring the highest ATC as it stockw up too much, disregarding MP rate and putting emphasis only on demand arrival rate. TKCS and EKCS perform well, with EKCS outperforming TKCS only by a small amount. EKCS is found to be only beneficial during low demand arrival and low backorder and shortage cost scenarios. In fact, the most important discovery made is that EKCS behaves similar to TKCS once it has base stock. However if EKCS is stockless, it can outperform TKCS when the number of un-dispatched kanbans is set to one.

This unique scenario for EKCS, in practice, is more commonly seen in high-end luxury items. For example, luxury cars tend to be customized, have extremely low demand, and manufacturers (or "crafters") do not produce them unless they receive orders. These cars are also different from normal cars in that they have low backorder and shortage costs compared to holding costs (Glover, 1999). Manufacturers for luxury cars, such as Ferrari, care more for their image than about mass production. In other words, their concern is not so much about backorders and long customer waiting time; rather, having too many Ferraris on the roads which may lower their image. Hence they even create "waitlists" for customers who wish to purchase their cars (Sardi, 2009).

Another example of exclusivity would be luxury items such as handbags, Hermes chooses not to open up too many retail stores so as to protect their exclusive image (Times, 2012). EKCS performs best only during low backorder and shortage costs scenario, which fits the above-described scenario well.

For economical cars such as Toyota or Honda, however, TKCS would still be the preferred system for managing production, as these cars have medium to high backorder and shortage costs compared to holding costs. Manufacturers of mass-produced products need to hold stock, and cannot afford to keep their customer waiting. Also, their cars usually are in high demand, requiring high utilization of manufacturing resources. Thus, TKCS performs best during medium- to high-backorder and shortage cost scenarios, coupled with high utilization rates of more than 50%.

As for BS, the results show that TKCS outperforms it in every scenario. This confirms that lean production seems to work best for mass produced products.

7.2 Insights from SS/MP/KCS Comparison

BS is again shown to have the highest cost of all the three systems. TKCS, De-EKCS and Sh-EKCS all fare well, with De- and Sh-EKCS outperforming TKCS slightly. However, the most important finding is that De- and Sh-EKCS are equivalent (under specific assumptions). This finding is important, as it reduces the need for future factory managers to explore the shared EKCS option. This important facet of multi-product KCS was left unexplored by proposers of the MP-EKCS (B. Baynat et al., 2002), and has not been investigated until now.

7.3 Future Work

This research can be extended by study of the Simultaneous Extended Kanban Control System (SEKCS) proposed by Chaouiya et al. (2000). Analyzing this system is important as it represents a new configuration, which takes into account multiple tiers and stages, in both horizontal and vertical directions. MPs are placed either in tandem or in parallel to each other. Although SEKCS is more complicated than EKCS, this system enacts real life situations better.

Chaouiya et al. (2000) further extends the EKCS and describes an assembly configuration with a tree-structured topology having manufacturing and assembly cells (Figure 7. 1). Basically, the authors are trying to enact a situation with many tiers, from manufacturing cells to assembly cells.

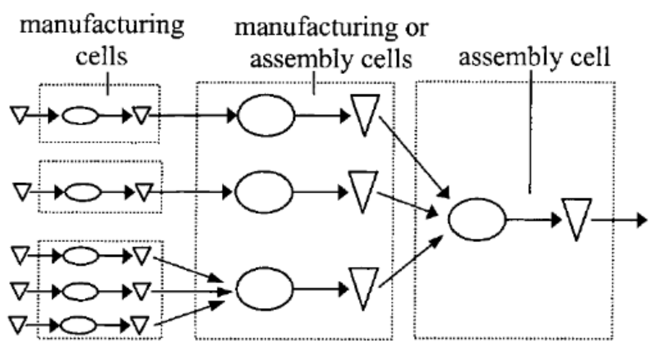

Figure 7. 1: General topology for assembly manufacturing systems (Chaouiya et al., 2000)

For simplicity, they restrict their study to assembly systems having only two tiers, and three manufacturing cells supplying a single assembly cell. However, in this research, to simplify their model further, SEKCS is limited to only two tiers, with two manufacturing cells supplying a single assembly cell. For a clearer picture, refer to Figure 7.2 below.

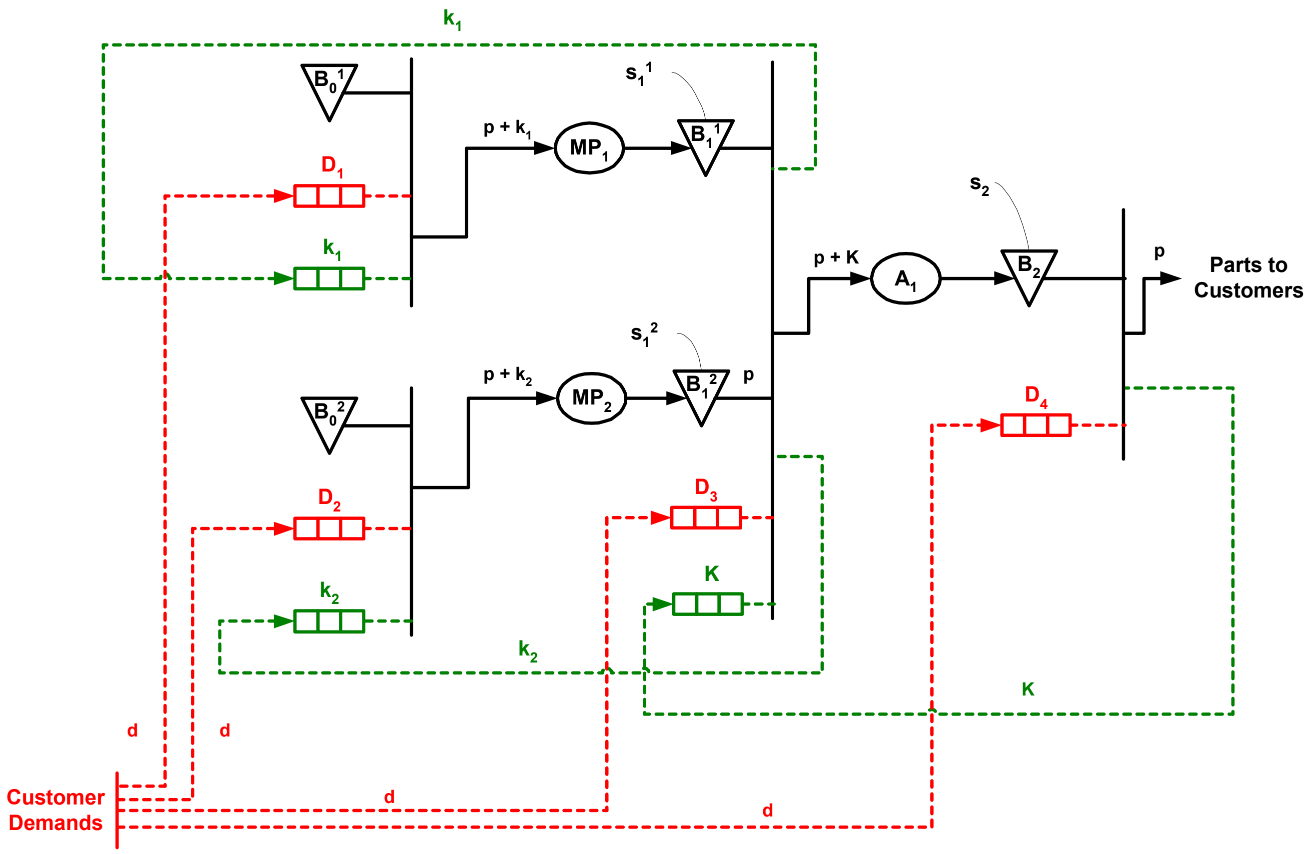

Figure 7. 2: Queuing network model of SEKCS

Figure 7.2 shows a queuing network model of a two-tier, two manufacturing cell, MP_1 and MP_2, supplying to one assembly cell, A_1. Table 7.1 below shows the initial state of the queues.

Table 7. 1: Initial state of the queues in SEKCS

Queue	Initial State Content
Manufacturing Process MP_1, MP_2 and Assembly Process A_1	0
Raw materials buffer, B_0^1 and B_0^2	Infinity
Customer demand queue, D_1, D_2, D_3, D_4	0
First Manufacturing Process's Kanban queue, k_1	$K_1 - S_1^1$

Second Manufacturing Process's Kanban queue, k_2	$K_2 - S_2^2$
Assembly Kanban queue, K	$K - S_2$
First tier MP output buffer, B_1^1	S_1^1 number of base stock (each attached a k_1 kanban)
Second tier MP output buffer, B_1^2	S_1^2 number of base stock (each attached a k_2 kanban)
Assembly output buffer, B_2	S_2 number of base stock (each attached a K kanban)

7.3.1 SEKCS Operation

When a customer demand arrives to the system, it is immediately transmitted to all demand queues by adding one to the queues D_i ($i = 1, 2, 3, 4$). Then the following occurs:

Delivery of finished part to the customer

At the assembly cell, assembly output buffer, A_1 and demand queue D_4 are joined in a synchronization station. The delivery of a finished assembled part can occur as soon as there is a pair S_2 in A_1 and a demand in D_4. When these conditions are satisfied

(1) Kanban K is detached from a part in S_2 and is transferred upstream to assembly kanban queue, K.
(2) A part from S_2 is released to the customer.
(3) Demand D_4 is satisfied.

Release of parts into the assembly cell

At the assembly cell A_1, MP output buffers B_1^1, B_1^2, demand queue D_3 and kanban queue K are joined in a synchronization station. This means that the assembly operation can begin only when there is at least one part in B_1^1, B_1^2, one demand waiting in queue D_3 and an assembly kanban in queue K. When these conditions are satisfied:

(1) Individual kanbans are simultaneously detached from the parts in B_1^1, B_1^2 and are transferred upstream to their corresponding kanban queues, k_1 and k_2.

(2) The assembly kanban K is attached onto the parts from B_1^1, B_1^2 , which are grouped together to be transferred to assembly station A_1 as a pair.

(3) Demand D_3 is satisfied.

Release of parts into the manufacturing cells

At each individual raw material cell, queues B_0^1, D_1 and K_1 are joined in a synchronization station; likewise for B_0^2, D_2 and K_2. This means that MP_1 can only begin producing a part when there is at least one part in B_0^1, one demand waiting in queue D_1 and a kanban in queue k_1; likewise for MP_2. When these conditions are met:

(1) Kanban k_i (i = 1,2) is attached to part p_i (i = 1, 2) and together they are transferred downstream to MP_i (i = 1, 2),

(2) Demand D_i (i = 1, 2) is satisfied.

This system can be studied following the approach of this research; the comparison can also include another system called Independent KCS (Chaouiya et al., 2000). The result would be insights into which system is better for use in mulitple tier assembly / manufacturing systems.

References

Aghajani, M., Keramati, A., & Javadi, B. (2012). Determination of number of kanban in a cellular manufacturing system with considering rework process. *The International Journal of Advanced Manufacturing Technology, 63*(9-12), 1177-1189. doi: 10.1007/s00170-012-3973-y

Al-Hawari, T., & Aqlan, F. (2012). A software application for E-Kanban-based WIP control in the aluminium industry. *International Journal of Modelling in Operations Management, 2*(2), 119-137.

Al-Tahat, M. D., Dalalah, D., & Barghash, M. A. (2012). Dynamic programming model for multi-stage single-product Kanban-controlled serial production line. *Journal of Intelligent Manufacturing, 23*(1), 37-48. doi: 10.1007/s10845-009-0336-0

Al-Tahat, M. D., & Mukattash, A. M. (2006). Design and analysis of production control scheme for Kanban-based JIT environment. *Journal of the Franklin Institute, 343*(4-5), 521-531.

Altiok, T. (1997). *Performance Analysis of Manufacturing Systems*: Springer New York.

Ang, A. (2014). *Why Shortage Cost is more suitable for computing "Pull" Systems while Backorder Cost is preferred in "Push"*. Paper presented at the Industrial Engineering and Operations Management (IEOM), Bali, Indonesia. http://iieom.org/ieom/ieom-2014/

Ang, A., & Piplani, R. (2010). *A Model for Determining the Optimal Number of Base Stock and Kanbans in a Single Stage Extended Kanban Control System (EKCS)*. Paper presented at the Proceedings of the 5th AOTULE International Postgraduate Students Conference on Engineering. Conference Proceeding retrieved from

Askin, R. G., Mitwasi, M. G., & Goldberg, J. B. (1993). DETERMINING THE NUMBER OF KANBANS IN MULTI-ITEM JUST-IN-TIME SYSTEMS. *IIE Transactions, 25*(1), 89 - 98.

Bard, J. F., & Golany, B. (1991). Determining the number of kanbans in a multiproduct, multistage production system. *International Journal of Production Research, 29*(5), 881.

Baynat, B., Buzacott, J. A., & Dallery, Y. (2002). Multiproduct Kanban-like Control Systems. *International Journal of Production Research, 40*(16), 4225-4255.

Baynat, B., Dallery, Y., Mascolo, M. D., & Frein, Y. (2001). A multi-class approximation technique for the analysis of kanban-like control systems. *International Journal of Production Research, 39*(2), 307-328.

Berkley, B. J. (1992). A Review of the Kanban Production Control Research Literature. *Production and Operations Management, 1*(4), 393-412.

Black, J. (2007). Design rules for implementing the Toyota Production System. *International Journal of Production Research, 45*(16), 3639-36641.

Bonvik, A. M., Couch, C. E., & Gershwin, S. B. (1997). A comparison of production-line control mechanisms. *International Journal of Production Research, 35*(3), 789-804.

Boonlertvanich, K. (2005). *Extended-CONWIP-Kanban System: Control and Performance Analysis.* (Ph.D.), Georgia Tech, USA.

Buzacott, J. A. (1989). Queuing Models of Kanban and MRP Controlled Production Systems. *Engineering costs and production economics, 17*(1-4), 3-20.

Chao-Hsien, C., & Wei-Ling, S. (1992). Simulation studies in JIT production. *International Journal of Production Research, 30*(11), 2573.

Chaouiya, C., Liberopoulos, G., & Dallery, Y. (2000). The extended kanban control system for production coordination of assembly manufacturing systems. *IIE Transactions (Institute of Industrial Engineers), 32*(10), 999-1012.

Clark, A. J., & Scarf, H. (1960). Optimal Policies for a Multi-Echelon Inventory Problem. *Management Science, 50*(12_supplement), 1782-1790. doi: 10.1287/mnsc.1040.0265

Claudio, D., & Krishnamurthy, A. (2008). Kanban-based pull systems with advance demand information. *International Journal of Production Research, ang*(1), 1 - 22.

Dallery, Y. (2000). Extended kanban control system: Combining kanban and base stock. *IIE Transactions, 32*(4), 369-386.

Deokar, S. S. (2004). Performance evaluation of multi-product Kanban-like control systems. *Thesis (M.S.I.E.)--University of South Florida, 2004.*, 71.

Diem, W., & Kimberley, W. (2001). the same only different. *Automotive Engineer, 26*(5), 46.

DiMascolo, M., Frein, Y., & Dallery, Y. (1996). An analytical method for performance evaluation of kanban controlled production systems. *Operations Research, 44*(1), 50-64.

El-Taha, M. (2009). Departure process in a mixed fork-join synchronization network. *Computers and Mathematics with Applications*(8), 1272. doi: 10.1016/j.camwa.2009.01.031

Fullerton, R. R., & McWatters, C. S. (2001). The production performance benefits from JIT implementation. *Journal of Operations Management, 19*(1), 81-96.

Glover, M. (1999). The niche vehicle takes centre stage. *Automotive Engineer, 24*(4), 22.

Golhar, D. Y., & Stamm, C. L. (1991). The just-in-time philosophy: A literature review. *International Journal of Production Research, 29*(4), 657.

Groenevelt, H., & Karmarkar, U. S. (1988). Dynamic Kanban System Case Study. *Prod Invent Manage J, 29*(2), 46-51.

Gupta, S. M., Al-Turki, Y. A. Y., & Perry, R. F. (1999). Flexible Kanban System. *International Journal of Operations & Production Management, 19*(10), 1065-1093. doi: 10.1108/01443579910271700

Gupta, Y. P., & Gupta, M. C. (1989). System dynamics model for a multi-stage multi-line dual-card JIT-kanban system. *International Journal of Production Research, 27*(2), 309-352.

Hao, Q., & Shen, W. (2008). Implementing a hybrid simulation model for a Kanban-based material handling system. *Robotics and Computer-Integrated Manufacturing, 24*(5), 635-646.

Hopp, W. J., & Spearman, M. L. (2004). To pull or not to pull: What is the question? *Manufacturing & service operations management, 6*(2), 133.

Hopp, W. J., & Spearman, M. L. (2008). *Factory physics* (3rd ed.). New York, NY: McGraw-Hill/Irwin/Irwin.

Huang, C. C., & Kusiak, A. (1996). Overview of Kanban systems. *International Journal of Computer Integrated Manufacturing, 9*(3), 169-189.

Jing, J., & Heragu, S. S. (2009). Solving Semi-Open Queuing Networks. *Operations Research, 57*(2), 391-401.

Jodlbauer, H., & Huber, A. (2008). Service-level performance of MRP, kanban, CONWIP and DBR due to parameter stability and environmental robustness. *International Journal of Production Research, 46*(8), 2179 - 2195.

Jou Lin, C., Frank Chen, F., & Min Chen, Y. (2013). Knowledge kanban system for virtual research and development. *Robotics and Computer-Integrated Manufacturing, 29*(3), 119-134.

Kao, E. P. C. (1997). *An introduction to stochastic processes*. Belmont, Calif., USA: Duxbury Press.

Karaesmen, F., & Dallery, Y. (2000). A performance comparison of pull type control mechanisms for multi-stage manufacturing. *International Journal of Production Economics, 68*(1), 59-71.

Keller, A. Z. (1993). ?Just-in-Time? Manufacturing Systems: A Literature Review. *Industrial Management & Data Systems, 93*(7), 2-32.

Khuller, B. (2006). A Comparison of Traditional and Extended Information Kanban Control System using Dedicated and Shared Kanbans. *Master's Thesis, Graduate College of Oklahoma State University*.

Köchel, P., & Nieländer, U. (2002). Kanban optimization by simulation and evolution. *Production Planning and Control, 13*(8), 725-734. doi: 10.1080/0953728031000057334

Korugan, A., & Çadırcı, Ö. (2008). A Comparison of Pull Control Policies in Hybrid Production Systems. *POMS 19th Annual Conference, La Jolla, California, U.S.A.*

Krajewski, L. J., King, B. E., Ritzman, L. P., & Wong, D. S. (1987). Kanban, MRP, and Shaping the Manufacturing Environment. *Management Science, 33*(1), 39-57.

Krieg, G. N. (2005). *Kanban-controlled manufacturing systems*. Berlin ; New York: Springer.

Krishnamurthy, A., & Suri, R. (2006). Performance analysis of single stage kanban controlled production systems using parametric decomposition. *Queueing systems, 54*(2), 141-162.

Krishnamurthy, A., Suri, R., & Vernon, M. (2003). Two-Moment Approximations for Throughput and Mean Queue Length of a Fork/Join Station with General Inputs from Finite Populations. In J. G. Shanthikumar, D. Yao & W. H. Zijm (Eds.), *Stochastic Modeling and Optimization of Manufacturing Systems and Supply Chains* (Vol. 63, pp. 87-126): Springer US.

Kumar, C. S. (2007). Literature review of JIT-KANBAN system. *International journal of advanced manufacturing technology, 32*(3-4), 393-408.

Lee-Mortimer, A. (2008). A continuing lean journey: An electronic manufacturer's adopting of Kanban. *Assembly Automation, 28*(2), 103-112.

Liberopoulos, G. (2005). Tradeoffs between base stock levels, numbers of kanbans, and planned supply lead times in production/inventory systems with advance demand information. *International Journal of Production Economics, 96*(2), 213-232.

Lind, D. A., Marchal, W. G., & Wathen, S. A. (2011). *Basic statistics for business & economics Douglas A. Lind ; William G. Marchal ; Samuel A. Wathen*: Boston [u.a.] McGraw-Hill 2011. 7th ed., internat. student ed.

Marek, R. P., Elkins, D. A., & Smith, D. R. (2001). *Understanding the fundamentals of Kanban and CONWIP pull systems using simulation.* Paper presented at the Winter Simulation Conference Proceedings.

Monden, Y. (1993). *Toyota production system : an integrated approach to just-in-time* (2nd ed.). Norcross, Ga.: Industrial Engineering and Management Press.

Monden, Y. (1998). *Toyota production system : an integrated approach to just-in-time* (3rd ed.). Norcross, Georgia: Engineering & Management Press.

Nomura, J., & Takakuwa, S. (2004). *Module-based modeling of flow-type multistage manufacturing systems adopting dual-card Kanban system.* Paper presented at the Proceedings - Winter Simulation Conference, Washington, DC.

Nori, V. S., & Sarker, B. R. (1998). Optimum number of Kanbans between two adjacent stations. *Production Planning & Control, 9*(1), 60-65.

Ohno, T. (1988). *Toyota production system : beyond large-scale production*: Cambridge.

Price, W., Gravel, M., & Nsakanda, A. L. (1994). A review of optimisation models of Kanban-based production systems. *European Journal of Operational Research, 75*(1), 1-12.

Ramakumar, R. (1993). *Engineering reliability : fundamentals and applications* (Prentice Hall international ed.). London: Prentice-Hall International.

Ross, S. M. (2007). *Introduction to probability models* (9th ed.). Amsterdam ; Boston: Academic Press.

Saltelli, A. (2008). Global Sensitivity Analysis: The Primer. *John Wiley & Sons, Ltd.*, 306.

Sardi, M. (2009). Ferrari Waiting List - So, You Want a Ferrari? *Automobile Magazine, February, 2009 issue*(February, 2009 issue).

Sarker, B. R., & Fitzsimmons, J. A. (1989). The performance of push and pull systems: a simulation and comparative study. *International Journal of Production Research, 27*(10), 1715.

Sarker, B. R., & Harris, R. D. (1988). The effect of imbalance in a just-in-time production system: A simulation study. *International Journal of Production Research, 26*(1).

Selvaraj, N. (2009). Determining the Number of Kanbans in EKCS: A Simulation Modeling Approach. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol II, IMECS 2009, March 18 - 20, 2009, Hong Kong*.

Singh, N., Kwok Hung, S., & Meloche, D. (1990). The Development of a Kanban System: A Case Study. *International Journal of Operations & Production Management, 10*(7), 28-36.

Spearman, M. L. (1990). Push and pull production systems. Issues and comparisons. *Operations Research, 40*(3), 521-532.

Spearman, M. L. (1992). Customer Service in Pull Production Systems. *Operations Research, 40*(5), 948-958.

Spearman, M. L., Woodruff, D. L., & Hopp, W. J. (1990). CONWIP. A pull alternative to kanban. *International Journal of Production Research, 28*(5), 879.

Sugimori, Y., Kusunoki, K., Cho, F., & Uchikawa, S. (1977). Toyota Production System and Kanban System: Materialization of Just-In-Time and Respect-For-Human System. *International Journal of Production Research, 15*(6), 553.

Takahashi, K., Morikawa, K., & Chen, Y. C. (2007). Comparing kanban control with the theory of constraints using Markov chains. *International Journal of Production Research, 45*(16), 3599-3617.

Takahashi, K., Morikawa, K., & Nakamura, N. (2004). Reactive JIT ordering system for changes in the mean and variance of demand. *International Journal of Production Economics, 92*(2), 181-196.

Takahashi, K., Nakamura, N., & Ohashi, K. (1996). Order Release in JIT Production Systems: A Simulation Study. *SIMULATION, 66*(2), 75-87. doi: 10.1177/003754979606600203

Tardif, V., & Maaseidvaag, L. (2001). An adaptive approach to controlling kanban systems. *European Journal of Operational Research, 132*(2), 411-424.

Times, S. (2012). Hermes to protect exclusive image with fewer new stores. www.straitstimes.com.

Veatch, M. H., & Wein, L. M. (1994). Optimal Control of a Two-Station Tandem Production/Inventory System. *Operations Research, 42*(2), 337-350.

Wang, H., & Hsu-Pin, W. (1991). Optimum number of kanbans between two adjacent workstations in a JIT system. *International Journal of Production Economics, 22*(3), 179-188.

Watanabe, K. (2007). Lessons from Toyota's long drive. *Harvard business review, 85*(7-8), 74.

Weitzman, R., & Rabinowitz, G. (2003). Sensitivity of 'Push' and 'Pull' strategies to information updating rate. *International Journal of Production Research, 41*(9), 2057-2074.

White, R. E., Pearson, J. N., & Wilson, J. R. (1999). JIT Manufacturing: A Survey of Implementations in Small and Large U.S. Manufacturers. *Management Science, 45*(1), 1-15. doi: 10.1287/mnsc.45.1.1

Womack, J. P., Jones Daniel T., & Roos Daniel. (2007). *The machine that changed the world : the story of lean production -- Toyota's secret weapon in the global car wars that is revolutionizing world industry* (1st trade pbk. ed.). New York: Free press.

Yavuz, I. H., & Satir, A. (1995). Kanban-based operational planning and control: simulation and modelling. *Production planning & control, 6*(4), 331.

Zipkin, P. H. (2000). *Foundations of inventory management*. Boston: McGraw-Hill.

APPENDIX A

Derivation of Expected Number of Undispatched Kanbans, E[K], in a Single Stage, Single Product, Extended Kanban Control System (EKCS)

Figure A1 below shows the schematic of a SS/SP/EKCS. The objective here is to derive a closed form expression for the Expected Number of Undispatched Kanbans, E [K_1], in the Undispatched Kanban Queue, K_1.

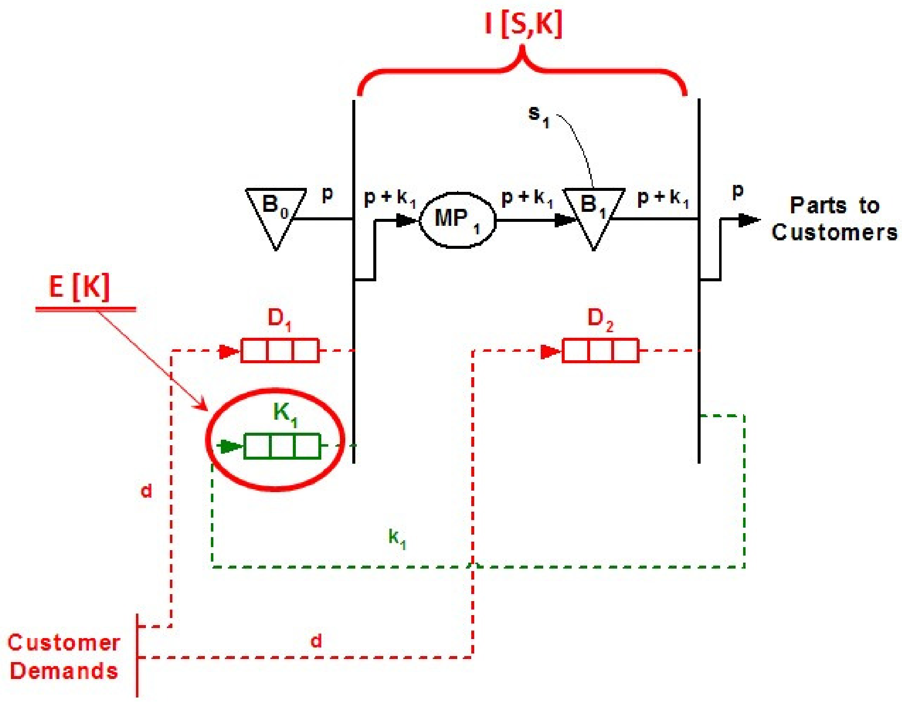

Figure A 1: The SS/SP/EKCS

At any one time, kanbans can only be in three places of the SS/SP/EKCS: Queue, K_1, MP_1 and/or Buffer B_1. Thus,

$$E [K_1] + E [MP_1] + E [B_1] = K + S \quad \text{(A1)}$$

Where

K: Initial number of undispatched kanbans in Kanban Queue K_1

S: Initial Base Stock level in Output Buffer B_1

$E[K_1]$: Expected number of undispatched kanbans in the Kanban Queue K_1

$E[MP_1]$: Expected number of parts with attached kanbans in MP_1

$E[B_1]$: Expected number of parts with attached kanbans in the Output Buffer B_1

Equation (A1) has also been verified by Dallery (2000). However,

$$E[MP_1] + E[B_1] = S - E[D_1] + E[D_2] \tag{A2}$$

Where

$E[D_2]$: Expected number of demands in the Customer Demand Arrival Queue D_2

$E[D_1]$: Expected number of demands in the Customer Demand Arrival Queue D_1

The proof for Equation (A2) can be found in Dallery (2000). However, to give a brief explanation for Equation (A2), it states that the expected number of parts with attached kanbans in MP_1 and Buffer B_1 is equal to the sum of the initial Base Stock Level, S, and the difference between the number of demands in Queues D_2 and D_1. The reason for deducting $E[D_1]$ is because the number of demands in Queue D_1 (which will be matched with the undispatched kanbans from Queue K_1) will be sent into MP_1, however, it will eventually be released as parts to customers since any incoming demand will be replicated to both Queues D_1 and D_2. The reason for adding $E[D_2]$ is because the number of demands in Queue D_2 (which will be matched with the parts from Output Buffer B_1) will detach kanbans and send it upstream upon arriving customer demands. These un-detached kanbans will immediately be sent into MP_1 since the incoming demands are replicated in Queue D_1 as well.

Substituting Equation (A2) into Equation (A1), we obtain

$$E[K_1] = K - E[D_2] + E[D_1] \tag{A3}$$

Proof for Equation (A3) can be found in Dallery (2000). Thus in order to find $E[K_1]$, we need to find $E[D_1]$ and $E[D_2]$ first. In the next section, we shall make use of A. Krishnamurthy and Suri (2006)'s method of parametric decomposition to obtain these parameters.

A1 Obtaining $E[D_1]$ and $E[D_2]$

A1.1 Brief Background of Parametric Decomposition

Kanban Controlled Systems (KCS) represent a special class of queuing network called the Semi-Open Queuing Network (SOQN); which has both the characteristics of an Open Queuing Network (OQN) and a Closed Queuing Network (CQN) (Jing & Heragu, 2009). Some authors have also termed this sort of network a mixed fork/join synchronization network (El-Taha, 2009). Locating exact performance measures in terms of closed form equations for KCS is a complex task because of the dynamic interactions at the synchronization stations (A. Krishnamurthy & Suri, 2006).

These synchronization stations permit arrivals from both external, ad-hoc, customer demands as well as internal downstream kanbans. Due to this mixture of arrivals, researchers can no longer use convenient, standard Continuous Time Markov Chains (CTMC) to model their KCS. This is because the underlying assumption for using CTMC is independent Poisson arrivals (Ross, 2007). But since these synchronization stations cause dependency amongst its arrivals, the outputs from these synchronization stations (and likewise concurrent inputs to upstream stations) are no longer Poisson Processes, hence voiding the independent arrivals assumption.

Since obtaining exact performance measures for sophisticated KCS is so difficult, several authors have used a method called parametric decomposition to breakdown KCS to analyse them in smaller portions (Bruno Baynat, Dallery, Mascolo, & Frein, 2001; DiMascolo, Frein, & Dallery, 1996; A. Krishnamurthy & Suri, 2006).

A1.2 A. Krishnamurthy and Suri (2006)'s Method of Parametric Decomposition

A. Krishnamurthy and Suri (2006) developed an approach that permits efficient performance analysis of kanban systems with general demand processes, material arrival processes, and service times. Although the parametric decomposition technique can only give rise to approximate solutions, A. Krishnamurthy and Suri (2006)'s technique has been reported to be fast and reasonably accurate when compared to simulation.

Their approach has two main features. First, it approximates the traffic processes (arrival and departure processes from the different stations) in the queuing network by renewal processes. A renewal process is simply a counting process that has the inter-arrival time distribution following a general distribution (Kao, 1997). This approximation is necessary since the traffic processes can no longer be assumed Poisson Processes (due to their interdependency as mentioned earlier). Second, it characterizes the distribution of the inter-renewal times of these renewal processes by two parameters, namely the mean and the Squared Coefficient of Variation (SCV).

A. Krishnamurthy and Suri (2006) mentioned that, in reality, the arrival and departure processes in the network are not renewal and successive inter-arrival and inter-departure times are not independent. Therefore the two parameter characterization is only an approximate representation of these traffic processes by "equivalent" renewal processes. The first parameter, corresponding to the mean, is equal to the mean of the corresponding inter-arrival or inter-departure times. The role of the second parameter, corresponding to the SCV of the equivalent renewal process is to account for the complex structure of the arrival and departure processes in an aggregate way.

Their approach consists of four steps: (i) decomposition, (ii) characterization, (iii) linkage and (iv) solution. Figure 2 below gives an overview of their parametric decomposition method. The basic idea here is:

- Step 1: Decompose the system into three stations: the first station being the synchronization station, J_{i-1}, the Manufacturing Process (MP) as the second station, S_i, and the last station being the synchronization station, J_i.
- Step 2: Characterize, or formulate closed form equations for each of this station. Characterization equations comprise of important performance measures such as the length of queues and arrival rates for each station. These expressions were derived earlier on in Ananth Krishnamurthy, Suri, and Vernon (2003).
- Step 3: Link, or perform stochastic transformation to tally the traffic processes between stations. A. Krishnamurthy and Suri (2006) devised two algorithms to link the arrival and departure processes at the individual nodes.
- Step 4: Solve a system of non-linear equations to determine the unknown parameters characterizing the internal traffic processes. Likewise, A. Krishnamurthy and Suri (2006) devised an iterative algorithm for this too.

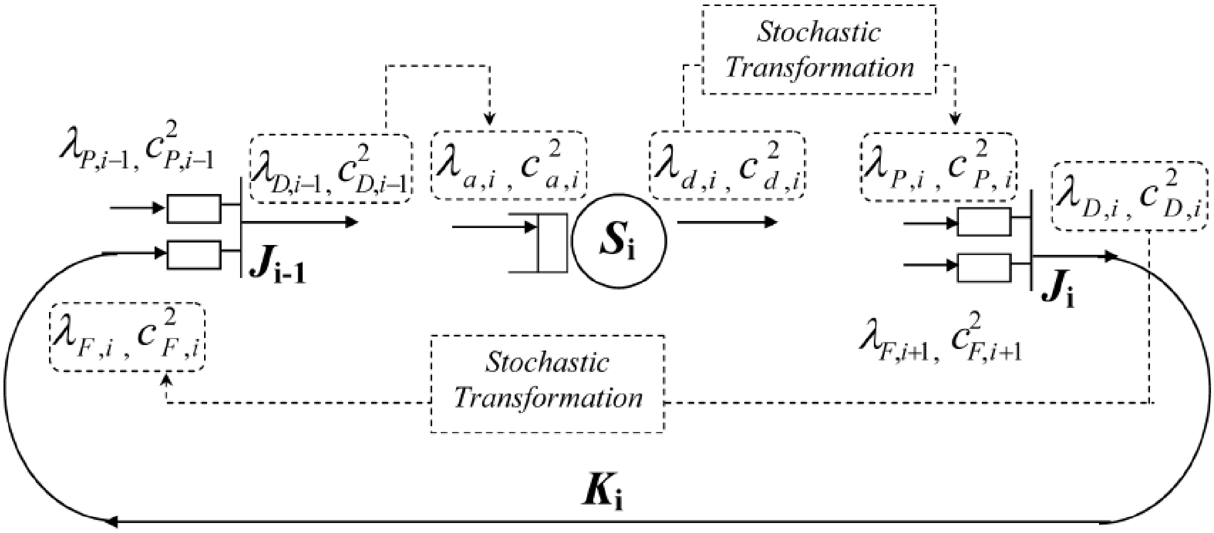

Figure A 2: Overview of parametric decomposition method (A. Krishnamurthy and Suri, 2006)

Finer details of how these four steps function can be found in A. Krishnamurthy and Suri (2006) and thus they will not be repeated here.

A1.3 Applying A. Krishnamurthy and Suri (2006)'s Method of Parametric Decomposition to the SS/SP/EKCS

Referring to Figures A1 and A2, they are very much equivalent. They both have three stations: the first synchronization station, a MP that consist of a single server, and the second synchronization station. And they both have an upper bound WIP; both of which are restricted by the number of kanbans in that stage. There are only two differences, but both of which can be resolved easily.

The first difference between Figures A1 and A2 is located at the first synchronization station, where Figure A1 shows three input buffers, while Figure A2 shows only two. This can be resolved because the assumption for our SS/SP/EKCS is that Input Buffer B_0 comprises of infinite raw components. Thus, since these raw parts are omnipresent, we can disregard its presence. Dallery (2000) has also confirmed this fact that B_0 can be disregarded if it holds infinite raw components.

The second difference is that A. Krishnamurthy and Suri (2006)'s model assumes blockage, while our SS/SP/EKCS does not. That is, all buffers in Figure A2 have a limit to the amount of WIP it can carry, and their preceding stations will come to a halt so long as the buffer in front of it is full. In order to resolve this difference, we shall assume that the long run proportion of time that arrivals to the specific buffer shutting down is null. In other words, in Step 3 of their parametric decomposition method, A. Krishnamurthy and Suri (2006) created two parameters, π_F and π_P, to denote the long run proportion of time the arrival to Queues F and P, respectively, shut down. Since π_F and π_P are long run probabilities (which can only be between a value of 0 and 1), we shall assume them to be zero for all subsequent expressions used.

With these two differences resolved, Figures A1 and A2 are now equivalent.

A1.3.1 Equivalent Representation of the SS/SP/EKCS and Parametric Decomposition Method

Table A1 displays the equivalent representation of the SS/SP/EKCS (Figure A1) and A. Krishnamurthy and Suri (2006)'s parametric decomposition method (Figure A2).

Table A 1: Equivalent Representation of the SS/SP/EKCS and Parametric Decomposition Method

Buffer or Queue at the SS/SP/EKCS (Figure A1)	Equivalent Representation of the Parametric Decomposition method (Figure A2)
Rate of Customer Demand Arrival to Demand Queue D_1	$\lambda_{P,i-1}$
Rate of Kanban Arrival to Undispatched Kanban Queue K_1	$\lambda_{F,i}$
Rate of Departure from First Synchronization Station	$\lambda_{D,i-1}$
Rate of Arrival to the Manufacturing Process (MP)	$\lambda_{a,i}$
Rate of Departure from the MP	$\lambda_{d,i}$
Rate of Arrival to the Output Buffer B_1	$\lambda_{P,i}$
Rate of Customer Demand Arrival to Demand Queue D_2	$\lambda_{F,i+1}$
Rate of Departure from Second	$\lambda_{D,i}$

Synchronization Station	

A1.4 Obtaining E [D₁] of the SS/SP/EKCS

We make use of A. Krishnamurthy and Suri (2006) characterization equation for the expected queue length of Buffer P_{i-1} to obtain

$$E[D_1] = \frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]} \tag{A4}$$

A1.4.1 Proof that $E[D_1] = \frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]}$

A. Krishnamurthy and Suri (2006) gave the characterization equation for the expected queue length for Buffer P_{i-1} (the buffer that has $\lambda_{P,i-1}$ as the arrival rate in Figure A2). This equation is repeated here in Equation (A5)

$$\overline{L}_{P,i-1} = \left[\left(\frac{r_{i-1}^{K_i+1}}{1-r_{i-1}}\right)\left(\frac{1-r_{i-1}^{K_{i-1}}}{1-r_{i-1}^{K_{i-1}+K_i+1}}\right)-\left(\frac{K_{i-1}r_{i-1}^{K_{i-1}+K_i+1}}{1-r_{i-1}^{K_{i-1}+K_i+1}}\right)\right]\times\left[1+\left(\frac{1-r_{i-1}}{1+r_{i-1}}\right)\left(\frac{r_{i-1}^4}{1+r_{i-1}^8}\right)\left(c_{i-1}^2-1\right)\right] \tag{A5}$$

Where

$\overline{L}_{P,i-1}$: Expected Queue Length for Buffer P_{i-1}, i = 1, 2, 3…

r_{i-1}: Defined as $= \frac{\lambda_{P,i-1}}{\lambda_{F,i}}$; which is simply the ratio of the two arrival rates at the first synchronization stations' incoming buffers (Figure A2).

K_i: Total WIP, or total number of kanbans, in Stage i, i = 1, 2, 3…

c_{i-1}^2: Defined as $= 0.5\left(c_{P,i-1}^2 + c_{F,i}^2\right)$; where $c_{P,i-1}^2$ and $c_{F,i}^2$ are the Squared Coefficient of Variation (SCV) of the two arrival rates at the first synchronization stations' incoming buffers respectively (Figure 2).

A1.4.1.1 Average Customer Demand Arrival Rates must be smaller than Average Manufacturing Process Rate

An important assumption for the characterization equations to hold is that the total average customer demand arrival rates must be smaller than the average manufacturing processing rate (A. Krishnamurthy & Suri, 2006; Ananth Krishnamurthy et al., 2003). Since the output rate from synchronization stations only follow the slowest incoming arrival rate (Altiok, 1997), this shows that the overall throughput rate of the SS/SP/EKCS only follow the rate of incoming customer demands. Thus $r_{i-1} < 1$, since $\lambda_{P,i-1} < \lambda_{F,i}$.

A1.4.1.2 K_{i-1} Equates to Infinity

A. Krishnamurthy and Suri (2006) assumed that Buffer P_{i-1} (Figure A2) has limited WIP because it is linked to the previous stage's kanban system. However, for our SS/SP/EKCS, we represent Buffer P_{i-1} as the external Customer Demand Arrival Queue D_1, which is assumed to have an infinite capacity. Thus $K_{i-1} \to \infty$.

A1.4.1.3 Assuming SCV of all arrivals to be exponentially distributed for uniformity

A. Krishnamurthy and Suri (2006) assumed the general distribution for all traffic processes in his parametric decomposition method. However, we will not be able to obtain any substantial result if we left the closed form equations in its general form. Furthermore, the basic assumption for our SS/SP/EKCS was based on the exponential distribution. Hence, for the sake of uniformity and simplicity, we shall assume the exponential distribution for all traffic processes in his parametric decomposition method.

The SCV for an exponential distribution is 1 (Kao, 1997). Therefore, $c_{i-1}^2 = 1$ since $c_{P,i-1}^2$ and $c_{F,i}^2$ are also equal to one.

A1.4.1.4 Limit of $\overline{L}_{P,i-1}$ as $r_{i-1} < 1$, $c_{i-1}^2 = 1$ and K_{i-1} tends towards infinity

Since $c_{i-1}^2 = 1$, $\overline{L}_{P,i-1}$ reduces to

$$\overline{L}_{P,i-1} = \left(\frac{r_{i-1}^{K_i+1}}{1-r_{i-1}}\right)\left(\frac{1-r_{i-1}^{K_{i-1}}}{1-r_{i-1}^{K_{i-1}+K_i+1}}\right) - \left(\frac{K_{i-1}r_{i-1}^{K_{i-1}+K_i+1}}{1-r_{i-1}^{K_{i-1}+K_i+1}}\right)$$

Also, $r_{i-1}^{K_{i-1}} \to 0$ as $K_{i-1} \to \infty$, since $r_{i-1} < 1$.

Therefore, $\lim\limits_{K_{i-1}\to\infty} \overline{L}_{P,i-1} = \dfrac{r_{i-1}^{K_i+1}}{1-r_{i-1}}$

Now, $r_{i-1} = \dfrac{\lambda_{P,i-1}}{\lambda_{F,i}}$ as stated earlier. Thus, we would need to obtain $\lambda_{P,i-1}$ and $\lambda_{F,i}$ before we can get an expression for $\overline{L}_{P,i-1}$

A1.4.1.5 Obtaining $\lambda_{P,i-1}$ and $\lambda_{F,i}$

Since $\lambda_{P,i-1}$ represents the external Customer Demand Arrival Rate into Queue D_1 (Table A1), it can simply be denoted as λ (which follows a Poisson Process according to the assumption for the SS/SP/EKCS).

In order to find $\lambda_{F,i}$, we need to make further use of A. Krishnamurthy and Suri (2006) equations.

From A. Krishnamurthy and Suri (2006), it is stated that $\lambda_{F,i} = \dfrac{\lambda_{D,i}}{1-\pi_{F,i}}$, where $\pi_{F,i}$ denotes the long run proportion of time the arrival to Queue F shuts down. However, as stated in Section A1.3, $\pi_{F,i} = 0$ because of the no blockage assumption in our SS/SP/EKCS. Thus $\lambda_{F,i} = \lambda_{D,i}$. Also, A. Krishnamurthy and Suri (2006) stated that $\lambda_{D,i} = \lambda_{d,i}$ and $\lambda_{d,i} = \lambda_{a,i}$. Thus $\lambda_{F,i} = \lambda_{a,i}$.

A1.4.1.6 Obtaining $\lambda_{a,i}$

Since $\ddot{W}_{q_{S,i}} = C_i \overline{W}_{q_{S,i}}^{GI/G/1}$ (A. Krishnamurthy & Suri, 2006)

Where

$\ddot{W}_{q_{S,i}}$: Mean waiting time in queue at the station in Figure 2.

C_i: Correction factor, which is $=\left(\frac{K_i-1}{K_i}\right)\left(\frac{1}{1+K_i^{-1}\mu_{S,i}\overline{W}_{q_{S,i}}^{GI/G/1}}\right)$

$\overline{W}_{q_{S,i}}^{GI/G/1}$: Mean waiting time in a GI/G/1 queue, which is $=\left(\frac{c_{a,i}^2+c_{S,i}^2}{2}\right)\left(\frac{\rho_{S,i}\mu_{S,i}^{-1}}{1-\rho_{S,i}}\right)$.

$c_{a,i}^2$: SCV of the manufacturing server queue (Figure A2).

$c_{S,i}^2$: SCV of the manufacturing server (Figure A2).

$\rho_{S,i}$: Defined as $=\frac{\lambda_{a,i}}{\mu_{S,i}}$.

$\mu_{S,i}$: Manufacturing process rate of server (which we can simply denote it as μ).

Since $c_{a,i}^2$ and $c_{S,i}^2$ are assumed to be one, after rearranging the equation, we obtain

$\lambda_{a,i}=\lambda_{F,i}=\frac{\mu_{S,i}\left(K_i+1\right)}{K_i}$. Therefore, $r_{i-1}=\left(\frac{\lambda}{\mu}\right)\left[\frac{K_i}{K_i+1}\right]$ and hence

$$\overline{L}_{P,i-1}=\frac{\left(\left(\frac{\lambda}{\mu}\right)\left[\frac{K_i}{K_i+1}\right]\right)^{K_i+1}}{1-\left(\frac{\lambda}{\mu}\right)\left[\frac{K_i}{K_i+1}\right]}.$$

Since $\overline{L}_{P,i}$ is just $E\left[D_1\right]$, and letting $\rho=\frac{\lambda}{\mu}$, and letting $\mu_{S,i}$ simply be denoted it as μ; and $K_i = K+S$,

$$\text{thus } E\left[D_1\right]=\frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]}.$$

A1.5 Obtaining E [D_2] of the SS/SP/EKCS

Likewise, we make use of A. Krishnamurthy and Suri (2006) characterization equation for the expected queue length of Buffer F_{i+1} to obtain

$$E[D_2]=\frac{\rho(K+S)}{K+S+1-\rho(K+S)} \tag{A6}$$

A1.5.1 Proof that $E[D_2]=\frac{\rho(K+S)}{K+S+1-\rho(K+S)}$

From Section A1.4.1.2, we derived $K_{i+1}\to\infty$ (which uses the same reasoning that Buffer F_{i+1} is assumed infinite capacity because it represents the external Customer Demand Arrival Queue D_2 of the SS/SP/EKCS). And from Section A1.4.1.3, we obtained $c_i^2=1$ because we assumed an exponential distribution for all the traffic processes. However, from Section A1.4.1.5, since the total average customer demand arrival rates must be smaller than the average manufacturing processing rate, r_i is now >1, since $r_i=\frac{\lambda_{P,i}}{\lambda_{F,i+1}}$, where $\lambda_{F,i+1}$ is now λ (the Customer Demand Arrival Rate into Queue D_2 (Table A1)). To obtain $\lambda_{P,i}$, A. Krishnamurthy and Suri (2006) stated that $\lambda_{P,i}=\frac{\lambda_{d,i}}{1-\pi_{P,i}}$, where $\pi_{P,i}$ denotes the long run proportion of time the arrival to Queue P shuts down. However, as stated in Section A1.3, $\pi_{P,i}=0$ because of the no blockage assumption in our SS/SP/EKCS. Thus $\lambda_{P,i}=\lambda_{d,i}$, and $\lambda_{d,i}=\lambda_{a,i}=\frac{\mu_{S,i}(K_i+1)}{K_i}$.

A1.5.1.1 Limit of $\overline{L}_{F,i+1}$ as $r_i>1$, $c_i^2=1$ and K_{i+1} tends towards infinity

The characterization equation for Buffer F_{i+1}, given by A. Krishnamurthy and Suri (2006), is as follows

$$\overline{L}_{F,i+1}=\left[\left(\frac{K_{i+1}}{1-r_i^{K_i+K_{i+1}+1}}\right)-\left(\frac{r_i}{1-r_i}\right)\left(\frac{1-r_i^{K_{i+1}}}{1-r_i^{K_i+K_{i+1}+1}}\right)\right]\times\left[1+\left(\frac{1-r_i}{1+r_i}\right)\left(\frac{r_i^4}{1+r_i^8}\right)\left(c_i^2-1\right)\right]$$

(A7)

Since $c_i^2=1$, Equation (5) reduces to

$$\overline{L}_{F,i+1}=\left[\left(\frac{K_{i+1}}{1-r_i^{K_i+K_{i+1}+1}}\right)-\left(\frac{r_i}{1-r_i}\right)\left(\frac{1-r_i^{K_{i+1}}}{1-r_i^{K_i+K_{i+1}+1}}\right)\right]$$

Also, $r_i^{K_{i+1}} \to \infty$ as $K_{i+1} \to \infty$, since $r_i > 1$.

Therefore, $\lim_{K_{i+1} \to \infty} \overline{L}_{F,i+1} = \frac{r_i}{r_i - 1} - 1$

Since $r_i = \frac{\lambda_{P,i}}{\lambda_{F,i+1}}$, where $\lambda_{F,i+1}$ is λ, $\lambda_{P,i} = \frac{\mu_{S,i}(K_i + 1)}{K_i}$, $\mu_{S,i}$ is μ $\Rightarrow r_i = \frac{\mu(K_i + 1)}{\lambda K_i}$.

And since $\overline{L}_{F,i+1}$ is just $E[D_2]$, and letting $\rho = \frac{\lambda}{\mu}$, and K_i = K + S, thus $E[D_2] = \frac{\rho(K+S)}{K+S+1-\rho(K+S)}$.

A2 Obtaining E [K1]

Since $E[D_1] = \frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]}$ and $E[D_2] = \frac{\rho(K+S)}{K+S+1-\rho(K+S)}$, from Equation (1), we obtain

$$E[K_1] \approx K - \frac{\rho(K+S)}{K+S+1-\rho(K+S)} + \frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]} \quad \text{(A8)}$$

A3 Validating E [K1] Against Simulation

Table A2 below shows eighteen randomly simulated scenarios. The "Numerical Model" column shows the values of E [K_1] obtained using Matlab, which was coded from Equation (A8). The Matlab code can be found in Section A4. The "Simulation" column shows values of E [K_1] obtained using Arena. Figure A3 shows a comparison of the values of E [K_1] obtained through the Numerical Model versus Simulation.

Table A 2: Random Scenarios Comparing Numerical Model of E [K_1] to Simulation

Scenario	Demand Arrival Rate (parts per hour)	Manufacturing Process Rate (parts per hour)	Utilization, p	Number of Undispatched Kanbans, K	Base Stock Level, S	Numerical Model	Simulation
1	1	2	0.50	1	1	0.78	0.75
2	1	3	0.33	2	1	1.8	1.85
3	1	4	0.25	3	1	2.83	2.92
4	2	4	0.50	3	7	2.69	2.94
5	2	5	0.40	6	4	5.73	5.88
6	2	6	0.33	7	2	6.77	6.81
7	4	7	0.57	3	3	2.68	2.12
8	4	8	0.50	8	7	7.68	7.62
9	4	9	0.44	5	4	4.71	4.52
10	8	10	0.8	10	7	8.44	6.94
11	8	13	0.62	9	3	5.83	7.69
12	8	15	0.53	15	24	14.66	14.89
13	12	13	0.92	15	15	13.19	6.91
14	12	15	0.80	10	6	6.05	6.98
15	12	18	0.67	13	9	10.33	11.24
16	20	21	0.95	30	10	26.9	17.58
17	20	22	0.91	27	14	25.14	19.17
18	20	25	0.80	19	11	15.06	15.57

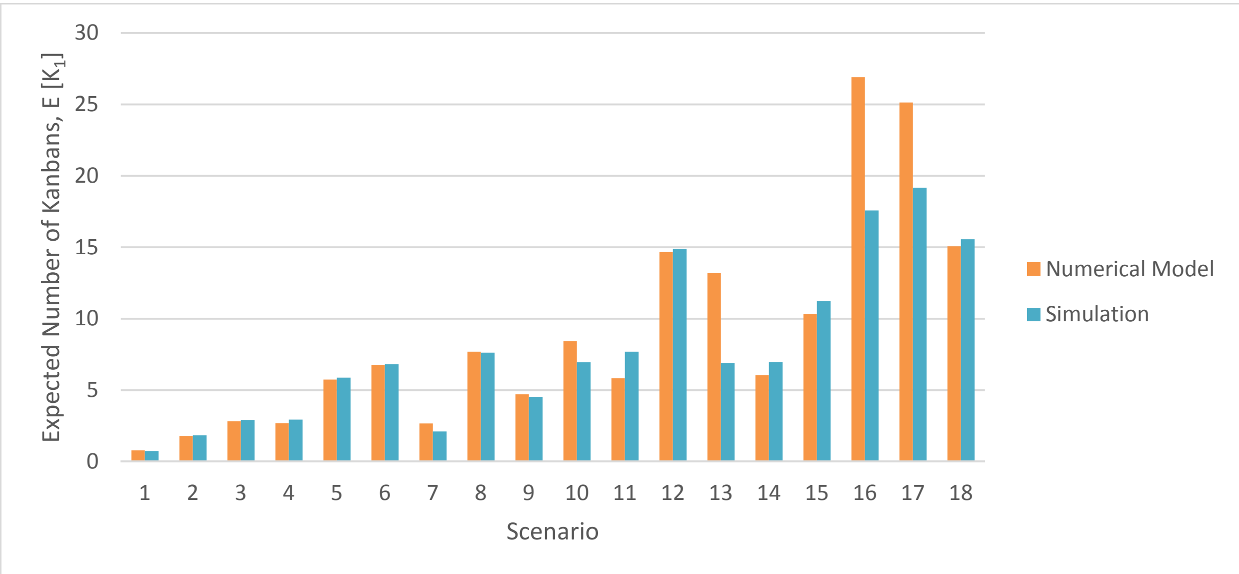

Figure A 3: Comparison of Values from the Numerical Model of E [K_1] versus Simulation

A3.1 Comments on the Differences between the Numerical Model and Simulation

A quick glance at Figure A3 shows Scenarios 13, 16 and 17 have significant difference between the values generated from the Numerical Model versus Simulation. Referring to Table A2, Scenarios 13, 16 and 17 have a common trait – their Utilization, ρ, are above 0.9. Since Utilization is defined as λ/μ, this shows that the Numerical Model is unable to account for cases which have a Customer Demand Arrival rate of over 90%, when compared to its Processing rate. In such cases, the values generated from the Numerical Model appears to have passed over Simulation by a range of six to nine kanbans.

However, as for the other scenarios, the values generated from the Numerical Model seems to mimic the Simulation results greatly. They all have Utilization below 0.9.

In conclusion, the E $[K_1]$ solution formulated in this report (Equation (A8)) is able to approximate the Undispatched Kanbans for Queue K_1 accurately for Utilization below 90%. However, for cases above 90%, more in depth studies are required for greater accuracy.

A4 Matlab Code for Equation A(8)

```
%Declare variables used
syms EK1 ED1 ED2 p K S positive

%Use bank format i.e. 2 dec places
format bank

%Request input from user
lambda = input ( '\n\n Enter demand arrival rate (lambda): ' );
miu = input ( '\n\n Enter MP processing rate (miu): ' );
K = input ( '\n\n Enter number of undispatched kanbans: ' );
S = input ( '\n\n Enter base stock level: ' );

if (lambda > miu)
    fprintf ( 'The demand arrival rate cannot exceed the MP
processing rate! \nPlease try again...');
    break;
end

if (lambda <=0) || (miu <=0)
    fprintf ( 'The demand arrival rate or MP processing rate you
entered is invalid! \nPlease try again...');
    break;
end
```

```
p = lambda/miu;

ED1 = ((p*(K+S)/(K+S+1))^(K+S+1)) / (1-p*(K+S)/(K+S+1));

ED2 = (p*(K+S)) / (K+S+1-p*(K+S));

EK1 = K - ED2 + ED1;

fprintf ('\n E[K1] = ')
disp(EK1);
```

APPENDIX B

Derivation of Expected Time Value of Stock Out, E{$t_{stockout}$} for Single Stage, Single Product Extended Kanban Control System (EKCS)

To derive $E\{t_{stockout}\} = \pi_0 = \left(\frac{\rho - 1}{\rho^{S+K+1} - 1}\right)$, the same Markov Chain method is used as in Nori and Sarker (1998). However, this time the system state is modelled as the output buffer, B_1, of the Single Stage, Single Product Extended Kanban Control System (EKCS), as seen in Figure 3.3. In Nori and Sarker (1998), Markov Simultaneous Equations were derived; next the Expected Time Value of Stock Out, E {$t_{stockout}$} is obtained.

B1 Expected Time Value of Stock Out, E{$t_{stockout}$} by induction

Expected Time Value of Stock Out, E {$t_{stockout}$} = π_0, where π_0, in Markov Chain is defined as long run average probability of time spent in state 0.

B1.1 Case Output Buffer, B=1

Similar to Nori and Sarker (1998), if B=1:

$$\pi_0 = \frac{1}{1+\rho} \tag{B1.1}$$

$$\text{And } \pi_1 = \frac{\rho}{1+\rho} \tag{B1.2}$$

Where $\rho = \frac{\lambda}{\mu}$.

B1.2 Case Output Buffer, B=2

Similar to Appendix A, if B=2

$$\pi_0 = \frac{1}{\left(1+\rho+\rho^2\right)} \tag{B1.3}$$

$$\pi_1 = \frac{\rho}{\left(1+\rho+\rho^2\right)} \tag{B1.4}$$

And $$\pi_2 = \frac{\rho^2}{\left(1+\rho+\rho^2\right)} \tag{B1.5}$$

B1.3 Case B=3

Similar to Appendix A, if B=3:

$$\pi_0 = \frac{1}{\left(1+\rho+\rho^2+\rho^3\right)} \tag{B1.6}$$

$$\pi_1 = \frac{\rho}{\left(1+\rho+\rho^2+\rho^3\right)} \tag{B1.7}$$

$$\pi_2 = \frac{\rho^2}{\left(1+\rho+\rho^2+\rho^3\right)} \tag{B1.8}$$

$$\pi_3 = \frac{\rho^3}{\left(1+\rho+\rho^2+\rho^3\right)} \tag{B1.9}$$

B1.4 Case for B=S+K

As it turns out, the maximum number of parts output buffer B_1, can store is S+K.

Hence $\pi_0 = \frac{1}{\left(1+\rho+\rho^2+\rho^3+\ldots+\rho^{S+K}\right)}$

(B1.10)

Since the sum of a geometric series is $\sum_{i=0}^{S+K} \rho^i = 1+\rho+\rho^2+\rho^3+\ldots+\rho^{S+K}$ (B1.11)

Multiplying both sides by $1-\rho$

$$\begin{aligned}(1-\rho)\sum_{i=0}^{S+K} \rho^i &= (1-\rho)\left(1+\rho+\rho^2+\rho^3+\ldots+\rho^{S+K}\right) \\ &= 1+\rho+\rho^2+\rho^3+\ldots+\rho^{S+K} \\ &\quad -\rho-\rho^2-\rho^3-\ldots-\rho^{S+K+1} \\ &= 1-\rho^{S+K+1}\end{aligned} \tag{B1.12}$$

Hence

$$\sum_{i=0}^{S+K} \rho^i = \frac{1-\rho^{S+K+1}}{1-\rho} \tag{B1.13}$$

And therefore $$E\left\{t_{stockout}\right\} = \pi_0 = \frac{\rho-1}{\rho^{S+K+1}-1} \text{ (shown)} \tag{B1.14}$$

APPENDIX C

Tables of Results for Simulation Comparison of SS/SP/KCS

C1 Scenario 1: Low Backorder and Shortage Cost

- ✓ Holding Cost, C_h = \$10 per unit per day
- ✓ Backorder Cost, C_b = \$20 per unit ($C_b = 2 \times C_h$)
- ✓ Shortage Cost, C_s = \$20 per day ($C_s = 2 \times C_h$)
- ✓ Manufacturing Process (MP) Rate = 20 units per day; assuming exponential processing times
- ✓ Demand Arrival Rates (Follows Poisson Process) = 10, 12, 14, 16, 18 units per day

Table C 1: Simulation Results of SS/SP/EKCS for Scenario 1

Extended Kanban Control System (EKCS)							
Demand Arrival Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
10	0	1	3711	3711	0	0	2.48
12	1	1	4435	4435	0	1	1.88
14	3	1	5172	5172	0	2	1.45
16	4	1	5821	5821	0	2	2.29
18	6	1	6541	6541	0	2	7.65

Table C 2: Costs for SS/SP/EKCS for scenario 1

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	0.00	2.07	2.07
0.00	10.00	1.57	11.57
0.00	20.00	1.21	21.21
0.00	20.00	1.91	21.91
0.00	20.00	6.38	26.38

Table C 3: Simulation Results of SS/SP/TKCS for scenario 1

Traditional Kanban Control System (TKCS)						
Demand Arrival Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
10	1	3711	3711	0	1	1.28
12	1	4435	4435	0	1	1.88
14	3	5172	5172	0	2	1.45
16	4	5821	5821	0	2	2.29
18	5	6541	6541	0	2	8.39

Table C 4: Costs for SS/SP/TKCS for scenario 1

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	10.00	1.07	11.07
0.00	10.00	1.57	11.57
0.00	20.00	1.21	21.21
0.00	20.00	1.91	21.91
0.00	20.00	6.99	26.99

Table C 5: Simulation Results of SS/SP/BS for scenario 1

Base Stock (BS)						
Demand Arrival Rate	Optimal Base Stock Level, S*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
10	11	3711	3711	0	10	0
12	13	4435	4435	0	12	0
14	16	5172	5172	0	14	0
16	18	5821	5821	0	15	0.05
18	20	6541	6541	0	12	2.24

Table C 6: Costs for SS/SP/BS for scenario 1

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	100.00	0.00	100.00
0.00	120.00	0.00	120.00
0.00	140.00	0.00	140.00
0.00	150.00	0.04	150.04
0.00	120.00	1.87	121.87

C2 Scenario 2: Medium Backorder and Shortage Cost

- ✓ Holding Cost, C_h = \$10 per unit per day
- ✓ Backorder Cost, C_b = \$200 per unit (C_b = 20 x C_h)
- ✓ Shortage Cost, C_s = \$200 per day (C_s = 20 x C_h)
- ✓ Manufacturing Process (MP) Rate = 20 units per day; assuming exponentially distributed processing times
- ✓ Demand Arrival Rates = 10, 12, 14, 16, 18 units per day; assuming Poisson process

Table C 7: Simulation Results of SS/SP/EKCS for scenario 2

Extended Kanban Control System (EKCS)							
Demand Arrival Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
10	0	4	3711	3711	0	0	2.48
12	1	4	4435	4435	0	1	1.88
14	5	4	5172	5172	0	3	0.66
16	4	5	5821	5821	0	2	2.29
18	8	9	6541	6541	0	3	6.38

Table C 8: Costs for SS/SP/EKCS for scenario 2

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	0.00	20.67	20.67
0.00	10.00	15.67	25.67
0.00	30.00	5.50	35.50
0.00	20.00	19.08	39.08
0.00	30.00	53.17	83.17

Table C 9: Simulation Results of SS/SP/TKCS for scenario 2

Traditional Kanban Control System (TKCS)						
Demand Arrival Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
10	1	3711	3711	0	1	1.28
12	3	4435	4435	0	2	0.68
14	4	5172	5172	0	3	0.99
16	4	5821	5821	0	2	2.29
18	5	6541	6541	0	2	8.39

Table C 10: Costs for SS/SP/TKCS for scenario 2

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	10.00	10.67	20.67
0.00	20.00	5.67	25.67
0.00	30.00	8.25	38.25
0.00	20.00	19.08	39.08
0.00	20.00	69.92	89.92

Table C 11: Simulation results of SS/SP/BS for scenario 2

Base Stock (BS)						
Demand Arrival Rate	Optimal Base Stock Level, S*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
10	15	3711	3711	0	14	0
12	18	4435	4435	0	17	0
14	20	5172	5172	0	18	0
16	23	5821	5821	0	20	0
18	25	6541	6541	0	17	1.51

Table C 12: Costs for SS/SP/BS for scenario 2

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	140.00	0.00	140.00
0.00	170.00	0.00	170.00
0.00	180.00	0.00	180.00
0.00	200.00	0.00	200.00
0.00	170.00	12.58	182.58

C3 Scenario 3: High Backorder and Shortage Cost

- ✓ Holding Cost, C_h = \$10 per unit per day
- ✓ Backorder Cost, C_b = \$2000 per unit (C_b = 200 x C_h)
- ✓ Shortage Cost, C_s = \$2000 per day (C_s = 200 x C_h)
- ✓ Manufacturing Process (MP) = 20 units per day; assuming exponential distributed processing times
- ✓ Demand Arrival Rates = 10, 12, 14, 16, 18 units per day; assuming Poisson process

Table C 13: Simulation Results of SS/SP/EKCS scenario 3

Extended Kanban Control System (EKCS)							
Demand Arrival Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
10	5	7	3711	3711	0	4	0.068
12	5	9	4435	4435	0	4	0.22
14	8	11	5172	5172	0	5	0.27
16	11	13	5821	5821	0	8	0.43
18	29	16	6541	6541	0	20	1.05

Table C 13: Costs for SS/SP/EKCS for scenario 3

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	40.00	5.67	45.67
0.00	40.00	18.33	58.33
0.00	50.00	22.50	72.50
0.00	80.00	35.83	115.83
0.00	200.00	87.50	287.50

Table C 14: Simulation Results of SS/SP/TKCS for scenario 3

Traditional Kanban Control System (TKCS)						
Demand Arrival Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
10	6	3711	3711	0	5	0.02
12	7	4435	4435	0	6	0.05
14	9	5172	5172	0	7	0.08
16	11	5821	5821	0	8	0.43
18	27	6541	6541	0	18	1.27

Table C 15: Costs for SS/SP/TKCS for scenario 3

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	50.00	1.67	51.67
0.00	60.00	4.17	64.17
0.00	70.00	6.67	76.67
0.00	80.00	35.83	115.83
0.00	180.00	105.83	285.83

Table C 16: Simulation Results of SS/SP/BS for scenario 3

Base Stock (BS)						
Demand Arrival Rate	Optimal Base Stock Level, S*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
10	18	3711	3711	0	17	0
12	21	4435	4435	0	20	0
14	24	5172	5172	0	22	0
16	26	5821	5821	0	23	0
18	29	6541	6541	0	20	1.05

Table C 17: Costs for SS/SP/BS for scenario 3

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	170.00	0.00	170.00
0.00	200.00	0.00	200.00
0.00	220.00	0.00	220.00
0.00	230.00	0.00	230.00
0.00	200.00	87.50	287.50

C4 Scenario 4: Comparing only EKCS and TKCS for Low Utilization Rate (Low Backorder and Shortage Cost)

- ✓ Holding Cost, C_h = \$10 per unit per day
- ✓ Backorder Cost, C_b = \$20 per unit ($C_b = 2 \times C_h$)
- ✓ Shortage Cost, C_s = \$20 per day ($C_s = 2 \times C_h$)
- ✓ Manufacturing Process (MP) = 20 units per day; assuming exponentially distributed processing times
- ✓ Demand Arrival Rates = 2, 4, 6, 8 units per day; assuming Poisson process

Table C 18: Simulation Results of SS/SP/EKCS scenario 4

Extended Kanban Control System (EKCS)							
Demand Arrival Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
2	0	1	725	725	0	0	1.35
4	0	1	1491	1491	0	0	1.49
6	0	1	2198	2198	0	0	1.7
8	0	1	2950	2950	0	0	2.01

Table C 19: Costs for SS/SP/EKCS for scenario 4

Total Backorder Cost (\$)	Total Holding Cost (\$)	Total Shortage Cost (\$)	Actual Total Cost (ATC) (\$)
0.00	0.00	1.13	1.13
0.00	0.00	1.24	1.24
0.00	0.00	1.42	1.42
0.00	0.00	1.68	1.68

Table C 20: Simulation Results of SS/SP/TKCS for scenario 4

Traditional Kanban Control System (TKCS)						
Demand Arrival Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	B1 (Units)	(hours)
2	1	725	725	0	1	0.19
4	1	1491	1491	0	1	0.32
6	1	2198	2198	0	1	0.52
8	1	2950	2950	0	1	0.82

Table C 21: Costs for SS/SP/TKCS for scenario 4

Total Backorder Cost ($)	Total Holding Cost($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	10.00	0.16	10.16
0.00	10.00	0.27	10.27
0.00	10.00	0.43	10.43
0.00	10.00	0.68	10.68

C5 Scenario 5: Comparing only EKCS and TKCS for Low Utilization Rate (Medium Backorder and Shortage Cost)

- ✓ Holding Cost, C_h = \$10 per unit per day
- ✓ Backorder Cost, C_b = \$200 per unit (C_b = 20 x C_h)
- ✓ Shortage Cost, C_s = \$200 per day (C_s = 20 x C_h)
- ✓ Manufacturing Process (MP) = 20 units per day; assuming exponentially distributed processing times
- ✓ Demand Arrival Rates = 2, 4, 6, 8 units per day; assuming Poisson process

Table C 22: Simulation Results of (SS – SP – EKCS) for scenario 5

Extended Kanban Control System (EKCS)							
Demand Arrival Rate	Optimal Base Stock Level, S*	Optimal y* (Kanbans)	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(hours)
2	0	1	725	725	0	0	1.35
4	0	2	1491	1491	0	0	1.49
6	0	3	2198	2198	0	0	1.7
8	0	4	2950	2950	0	0	2.01

Table C 23: Costs for SS/SP/EKCS for scenario 5

Total Backorder Cost (\$)	Total Holding Cost (\$)	Total Shortage Cost (\$)	Actual Total Cost (ATC) (\$)
0.00	0.00	11.25	11.25
0.00	0.00	12.42	12.42
0.00	0.00	14.17	14.17
0.00	0.00	16.75	16.75

Table C 24: Simulation Results of SS/SP/TKCS for scenario 5

Traditional Kanban Control System (TKCS)						
Demand Arrival Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	B1 (Units)	(hours)
2	1	725	725	0	1	0.19
4	1	1491	1491	0	1	0.32
6	1	2198	2198	0	1	0.52
8	1	2950	2950	0	1	0.82

Table C 25: Costs for SS/SP/TKCS for scenario 5

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0.00	10.00	1.58	11.58
0.00	10.00	2.67	12.67
0.00	10.00	4.33	14.33
0.00	10.00	6.83	16.83

C6 Scenario 6: Comparing only EKCS and TKCS for Low Utilization Rate (High Backorder and Shortage Cost)

- ✓ Holding Cost, C_h = \$10 per unit per day
- ✓ Backorder Cost, C_b = \$2000 per unit ($C_b = 200 \times C_h$)
- ✓ Shortage Cost, C_s = \$2000 per day ($C_s = 200 \times C_h$)
- ✓ Manufacturing Process (MP) = 20 units per day; assuming exponentially distributed processing times
- ✓ Demand Arrival Rates = 2, 4, 6, 8 units per day; assuming Poisson process

Table C 26: Simulation Results of SS/SP/EKCS for scenario 6

Extended Kanban Control System (EKCS)							
Demand Arrival Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	B1 (Units)	(Hours)
2	1	3	725	725	0	1	0.19
4	3	4	1491	1491	0	3	0.02
6	4	5	2198	2198	0	4	0.01
8	5	6	2950	2950	0	5	0.02

Table C 27: Costs for SS/ SP/EKCS for scenario 6

Total Backorder Cost (\$)	Total Holding Cost (\$)	Total Shortage Cost (\$)	Actual Total Cost (ATC) (\$)
0.00	10.00	15.83	25.83
0.00	30.00	1.67	31.67
0.00	40.00	0.83	40.83
0.00	50.00	1.67	51.67

Table C 28: Simulation Results of SS/SP/TKCS for scenario 6

Traditional Kanban Control System (TKCS)						
Demand Arrival Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory in Output Buffer, B_1	Average Customer Waiting Time
(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
2	1	725	725	0	1	0.19
4	3	1491	1491	0	3	0.02
6	4	2198	2198	0	4	0.01
8	5	2950	2950	0	5	0.02

Table C 29: Costs for SS/SP/TKCS for scenario 6

Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost ($)
0.00	10.00	15.83	25.83
0.00	30.00	1.67	31.67
0.00	40.00	0.83	40.83
0.00	50.00	1.67	51.67

APPENDIX D

Deriving Expected Total Cost (ETC) for SS/MP/TKCS

Askin et al. (1993) first optimized a KCS – deriving the Expected Total Cost (ETC) using Markov Chains, then finding the lowest ETC by exhaustively searching over number of kanbans, K. The 'K' that gives the lowest ETC is the optimal K. Since ETC comprises of two main components: expected total holding cost and expected shortage cost, this method is computationally intensive as expected total shortage cost is based on the sum of the steady state probabilities of backorders, or

$$\underset{k}{\text{Minimize}}\, ETC = \sum_{j=1}^{m}\left\{ C_{hj}\sum_{x=0}^{k_j} x p_j\left(x\right) + C_{sj}\sum_{x=-\infty}^{-1} p_j\left(x\right)\right\} \tag{D.1}$$

The Expected Total Shortage Cost (ETSC) component of the ETC is

$$\text{ETSC} = \sum_{j=1}^{m}\left[C_{sj}\sum_{x=-\infty}^{-1} p_j\left(x\right)\right] \tag{D.2}$$

This computation of the shortage cost is computationally intensive as it requires the long run probability, π_i (x), of different number of backorders. This is really not necessary since the shortage cost per unit time is more important than the backorder cost per unit in kanban systems (Nori & Sarker, 1998; Wang & Hsu-Pin, 1991). Thus, the sum of all backorder probabilities can simply be replaced by the long run probability of a stock out condition. That is, the ETSC equation can be written as

$$\text{ETSC} = \sum_{j=1}^{m}\left[s_j \pi_j\left(0\right)\right] \tag{D3}$$

Nori and Sarker (1998) and Wang and Hsu-Pin (1991) also used the Markov chains to obtain optimal number of kanbans. Additionally, the original Markov chain used by Askin et al. (1993) has also been shortened, increasing the computational speed. The modified ETC equation for a MP/ TKCS can thus be written as

$$\underset{k}{\text{Minimize}}\, ETC = \sum_{j=1}^{m}\left\{ C_{s_j}\pi_j\left(0\right) + C_{h_j}\sum_{x=0}^{k_j} x_j \pi_j\left(x\right)\right\} \tag{D4}$$

Askin et al. (1993) noted that since each k_i can be solved for separately (as it is dedicated), hereafter the subscript j is omitted. The optimal number of kanbans for each product type is found by optimizing ETC. Hence, referring to just one specific product type, its ETC can be written as:

$$\underset{k}{\text{Minimize}}\, ETC = C_s \pi_0 + C_h \sum_{x=1}^{k} x \pi_x \tag{D5}$$

D1 Optimal Kanbans, K*

To obtain the optimal number of kanbans, K*, firstly, the output buffer of the particular product type is modelled as a Markov chain.

$$\mu_A = \frac{\sum_{j=1}^{m} \lambda_j}{\sum_{j=1}^{m} \lambda_j t_j} \tag{D6}$$

The total number of jobs in process (or total WIP for all product types, j = 1, 2), L, for an M/G/1 model (Askin et al., 1993; Ross, 2007) is

$$L = \frac{D^2 / {\mu_A}^2}{\left(1 - D/\mu_A\right)} + \frac{D}{\mu_A} \tag{D7}$$

Figures D1(a) to (c) show the state space transition diagrams for the output buffer of a SS/MP/TKCS. They reflect the system transitions from state π(x) to π (x-1) at the rate λ (demand from successors). The system changes from state π(x) to π(x+1) whenever a job finishes at MP. As there are L jobs in process, service completions occur at the rate μ_A, and the probability a completion is of type j is equal to the proportion of jobs which are of type j. If x is the actual on-hand inventory, this proportion is (k - x) / L and the processing rate should be (k - x) / L * μ_A.

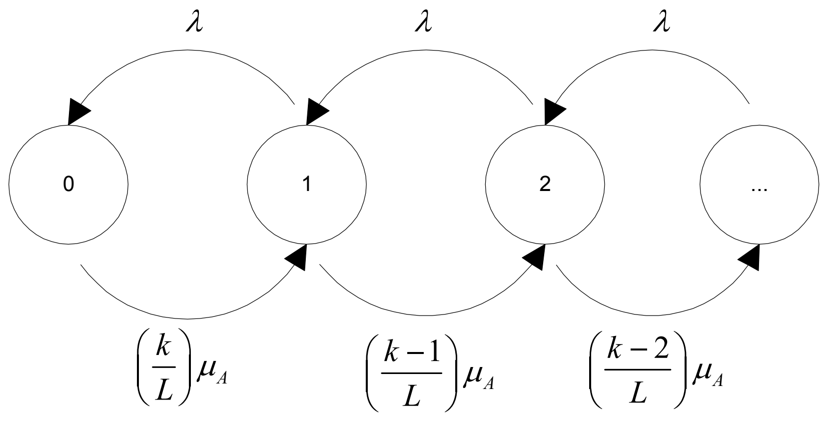

Figure D1(a): Start of Markov chain for MP/TKCS

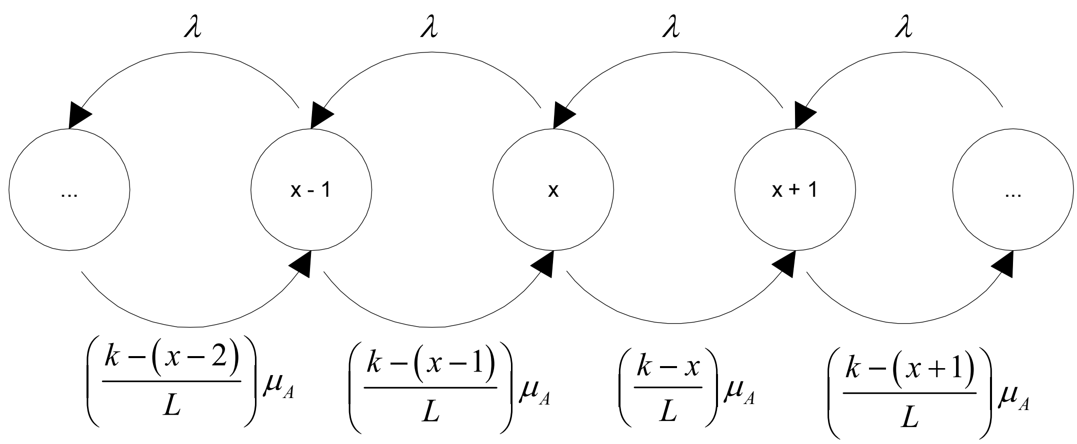

Figure D1(b): Middle potion of Markov chain for MP/TKCS

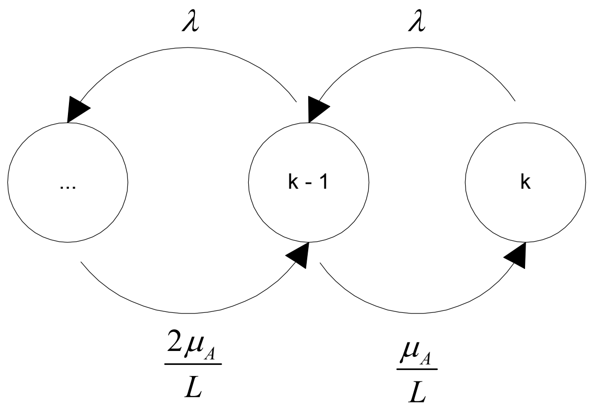

Figure D1(c): End of Markov chain for MP/TKCS

Next, Ramakumar (1993) standard technique in stochastic processes is used to develop a coefficient matrix S. This matrix is used to obtain steady state probabilities for different states. These probabilities are then used in developing an expression for the expected total cost. To begin, the rate of departure matrix, R, for this Markov chain is

$$R = \begin{array}{c|ccccccc} & S_0 & S_1 & S_2 & \cdots & S_{k-2} & S_{k-1} & S_k \\ \hline S_0 & 0 & \frac{k\mu_A}{L} & 0 & \cdots & 0 & 0 & 0 \\ S_1 & \lambda & 0 & \frac{(k-1)\mu_A}{L} & \cdots & 0 & 0 & 0 \\ S_2 & 0 & \lambda & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ S_{k-2} & 0 & 0 & 0 & \cdots & 0 & \frac{2\mu_A}{L} & 0 \\ S_{k-1} & 0 & 0 & 0 & \cdots & \lambda & 0 & \frac{\mu_A}{L} \\ S_k & 0 & 0 & 0 & \cdots & 0 & \lambda & 0 \end{array} \tag{D8}$$

The coefficient matrix, S, developed from the rate of departures matrix R, is

$$
S = \begin{array}{c|ccccccc}
 & S_0 & S_1 & S_2 & \cdots & S_{k-2} & S_{k-1} & S_k \\
\hline
S_0 & -\left(\dfrac{k\mu_A}{L}\right) & \lambda & 0 & \cdots & 0 & 0 & 0 \\
S_1 & \dfrac{k\mu_A}{L} & -\left(\lambda + \dfrac{(k-1)\mu_A}{L}\right) & \lambda & \cdots & 0 & 0 & 0 \\
S_2 & 0 & \dfrac{(k-1)\mu_A}{L} & -\left(\lambda + \dfrac{(k-2)\mu_A}{L}\right) & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
S_{k-2} & 0 & 0 & 0 & \cdots & -\left(\lambda + \dfrac{2\mu_A}{L}\right) & \lambda & 0 \\
S_{k-1} & 0 & 0 & 0 & \cdots & \dfrac{2\mu_A}{L} & -\left(\lambda + \dfrac{\mu_A}{L}\right) & \lambda \\
S_k & 0 & 0 & 0 & \cdots & 0 & \dfrac{\mu_A}{L} & -\lambda
\end{array} \tag{D9}
$$

Let $\pi_0, \pi_1, \ldots, \pi_k$ be the steady state probabilities that the Markov Process is in state 'k'. The set of Markov differential equations for the system, defined by coefficient matrix S, can be written as

$$\pi'(t) = S\,\pi(t) \tag{D10}$$

Where $\pi(t) = \begin{bmatrix} \pi_0(t) \\ \pi_1(t) \\ \vdots \\ \pi_{k-1}(t) \\ \pi_k(t) \end{bmatrix}$, $\pi'(t) = \begin{bmatrix} \pi'_0(t) \\ \pi'_1(t) \\ \vdots \\ \pi'_{k-1}(t) \\ \pi'_k(t) \end{bmatrix}$ and $\pi'_k(t) = \dfrac{d\pi_k(t)}{dt}$

Now using $\sum_{k=1}^{K} \pi_k = 1$ and the S matrix, the expanded form of the Markov differential equation is

$$\begin{bmatrix}0\\0\\0\\0\\0\\0\\1\end{bmatrix} = \begin{bmatrix} -\left(\frac{k\mu_A}{L}\right) & \lambda & 0 & \cdots & 0 & 0 & 0 \\ \frac{k\mu_A}{L} & -\left(\lambda+\frac{(k-1)\mu_A}{L}\right) & \lambda & \cdots & 0 & 0 & 0 \\ 0 & \frac{(k-1)\mu_A}{L} & -\left(\lambda+\frac{(k-2)\mu_A}{L}\right) & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\left(\lambda+\frac{2\mu_A}{L}\right) & \lambda & 0 \\ 0 & 0 & 0 & \cdots & \frac{2\mu_A}{L} & -\left(\lambda+\frac{\mu_A}{L}\right) & \lambda \\ 1 & 1 & 1 & \cdots & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix}\pi_0\\ \pi_1\\ \pi_2\\ \vdots\\ \pi_{k-2}\\ \pi_{k-1}\\ \pi_k\end{bmatrix} \tag{D11}$$

If k = 1,

$$R = \begin{array}{c|cc} & S_0 & S_1 \\ \hline S_0 & 0 & \frac{k\mu_A}{L} \\ S_1 & \lambda & 0 \end{array} \tag{D12}$$

and

$$S = \begin{array}{c|cc} & S_0 & S_1 \\ \hline S_0 & -\frac{k\mu_A}{L} & \lambda \\ S_1 & 1 & 1 \end{array} \tag{D13}$$

Hence

$$-\left(\frac{k\mu_A}{L}\right)\pi_0 + \lambda\pi_1 = 0 \tag{D14}$$

And

$$\pi_0 + \pi_1 = 1 \tag{D15}$$

If $\rho = \frac{\mu_A}{L}$, then

$$\pi_0 = \frac{\lambda}{k\rho + \lambda} \tag{D16}$$

$$\pi_1 = \frac{k\rho}{k\rho + \lambda} \tag{D17}$$

Now if k = 2,

$$R = \begin{array}{c|ccc} & S_0 & S_1 & S_2 \\ \hline S_0 & 0 & \frac{k\mu_A}{L} & 0 \\ S_1 & \lambda & 0 & \frac{(k-1)\mu_A}{L} \\ S_2 & 0 & \lambda & 0 \end{array} \tag{D18}$$

And

$$S = \begin{array}{c|ccc} & S_0 & S_1 & S_2 \\ \hline S_0 & -\frac{k\mu_A}{L} & \lambda & 0 \\ S_1 & \frac{k\mu_A}{L} & -\left(\lambda + \frac{(k-1)\mu_A}{L}\right) & \lambda \\ S_2 & 1 & 1 & 1 \end{array} \tag{D19}$$

Hence

$$-\left(\frac{k\mu_A}{L}\right)\pi_0 + \lambda\pi_1 = 0 \tag{D20}$$

$$\left(\frac{k\mu_A}{L}\right)\pi_0 - \left(\lambda + \frac{(k-1)\mu_A}{L}\right)\pi_1 + \lambda\pi_2 = 0 \tag{D21}$$

And

$$\pi_0 + \pi_1 + \pi_2 = 1 \tag{D22}$$

If $\rho = \frac{\mu_A}{L}$, then

$$\pi_0 = \frac{\lambda^2}{\lambda(k\rho + \lambda) + k\rho^2(k-1)} \tag{D23}$$

$$\pi_1 = \frac{k\rho\lambda}{\lambda(k\rho+\lambda)+k\rho^2(k-1)} \tag{D24}$$

$$\pi_2 = \frac{k\rho^2(k-1)}{\lambda(k\rho+\lambda)+k\rho^2(k-1)} \tag{D25}$$

Now letting k = 3,

$$R = \begin{array}{c|cccc} & S_0 & S_1 & S_2 & S_3 \\ \hline S_0 & 0 & \frac{k\mu_A}{L} & 0 & 0 \\ S_1 & \lambda & 0 & \frac{(k-1)\mu_A}{L} & 0 \\ S_2 & 0 & \lambda & 0 & \frac{(k-2)\mu_A}{L} \\ S_3 & 0 & 0 & \lambda & 0 \end{array} \tag{D26}$$

and

$$S = \begin{array}{c|cccc} & S_0 & S_1 & S_2 & S_3 \\ \hline S_0 & -\left(\frac{k\mu_A}{L}\right) & \lambda & 0 & 0 \\ S_1 & \frac{k\mu_A}{L} & -\left(\lambda+\frac{(k-1)\mu_A}{L}\right) & \lambda & 0 \\ S_2 & 0 & \frac{(k-1)\mu_A}{L} & -\left(\lambda+\frac{(k-2)\mu_A}{L}\right) & \lambda \\ S_3 & 1 & 1 & 1 & 1 \end{array} \tag{D27}$$

And after substitution and letting $\rho = \frac{\mu_A}{L}$,

$$\pi_0 = \frac{\lambda^3}{\lambda\left[\lambda(k\rho+\lambda)+k\rho^2(k-1)\right]+k(k-1)(k-2)\rho^3} \tag{D28}$$

$$\pi_1 = \frac{k\rho\lambda^2}{\lambda\left[\lambda(k\rho+\lambda)+k\rho^2(k-1)\right]+k(k-1)(k-2)\rho^3} \tag{D29}$$

$$\pi_2 = \frac{k(k-1)\rho^2\lambda}{\lambda\left[\lambda(k\rho+\lambda)+k\rho^2(k-1)\right]+k(k-1)(k-2)\rho^3} \tag{D30}$$

$$\pi_3 = \frac{k(k-1)(k-2)\rho^3}{\lambda\left[\lambda(k\rho+\lambda)+k\rho^2(k-1)\right]+k(k-1)(k-2)\rho^3} \tag{D31}$$

By induction,, generalizing the numerators gives

$$\pi_0 = \lambda^k \tag{D32}$$

and

$$\pi_x = \frac{k!}{(k-x)!}\rho^x\lambda^{k-x} \tag{D33}$$

As for the denominator, generalization leads to

$$\sum_{x=0}^{k}\frac{k!}{(k-x)!}\lambda^{k-x}\rho^x \tag{D34}$$

Finally, substituting (D32) and (D34) into (D5) gives

$$\underset{k}{\text{Minimize}}\, ETC(k) = \frac{C_s\left(\lambda^k\right)+C_h\left[\sum_{x=1}^{k}x\left(\frac{k!}{(k-x)!}\lambda^{k-x}\rho^x\right)\right]}{\sum_{x=0}^{k}\frac{k!}{(k-x)!}\lambda^{k-x}\rho^x} \tag{D35}$$

APPENDIX E

Expected Total Cost (ETC) for SS/MP/De-EKCS

Equation (E1) shows the expected total cost, ETC (S_i, K_i), for MP/De-EKCS. It is the sum of the total expected inventory holding and shortage costs.

$$ETC\left(S_i, K_i\right) = \text{ Total Inventory Holding Cost + Total Shortage Cost}$$
$$= \sum_{i=1}^{m}\left[C_{h_i}\left(I\left[S_i, K_i\right]\right) + C_{s_i}\left(E\left\{t_{stockout_i}\right\}\right)\right] \qquad \text{(E1)}$$

As the kanbans are dedicated, each k_i can be solved separately (Askin et al., 1993), hereafter the subscript i is omitted. That is, the optimal S* and K* for each product type is found by obtaining the lowest ETC separately. Hence, referring to just one specific product type, its ETC equation is:

$$ETC(S,K) = C_h\left(I[S,K]\right) + C_s\left(E\left\{t_{stockout}\right\}\right) \qquad \text{(E2)}$$

E1 Expected Inventory Level, I [S_i, K_i]

The expected inventory level, I[S,K], of MP/De-EKCS represents the expected number of parts in the MP as well as in the output buffer in the long run, for that particular product type. Figure E1 shows the location of I[S,K] and E[K], the expected number of undispatched kanbans in the kanban queue, K_1[1], in a SS/MP/De-EKCS. In EKCS, kanbans can only be present in 3 places: the undispatched kanban queue, K; in MP or attached to stock in output buffer, B; thus,

$$I\,[S, K] = (S+K) - E\,[K_1] \qquad \text{(E3)}$$

$$E\left[K_1\right] = K - \frac{\rho(K+S)}{K+S+1-\rho(K+S)} + \frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]}$$, is the expected number of undispatched kanbans in the kanban queue for the particular product type (see Appendix A for its derivation).

The method presented so far is similar to Ang and Piplani (2010) – where SP/EKCS was optimized. As the two-product Dedicated EKCS has properties similar to the single product; it is possible to segregate the two products and optimize them individually; hence the name "dedicated".

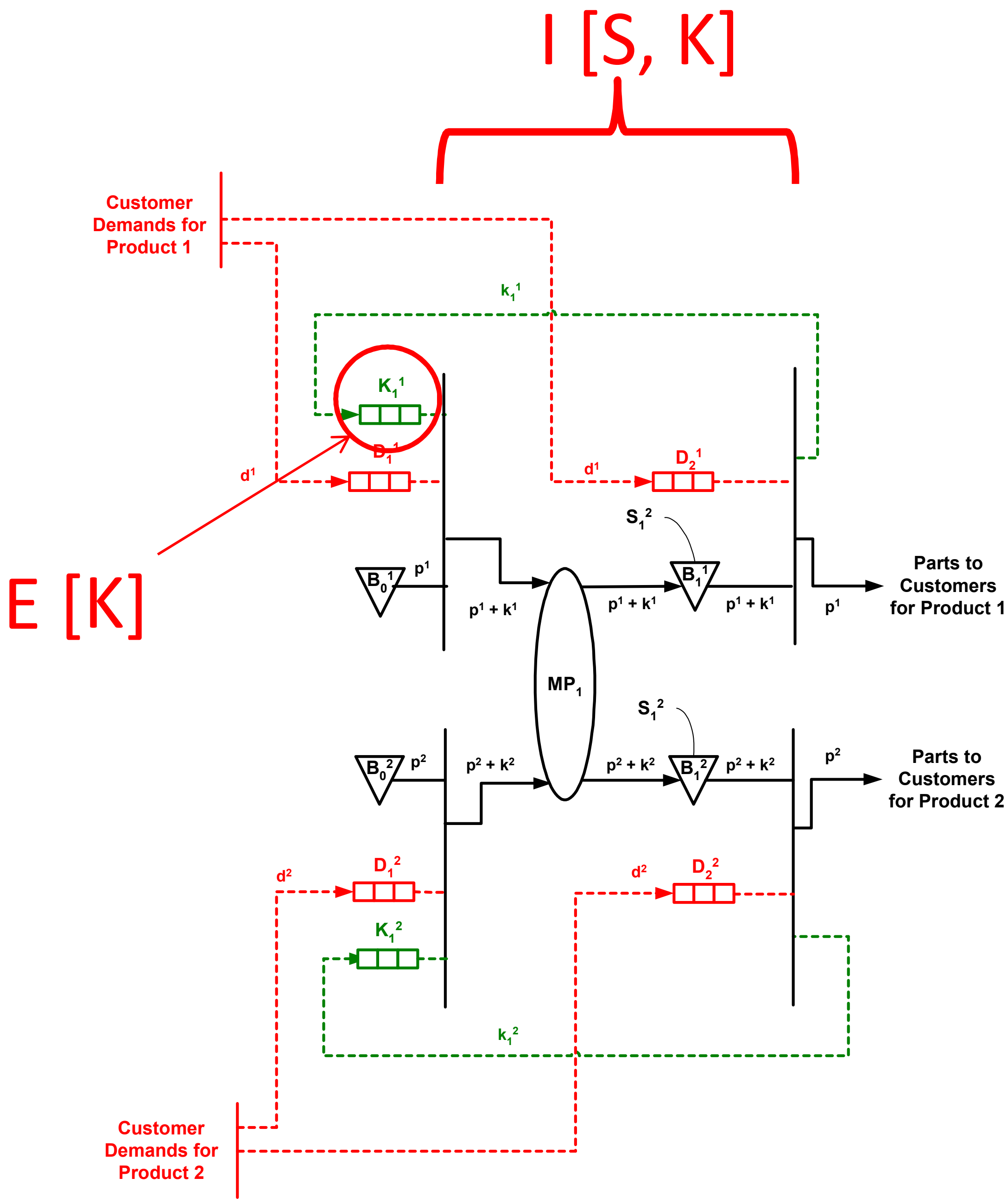

Figure E 1: Location of the expected inventory level, I [S,K] and expected undispatched kanbans, E[K], in a SS/MP/De-EKCS

E2 Expected Time Value of Stock-out, E{$t_{stock\text{-}out}$}

E{$t_{stock\text{-}out}$} represents the expected time value of a stock-out in the output buffer, B_1 of a particular product, in EKCS. Normally, E{$t_{stock\text{-}out}$} is denoted by π_0, which is defined in Markov chain theory as the long run average probability of time spent in state '0'. Figure E2 shows the location of E{$t_{stock\text{-}out}$} where the long run average stock-out time in buffer B_1 is calculated.

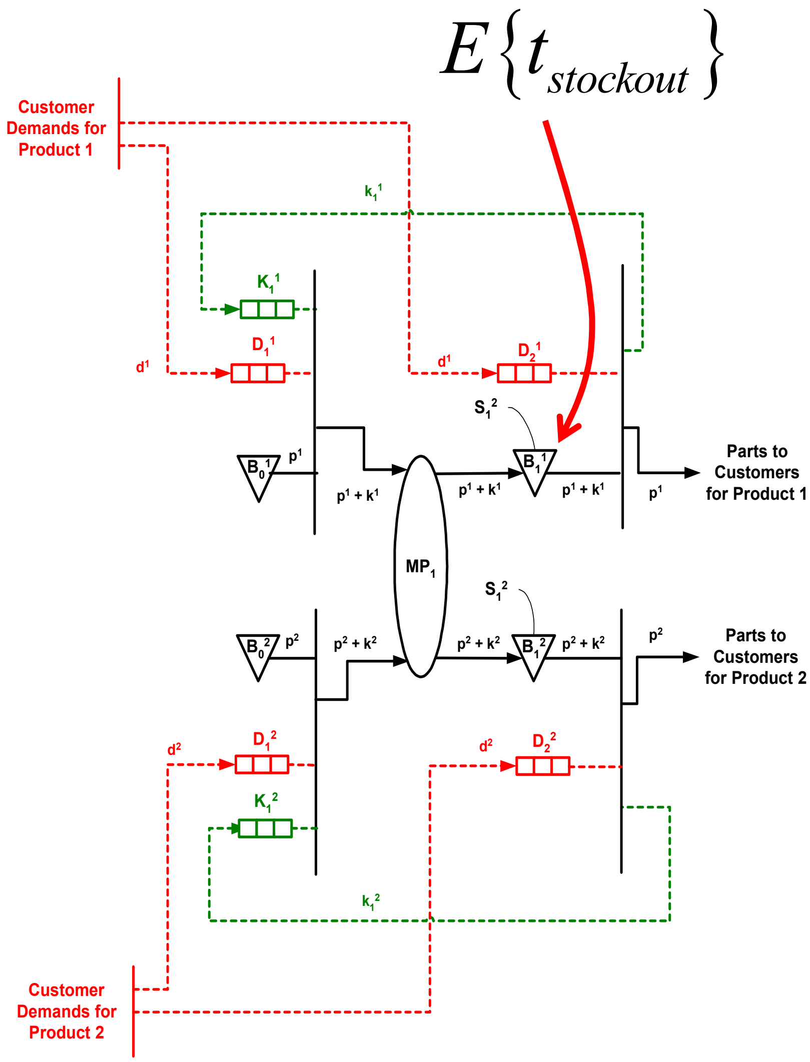

Figure E 2: Expected time value of stock-out in output buffer B_1 of SS/MP/De-EKCS

The state space of the Markov chain is developed around the output buffer of the particular product type. Figure D1, D2 and D3 below show the state space transition diagrams for the output buffer of a SS/MP/De-EKCS. They reflect the system transitions from state $\pi(x)$ to $\pi(x-1)$ at the rate λ (demand from successors). The system changes from state $\pi(x)$ to $\pi(x+1)$ whenever a product finishes processing at the MP. As the assumption is that L jobs are in process, service completions occur at the rate μ_A and the probability an arbitrary completion is of type j is equal to the proportion of in-process jobs of type j. If x is the actual on-hand inventory, this proportion is (S+K - x) / L and the processing rate should be (S+K - x) / L * μ_A.

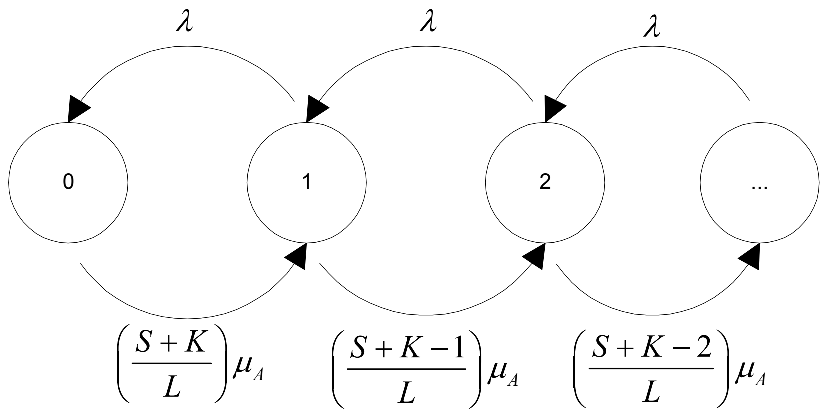

Figure E 3: Start of Markov chain for MP/De-EKCS

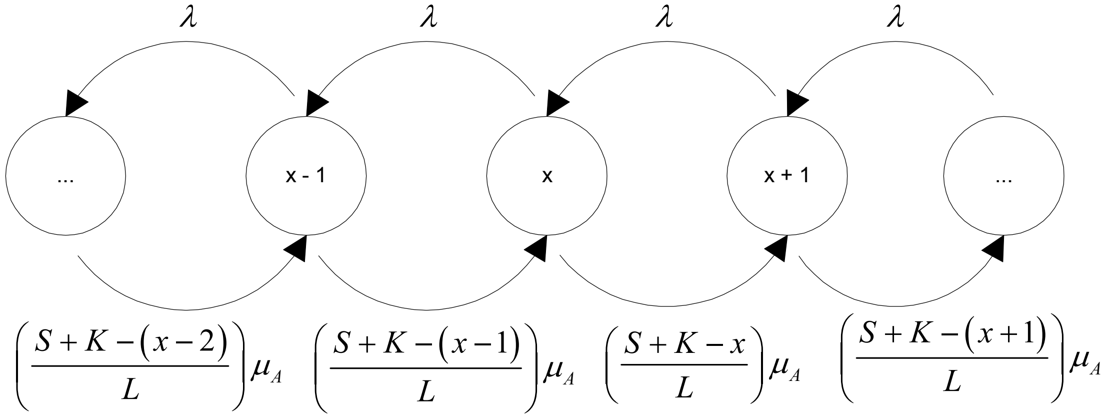

Figure E 4: Middle of the Markov chain for MP/De-EKCS

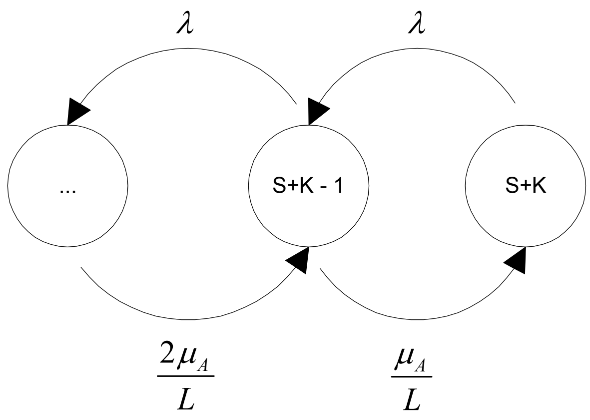

Figure E 5: End of Markov chain for MP/De-EKCS

Next, Ramakumar (1993) standard technique in stochastic processes is used to develop a coefficient matrix S. However, since the steps that follow are identical to those found in Appendix D, it is considered unnecessary to repeat them here. In a nutshell, the coefficient matrix, S, and Markov differential equations are used to generalize the steady state equations. Finally, the expected long run stock out probability, π_0, is obtained by means of induction. π_0, or $E\{t_{\text{stock-out}}\}$ comes out to be

$$E\{t_{stockout}\} = \frac{\lambda^{S+K}}{\sum_{x=0}^{S+K} \frac{(S+K)!}{(S+K-x)!} \lambda^{S+K-x} \rho^{x}} \tag{E4}$$

Both the MP/TKCS and De-EKCS have the same expected time value of a stock-out in their output buffers, B_1. In other words, the Markov chains for both are the same since their arrival and departure rates between states are identical.

E3 Expected Total Cost for SS/MP/De-EKCS

After substituting (E3) and (E4) into ETC expression, it can be simplified to:

$$\begin{aligned} ETC(S,K) &= C_h\left(I[S,K]\right) + C_s\left(E\{t_{stockout}\}\right) \\ &= C_h\left(S + E[K_1]\right) + C_s\left(\frac{\lambda^{S+K}}{\sum_{x=0}^{S+K} \frac{(S+K)!}{(S+K-x)!} \lambda^{S+K-x} \rho^{x}}\right) \end{aligned} \tag{E5}$$

APPENDIX F

Expected Total Cost (ETC) for SS/MP/Sh-EKCS

The ETC formulation for a Sh-EKCS is the same as De-EKCS as shown in equation (F1). It comprises of total expected inventory holding and shortage costs.

$$ETC\left(S_i, K_i\right) = \text{ Total Inventory Holding Cost + Total Shortage Cost}$$

$$= \sum_{i=1}^{m}\left[C_{h_i}\left(I\left[S_i, K_i\right]\right) + C_{s_i}\left(E\left\{t_{stockout_i}\right\}\right)\right] \tag{F1}$$

Total WIP for each product can be obtained separately ($S_i + K_i$; where i = 1, 2) (Askin et al., 1993) and hereafter the subscript 'i' is omitted. The optimal S* and K* for each product type is found by optimizing ETC for each type. Referring to just one specific product type, its ETC is

$$ETC\left(S, K\right) = C_h\left(I\left[S, K\right]\right) + C_s\left(E\left\{t_{stockout}\right\}\right) \tag{F2}$$

F1 Expected inventory level, I [S_i, K_i]

Figure F1 shows the position of the expected Inventory Level, I [S_1, S_2, K], and expected total number of undispatched kanbans, E[K], in an MP/Sh-EKCS. I [S_1, S_2, K] refers to the total WIP of all product types, both in the MP and the output buffer, B_1.

$$I\left[S_1, S_2, K\right] = L - E\left[K_1\right] \tag{F3}$$

where L is the total WIP for all product types in Sh-EKCS. Thus, L = S_1 + S_2 + K, and, E [K_1] is the expected total number of undispatched kanbans in the kanban queue.

$E\left[K_1\right] = K - \dfrac{\rho\left(K+S\right)}{K+S+1-\rho\left(K+S\right)} + \dfrac{\left(\rho\left[\dfrac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\dfrac{K+S}{K+S+1}\right]}$ as shown in Appendix A, If the state space of the Markov Chain is positioned on a kanban queue and its arrival rate is the same as its departure rate, the expected average number of kanbans is K/2. In this case, referring to the Shared Kanban queue, since both its arrival and departure rates are equal to the average customer demand arrival, λ, $E\left[K_1\right] = K - \dfrac{\rho\left(K+S\right)}{K+S+1-\rho\left(K+S\right)} + \dfrac{\left(\rho\left[\dfrac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\dfrac{K+S}{K+S+1}\right]}$ holds.

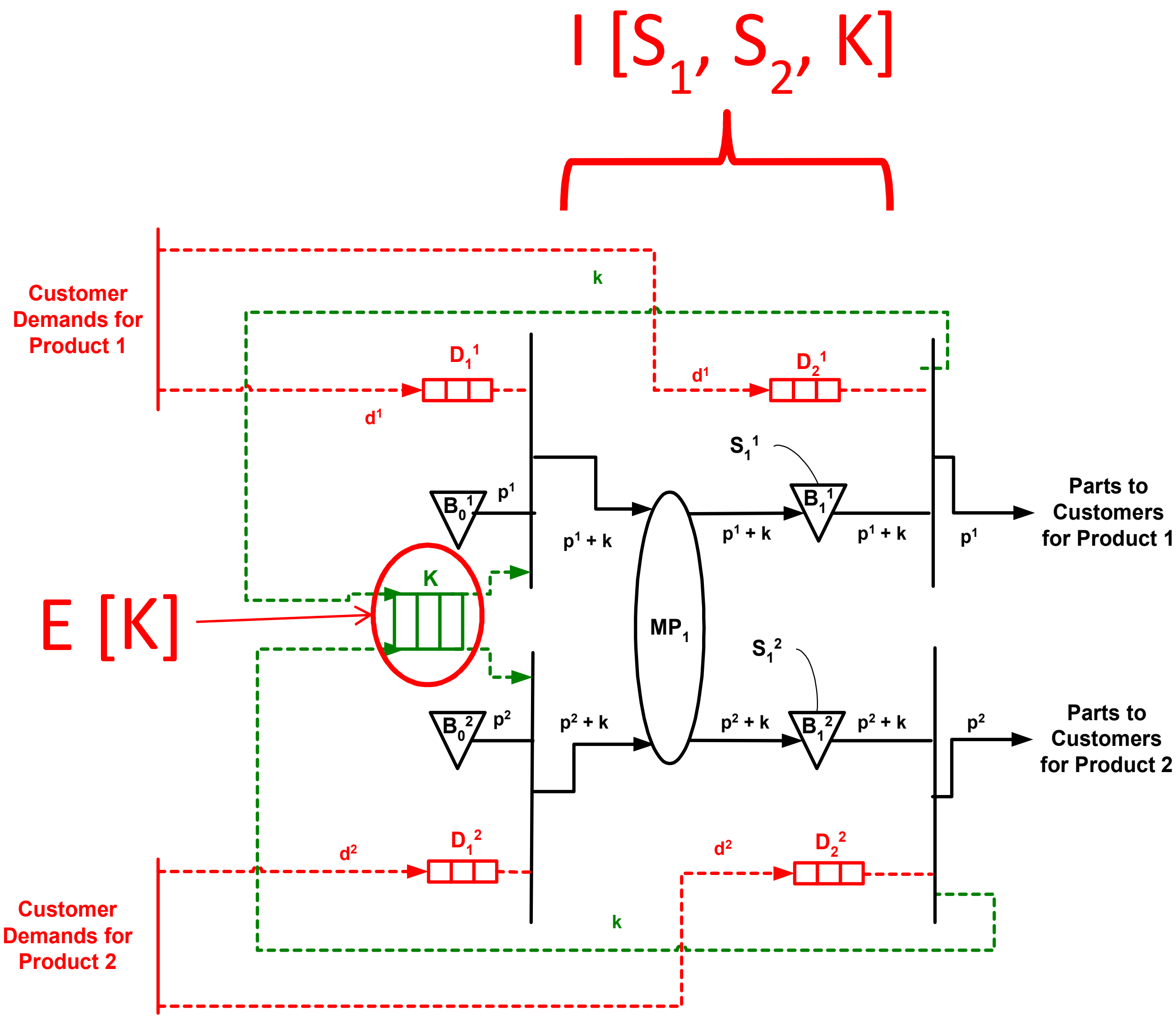

Figure F 1: Position of expected inventory level, I [S₁, S₂, K], and expected total number of undispatched kanbans, E [K], in an MP/Sh-EKCS

Thus,

$$I[S_1,S_2,K] = S_1 + S_2 + K - \left[K - \frac{\rho(K+S)}{K+S+1-\rho(K+S)} + \frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]} \right]$$

$$= S_1 + S_2 + \frac{\rho(K+S)}{K+S+1-\rho(K+S)} + \frac{\left(\rho\left[\frac{K+S}{K+S+1}\right]\right)^{K+S+1}}{1-\rho\left[\frac{K+S}{K+S+1}\right]} \qquad \text{(F4)}$$

Since $K = K_1 + K_2$ (once again noting that K_1 and K_2 are not dedicated kanbans but shared kanbans in the shared kanbans queue K), and if I [S_1, S_2, K] is separated according to its product type, then the

Expected WIP for each product type is

$$I\left[S_i, K_i\right] = S_i + \frac{\rho\left(K_i + S\right)}{K_i + S + 1 - \rho\left(K_i + S\right)} + \frac{\left(\rho\left[\frac{K_i + S}{K_i + S + 1}\right]\right)^{K_i + S + 1}}{1 - \rho\left[\frac{K_i + S}{K_i + S + 1}\right]} \tag{F5}$$

It turns out that the expected inventory level for Sh-EKCS is the same as De-EKCS.

F2 Expected Time Value of Stock-out, E{$t_{stock\text{-}out}$}

E{$t_{stock\text{-}out}$} represents the expected time value of stock-out in the output buffer, B_1, of Sh–EKCS. Normally, E{$t_{stock\text{-}out}$} is denoted by π_0, which is defined in Markov Chain theory as the long run average probability of time spent in state 0. Figure F2 shows the location of E{$t_{stock\text{-}out}$} where the long run average stock-out time in buffer B_1 is calculated, and represented by equation (E4) .

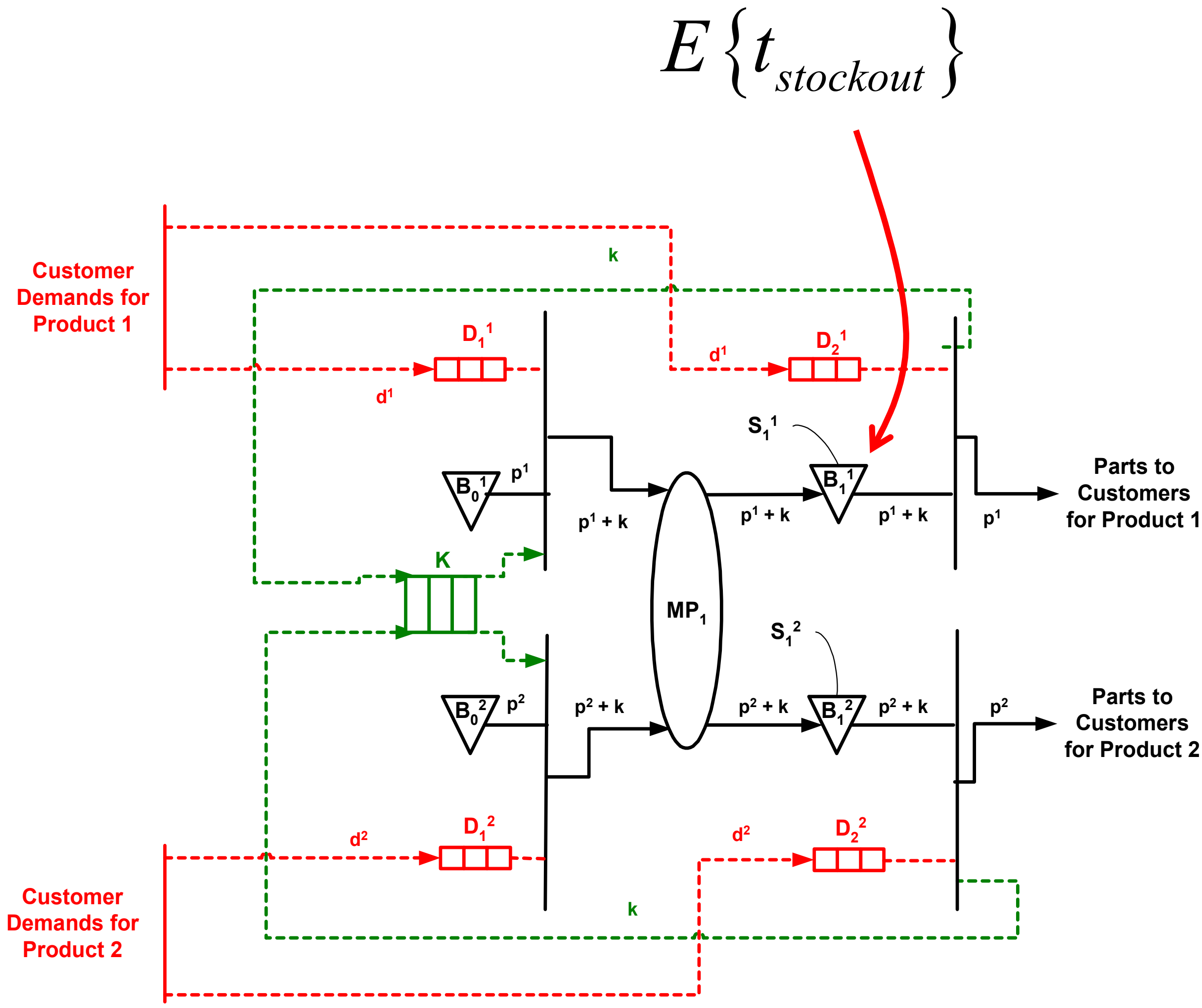

Figure F 2: Expected time value of a stock out, E {$t_{stock\text{-}out}$}

Next an explanation as to why E{$t_{stock\text{-}out}$} is the same for both Sh-EKCS and De-EKCS follows. In a Markov chain, he departure rate for a part type is obtained by multiplying the average

MP processing rate, μ, with the probability that a part exiting the MP is of that type. Referring to Figure F3, at the very start of the chain, its initial departure rate is $\left(\frac{S_i + K_i}{L}\right) * \mu$ (a scenario where the output buffer B_1 is empty).

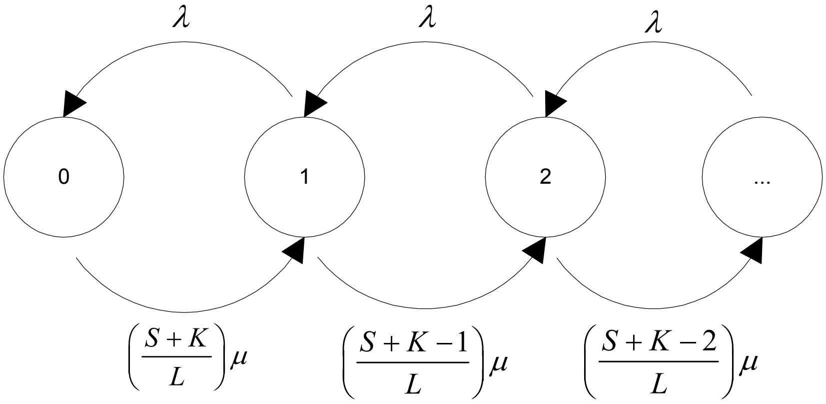

Figure F 3: Start of Markov Chain for MP/Sh-EKCS

APPENDIX G

Tables of Results for Simulation Comparison of SS/MP/KCS

Table G 3: Holding, Shortage and Backorder Costs for SS/ MP/ KCS

Product 1 (Expensive Item)		
Holding Cost ($ per unit per day)	C_h	10
Shortage Cost ($ per day)	C_s	100
Backorder Cost ($ per unit)	C_b	100
Product 2 (Cheap Item)		
Holding Cost ($ per unit per day)	C_h	5
Shortage Cost ($ per day)	C_s	20
Backorder Cost ($ per unit)	C_b	20

G1. MP Rate of Product Type 1 slower than Product Type 2 (50 to 90% Utilization Rates) – Scenario A1

Table G 4: Simulation Parameters for SS/ MP/ KCS – Scenario A1

Demand Arrival Rate for Product 1, λ_1	Demand Arrival Rate for Product 2, λ_2	Manufacturing Process (MP) Rate for Product 1, μ_1	Manufacturing Process (MP) Rate for Product 2, μ_2	Average Demand Arrival Rate, λ_A	Average Manufacturing Process (MP) Rate, μ_A	Percentage Average Utilization
(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(%)
1	2	3	8	3	5	0.58
2	3	6	9	5	8	0.67
3	4	9	10	7	10	0.73
4	5	10	11	9	11	0.85
5	6	11	12	11	12	0.95

Table G 5: Simulated Values for SS/ MP/ BS (Product 1) – Scenario A1

Base Stock (BS) (Product 1)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
1	3	2	387	387	0	2	1.7
2	6	4	748	748	0	4	0
3	9	5	1054	1054	0	5	0
4	10	7	1449	1449	0	7	0
5	11	8	1819	1819	0	8	0

Table G 6: Costs for SS/ MP/ BS (Product 1) – Scenario A1

Base Stock (BS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	20	7.08	27.08
0	40	0.00	40.00
0	50	0.00	50.00
0	70	0.00	70.00
0	80	0.00	80.00

Table G 7: Simulated Values for SS/ MP/ BS (Product 2) – Scenario A1

Base Stock (BS) (Product 2)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
2	8	3	722	722	0	3	3.6
3	9	4	1107	1107	0	3	1
4	10	6	1499	1499	0	5	1
5	11	7	1843	1843	0	4	4.8
6	12	9	2196	2196	0	4	7.11

Table G 8: Costs for SS/ MP/ BS (Product 2) – Scenario A1

Base Stock (BS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	15	3.00	18.00
0	15	0.83	15.83
0	25	0.83	25.83
0	20	4.00	24.00
0	20	5.93	25.93

Table G 9: Total Costs for SS/ MP/ BS (Both Products) – Scenario A1

Total Cost for Base Stock (BS) ($)
45.08
55.83
75.83
94.00
105.93

Table G 10: Simulated Values for SS/ MP/ TKCS (Product 1) – Scenario A1

Traditional Kanban Control System (TKCS) (Product 1)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
1	3	2	377	377	0	2	1.2
2	6	3	725	725	0	3	0.3
3	9	3	1107	1107	0	3	0.2
4	10	3	1448	1448	0	3	0.4
5	11	4	1860	1860	0	3	1.12

Table G 11: Costs for SS/ MP/ TKCS (Product 1) – Scenario A1

Traditional Kanban Control System (TKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	20	5.00	25.00
0	30	1.25	31.25
0	30	0.83	30.83
0	30	1.67	31.67
0	30	4.67	34.67

Table G 12: Simulated Values for SS/ MP/ TKCS (Product 2) – Scenario A1

Traditional Kanban Control System (TKCS) (Product 2)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
2	8	2	718	718	0	2	8.3
3	9	2	1083	1083	0	2	2.5
4	10	3	1428	1428	0	2	3.1
5	11	4	1810	1810	0	2	6.4
6	12	4	2150	2150	0	3	3.8

Table G 13: Costs for SS/ MP/ TKCS (Product 2) – Scenario A1

Traditional Kanban Control System (TKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	10	6.92	16.92
0	10	2.08	12.08
0	10	2.58	12.58
0	10	5.33	15.33
0	15	3.17	18.17

Table G 14: Total Costs for SS/ MP/ TKCS (Both Products) – Scenario A1

Total Cost for Traditional Kanban Control System (TKCS) ($)
41.92
43.33
43.42
47.00
52.83

Table G 15: Simulated Values for SS/ MP/De- EKCS (Product 1) – Scenario A1

Dedicated Extended Kanban Control System (De - EKCS) (Product 1)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
1	3	2	2	369	369	0	2	1.1
2	6	2	2	750	750	0	2	0.65
3	9	2	3	1039	1039	0	2	0.64
4	10	2	5	1426	1426	0	2	0.97
5	11	2	4	1780	1780	0	2	1.27

Table G 16: Costs for SS/ MP/ De–EKCS (Product 1) – Scenario A1

Dedicated Extended Kanban Control System (De - EKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	20	4.58	24.58
0	20	2.71	22.71
0	20	2.67	22.67
0	20	4.04	24.04
0	20	5.29	25.29

Table G 17: Simulated Values for SS/ MP/ De–EKCS (Product 2) – Scenario A1

Dedicated Extended Kanban Control System (De - EKCS) (Product 2)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
2	8	0	3	735	735	0	0	12.2
3	9	2	3	1102	1102	0	2	3.36
4	10	2	3	1489	1489	0	2	4.65
5	11	2	3	1860	1860	0	2	6.4
6	12	3	3	2176	2176	0	2	7.9

Table G 18: Costs for SS/ MP/ De–EKCS (Product 2) – Scenario A1

Dedicated Extended Kanban Control System (De - EKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	10.17	10.17
0	10	2.80	12.80
0	10	3.88	13.88
0	10	5.33	15.33
0	10	6.58	16.58

Table G 19: Total Cost for SS/ MP/ De–EKCS (Both Products) – Scenario A1

Total Cost for Dedicated Extended Kanban Control System (De - EKCS) ($)
34.75
35.51
36.54
39.38
41.88

Table G 20: Simulated Values for SS/ MP/Sh – EKCS (Product 1) – Scenario A1

Shared Extended Kanban Control System (Sh - EKCS) (Product 1)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
1	3	2	5	398	398	0	2	0.97
2	6	2	4	749	749	0	2	0.7
3	9	2	4	1043	1043	0	2	0.66
4	10	2	3	1423	1423	0	2	0.9
5	11	2	4	1856	1856	0	2	1.35

Table G 21: Costs for SS/ MP/Sh–EKCS (Product 1) – Scenario A1

Shared Extended Kanban Control System (Sh - EKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	20	4.04	24.04
0	20	2.92	22.92
0	20	2.75	22.75
0	20	3.75	23.75
0	20	5.63	25.63

Table G 22: Simulated Values for SS/MP/Sh–EKCS (Product 2) – Scenario A1

Shared Extended Kanban Control System (Sh - EKCS) (Product 2)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
2	8	0	5	700	700	0	0	11.7
3	9	2	4	1107	1107	0	2	3
4	10	2	4	1483	1483	0	2	3.68
5	11	2	3	1880	1880	0	2	6.4
6	12	3	4	2136	2136	0	2	8.8

Table G 23: Costs for SS/MP/Sh–EKCS (Product 2) – Scenario A1

Shared Extended Kanban Control System (Sh - EKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	9.75	9.75
0	10	2.50	12.50
0	10	3.07	13.07
0	10	5.33	15.33
0	10	7.33	17.33

Table G 24: Total Cost for SS/MP/Sh–EKCS (Both Products) – Scenario A1

Total Cost for Shared Extended Kanban Control System (Sh - EKCS) ($)
33.79
35.42
35.82
39.08
42.96

G2. MP Rate of Product Type 1 faster than Product Type 2 (50 to 90% Utilization Rates) – Scenario A2

Table G 25: Simulation Parameters for SS/MP/KCS – Scenario A2

Demand Arrival Rate for Product 1, λ_1	Demand Arrival Rate for Product 2, λ_2	Manufacturing Process (MP) Rate for Product 1, μ_1	Manufacturing Process (MP) Rate for Product 2, μ_2	Average Demand Arrival Rate, λ_A	Average Manufacturing Process (MP) Rate, μ_A	Percentage Average Utilization
(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(%)
1	2	8	5	3	6	0.53
2	3	9	7	5	8	0.65
3	4	10	9	7	9	0.74
4	5	11	10	9	10	0.86
5	6	12	11	11	11	0.96

Table G 26: Simulated Values for SS/MP /BS (Product 1) – Scenario A2

Base Stock (BS) (Product 1)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
1	8	2	387	387	0	2	0.21
2	9	4	748	748	0	4	0.02
3	10	5	1054	1054	0	5	0.02
4	11	7	1449	1449	0	7	0.02
5	12	8	1819	1819	0	7	0

Table G 27: Costs for SS – MP – BS (Product 1) – Scenario A2

Base Stock (BS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	20	0.88	20.88
0	40	0.08	40.08
0	50	0.08	50.08
0	70	0.08	70.08
0	70	0.00	70.00

Table G 28: Simulated Values for SS/MP/BS (Product 2) – Scenario A2

Base Stock (BS) (Product 2)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
2	5	3	722	722	0	3	0.74
3	7	4	1107	1107	0	3	0.94
4	9	6	1499	1499	0	5	0.92
5	10	7	1843	1843	0	4	4.56
6	11	10	2196	2196	0	8	7.05

Table G 29: Costs for SS/MP/BS (Product 2) – Scenario A2

Base Stock (BS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	15	0.62	15.62
0	15	0.78	15.78
0	25	0.77	25.77
0	20	3.80	23.80
0	40	5.88	45.88

Table G 30: Total Costs for SS/MP/BS (Both Products) – Scenario A2

Total Cost for Base Stock (BS) ($)
36.49
55.87
75.85
93.88
115.88

Table G 31: Simulated Values for SS/MP/TKCS (Product 1) – Scenario A2

Traditional Kanban Control System (TKCS) (Product 1)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
1	8	2	354	354	0	2	0.08
2	9	2	751	751	0	2	0.4
3	10	2	1087	1087	0	2	0.66
4	11	2	1394	1394	0	2	0.68
5	12	4	1863	1863	0	4	0.5

Table G 32: Costs for SS/MP/TKCS (Product 1) – Scenario A2

Traditional Kanban Control System (TKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	20	0.33	20.33
0	20	1.67	21.67
0	20	2.75	22.75
0	20	2.83	22.83
0	40	2.08	42.08

Table G 33: Simulated Values for SS/MP/TKCS (Product 2) – Scenario A2

Traditional Kanban Control System (TKCS) (Product 2)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
2	5	2	767	767	0	2	2.15
3	7	2	1083	1083	0	2	2.57
4	9	2	1492	1492	0	1	9.3
5	10	3	1800	1800	0	2	5.9
6	11	6	2112	2112	0	2	8

Table G 34: Costs for SS/MP/TKCS (Product 2) – Scenario A2

Traditional Kanban Control System (TKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	10	1.79	11.79
0	10	2.14	12.14
0	5	7.75	12.75
0	10	4.92	14.92
0	10	6.67	16.67

Table G 35: Total Costs for SS/MP/TKCS (Both Products) – Scenario A2

Total Cost for Traditional Kanban Control System (TKCS) ($)
32.13
33.81
35.50
37.75
58.75

Table G 36: Simulated Values for SS/MP/De–EKCS (Product 1) – Scenario A2

Dedicated Extended Kanban Control System (De - EKCS) (Product 1)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
1	8	1	3	387	387	0	1	1.2
2	9	1	4	742	742	0	1	1.5
3	10	1	4	1069	1069	0	1	1.6
4	11	1	5	1452	1452	0	1	2.36
5	12	3	3	1771	1771	0	3	0.34

Table G 35: Costs for SS/MP/De–EKCS (Product 1) – Scenario A2

Dedicated Extended Kanban Control System (De - EKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	10	5.00	15.00
0	10	6.25	16.25
0	10	6.67	16.67
0	10	9.83	19.83
0	30	1.42	31.42

Table G 37: Simulated Values for SS/MP/De–EKCS (Product 2) – Scenario A2

Dedicated Extended Kanban Control System (De - EKCS) (Product 2)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
2	5	0	4	721	721	0	0	8.7
3	7	1	4	1102	1102	0	1	5
4	9	1	4	1459	1459	0	1	7.14
5	10	2	6	1828	1828	0	1	8.68
6	11	4	2	2163	2163	0	1	9.32

Table G 38: Costs for SS/ MP/De–EKCS (Product 2) – Scenario A2

Dedicated Extended Kanban Control System (De - EKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	7.25	7.25
0	5	4.17	9.17
0	5	5.95	10.95
0	5	7.23	12.23
0	5	7.77	12.77

Table G 39: Total Costs for SS/ MP/De–EKCS (Both Products) – Scenario A2

Total Cost for Dedicated Extended Kanban Control System (De - EKCS) ($)
22.25
25.42
27.62
32.07
44.18

Table G 40: Simulated Values for SS/MP/Sh–EKCS (Product 1) – Scenario A2

Shared Extended Kanban Control System (Sh - EKCS) (Product 1)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
1	8	1	3	382	382	0	1	0.94
2	9	1	3	740	740	0	1	1.67
3	10	1	4	1054	1054	0	1	1.7
4	11	1	5	1449	1449	0	1	2.5
5	12	3	6	1759	1759	0	3	0.54

Table G 41: Costs for SS/MP/Sh–EKCS (Product 1) – Scenario A2

Shared Extended Kanban Control System (Sh - EKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	10	3.92	13.92
0	10	6.96	16.96
0	10	7.08	17.08
0	10	10.42	20.42
0	30	2.25	32.25

Table G 42: Simulated Values for SS/MP/Sh–EKCS (Product 2) – Scenario A2

Shared Extended Kanban Control System (Sh - EKCS) (Product 2)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
2	5	0	3	735	735	0	0	9.14
3	7	1	3	1105	1105	0	1	5.2
4	9	2	4	1499	1499	0	1	5.1
5	10	2	5	1843	1843	0	1	7.8
6	11	4	6	2179	2179	0	1	9.7

Table G 43: Costs for SS/MP/Sh–EKCS (Product 2) – Scenario A2

Shared Extended Kanban Control System (Sh - EKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	7.62	7.62
0	5	4.33	9.33
0	5	4.25	9.25
0	5	6.50	11.50
0	5	8.08	13.08

Table G 44: Total Costs for SS/MP/Sh–EKCS (Both Products) – Scenario A2

Total Cost for Shared Extended Kanban Control System (Sh - EKCS) ($)
21.53
26.29
26.33
31.92
45.33

G3. Tables of Results for SS – MP – KCS Before Validation (MP Rate of Product Type 1 faster than Product Type 2 (50 to 90% Utilization Rates)) – Scenario A3

Table G 45: Simulated Values for SS/MP/BS (Product 1) – Scenario A3

Base Stock (BS) (Product 1)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
1	8	2	388	388	0	2	0.23
2	9	4	748	748	0	4	0.02
3	10	5	1053	1053	0	5	0.02
4	11	7	1448	1448	0	7	0.04
5	12	8	1819	1819	0	7	0

Table G 46: Costs for SS/MP/BS (Product 1) – Scenario A3

Base Stock (BS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	20	0.96	20.96
0	40	0.08	40.08
0	50	0.08	50.08
0	70	0.17	70.17
0	70	0.00	70.00

Table G 47: Simulated Values for SS/MP/BS (Product 2) – Scenario A3

Base Stock (BS) (Product 2)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
2	5	3	722	722	0	3	0.74
3	7	4	1107	1107	0	3	0.94
4	9	6	1500	1500	0	5	0.88
5	10	7	1843	1843	0	4	4.56
6	11	10	2198	2198	0	8	5.88

Table G 48: Costs for SS/MP/BS (Product 2) – Scenario A3

Base Stock (BS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	15	0.62	15.62
0	15	0.78	15.78
0	25	0.73	25.73
0	20	3.80	23.80
0	40	4.90	44.90

Table G 49: Total Costs for SS/MP/BS (Both Products) – Scenario A3

Total Cost for Base Stock (BS) ($)
36.58
55.87
75.82
93.97
114.90

Table G 50: Simulated Values for SS/MP/TKCS (Product 1) – Scenario A3

Traditional Kanban Control System (TKCS) (Product 1)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
1	8	1	353	352	1	1	0.1
2	9	2	752	752	0	2	0.43
3	10	2	1087	1087	0	2	0.66
4	11	2	1394	1394	0	2	0.68
5	12	3	1763	1762	1	3	1.9

Table G 51: Costs for SS/MP/TKCS (Product 1) – Scenario A3

Traditional Kanban Control System (TKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
100	10	0.42	110.42
0	20	1.79	21.79
0	20	2.75	22.75
0	20	2.83	22.83
100	30	7.92	137.92

Table G 52: Simulated Values for SS/MP/TKCS (Product 2) – Scenario A3

Traditional Kanban Control System (TKCS) (Product 2)							
Demand Arrival Rate	Manufacturing Process Rate	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)	(Units)	(Units)	(Units)	(Units)	(Hours)
2	5	2	757	757	0	2	2.2
3	7	2	1083	1083	0	2	2.57
4	9	2	1499	1499	0	1	9.1
5	10	3	1801	1801	0	2	6
6	11	4	2001	2001	0	3	7

Table G 53: Costs for SS/MP/TKCS (Product 2) – Scenario A3

Traditional Kanban Control System (TKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	10	1.83	11.83
0	10	2.14	12.14
0	5	7.58	12.58
0	10	5.00	15.00
0	15	5.83	20.83

Table G 54: Total Costs for SS/MP/TKCS (Both Products) – Scenario A3

Total Cost for Traditional Kanban Control System (TKCS) ($)
122.25
33.93
35.33
37.83
158.75

Table G 55: Simulated Values for SS/MP/De–EKCS (Product 1) – Scenario A3

Dedicated Extended Kanban Control System (De - EKCS) (Product 1)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
1	8	1	3	388	388	0	1	1.21
2	9	1	4	742	742	0	1	1.56
3	10	1	4	1070	1070	0	1	1.6
4	11	1	5	1455	1455	0	1	2.4
5	12	2	3	1772	1771	1	2	1.2

Table G 56: Costs for SS/MP/De–EKCS (Product 1) – Scenario A3

Dedicated Extended Kanban Control System (De - EKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	10	5.04	15.04
0	10	6.50	16.50
0	10	6.67	16.67
0	10	10.00	20.00
100	20	5.00	125.00

Table G 57: Simulated Values for SS/MP/De–EKCS (Product 2) – Scenario A3

Dedicated Extended Kanban Control System (De - EKCS) (Product 2)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
2	5	0	4	721	721	0	0	8.6
3	7	1	4	1102	1102	0	1	5.2
4	9	1	4	1460	1460	0	1	6.88
5	10	2	6	1829	1829	0	1	8.68
6	11	4	2	2163	2163	0	1	9.32

Table G58: Costs for SS/MP/De–EKCS (Product 1) – Scenario A3

Dedicated Extended Kanban Control System (De - EKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	7.17	7.17
0	5	4.33	9.33
0	5	5.73	10.73
0	5	7.23	12.23
0	5	7.77	12.77

Table G 59: Total Costs for SS/MP/De–EKCS (Both Products) – Scenario A3

Total Cost for Dedicated Extended Kanban Control System (De - EKCS) ($)
22.21
25.83
27.40
32.23
137.77

Table G 60: Simulated Values for SS/MP/Sh–EKCS (Product 1) – Scenario A3

Shared Extended Kanban Control System (Sh - EKCS) (Product 1)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
1	8	1	3	381	381	0	1	0.89
2	9	1	3	740	740	0	1	1.66
3	10	1	4	1055	1055	0	1	1.7
4	11	1	5	1439	1439	0	1	2.49
5	12	2	6	1759	1759	0	2	0.55

Table G 61: Costs for SS/MP/Sh–EKCS (Product 1) – Scenario A3

Shared Extended Kanban Control System (Sh - EKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	10	3.71	13.71
0	10	6.92	16.92
0	10	7.08	17.08
0	10	10.38	20.38
0	30	2.29	32.29

Table G 62: Simulated Values for SS/MP/Sh–EKCS (Product 2) – Scenario A3

Shared Extended Kanban Control System (Sh - EKCS) (Product 2)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
2	5	0	3	735	735	0	0	9.14
3	7	1	3	1105	1105	0	1	5.2
4	9	2	4	1498	1498	0	1	5.2
5	10	2	5	1844	1844	0	1	7.6
6	11	3	6	2180	2179	1	1	11.8

Table G 63: Costs for SS/ MP/Sh–EKCS (Product 2) – Scenario A3

Shared Extended Kanban Control System (Sh - EKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	7.62	7.62
0	5	4.33	9.33
0	5	4.33	9.33
0	5	6.33	11.33
20	5	9.83	34.83

Table GA 64: Total Costs for SS/MP/Sh–EKCS (Both Products) – Scenario A3

Total Cost for Shared Extended Kanban Control System (Sh - EKCS) ($)
21.33
26.25
26.42
31.71
67.13

APPENDIX H

Tables of Results for Simulation Comparison of SS/MP/De and Sh EKCS (Comparing 10 to 50% Utilization Rates) – Scenario B

Table H 1: Simulation Parameters for SS/MP/De and Sh–EKCS – Scenario B

Demand Arrival Rate for Product 1, λ_1	Demand Arrival Rate for Product 2, λ_2	Manufacturing Process (MP) Rate for Product 1, μ_1	Manufacturing Process (MP) Rate for Product 2, μ_2	Average Demand Arrival Rate, λ_A	Average Manufacturing Process (MP) Rate, μ_A	Percentage Average Utilization
(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(Units per Day)	(%)
1	2	16	21	3	19	16
2	3	17	22	5	20	25
3	4	18	23	7	21	34
4	5	19	24	9	21	42
5	6	20	25	11	22	49

Table H 2: Simulated Values for SS/MP/De–EKCS (Product 1) – Scenario B

Dedicated Extended Kanban Control System (De - EKCS) (Product 1)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
1	16	0	2	388	388	0	0	1.75
2	17	0	2	737	737	0	0	1.78

3	18	1	3	1047	1047	0	1	0.29
4	19	1	5	1466	1466	0	1	0.48
5	20	1	5	1837	1837	0	1	0.52

Table H 3: Costs for SS/MP/De–EKCS (Product 1) – Scenario B

Dedicated Extended Kanban Control System (De - EKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	7.29	7.29
0	0	7.42	7.42
0	10	1.21	11.21
0	10	2.00	12.00
0	10	2.17	12.17

Table H 4: Simulated Values for SS/MP/De–EKCS (Product 2) – Scenario B

Dedicated Extended Kanban Control System (De - EKCS) (Product 2)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
2	21	0	2	1102	1102	0	0	1.36
3	22	0	2	1126	1126	0	0	1.5
4	23	0	2	1499	1499	0	0	1.66
5	24	0	5	1844	1844	0	0	2.04
6	25	0	5	2195	2195	0	0	2.36

Table H 5: Costs for SS/MP/De–EKCS (Product 2) – Scenario B

Dedicated Extended Kanban Control System (De - EKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	1.13	1.13
0	0	1.25	1.25
0	0	1.38	1.38
0	0	1.70	1.70
0	0	1.97	1.97

Table H 6 Total Costs for SS/MP/De–EKCS (Both Products) – Scenario B

Total Cost for Dedicated Extended Kanban Control System (De - EKCS) ($)
8.43
8.67
12.59
13.70
14.13

Table H 7: Simulated Values for SS/MP/Sh–EKCS (Product 1) – Scenario B

Shared Extended Kanban Control System (Sh - EKCS) (Product 1)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
1	16	0	3	385	385	0	0	1.8
2	17	0	3	735	735	0	0	1.6
3	18	1	5	1050	1050	0	1	0.3
4	19	1	5	1455	1455	0	1	0.51
5	20	1	5	1833	1833	0	1	0.61

Table H 8 Costs for SS/MP/Sh–EKCS (Product 1) – Scenario B

Shared Extended Kanban Control System (Sh - EKCS) (Product 1)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	7.50	7.50
0	0	6.67	6.67
0	10	1.25	11.25
0	10	2.13	12.13
0	10	2.54	12.54

Table H 9: Simulated Values for SS/MP/Sh–EKCS (Product 2) – Scenario B

Shared Extended Kanban Control System (Sh - EKCS) (Product 2)								
Demand Arrival Rate	Manufacturing Process Rate	Optimal Base Stock Level, S*	Optimal Kanban Number, K*	Parts In	Parts Out	Backorders	Average Inventory Level, B_1	Average Customer Waiting Time
(Units per day)	(Units per day)	(from MATLAB)		(Units)	(Units)	(Units)	(Units)	(Hours)
2	21	0	3	724	724	0	0	1.3
3	22	0	3	1121	1121	0	0	1.48
4	23	0	5	1478	1478	0	0	1.68
5	24	0	5	1833	1833	0	0	1.96
6	25	0	5	2190	2190	0	0	2.4

Table H 10 Costs for SS/MP/Sh–EKCS (Product 2) – Scenario B

Shared Extended Kanban Control System (Sh - EKCS) (Product 2)			
Total Backorder Cost ($)	Total Holding Cost ($)	Total Shortage Cost ($)	Actual Total Cost (ATC) ($)
0	0	1.08	1.08
0	0	1.23	1.23
0	0	1.40	1.40
0	0	1.63	1.63
0	0	2.00	2.00

Table H 11 Total Costs for SS/MP/Sh–EKCS (Both Products) – Scenario B

Total Cost for Shared Extended Kanban Control System (Sh - EKCS) ($)
8.58
7.90
12.65
13.76
14.54

APPENDIX I

MATLAB Codes

I1 MATLAB Program Optimizing SS/SP/EKCS

```
%This program finds the optimal K* and S* in Single Stage EKCS. It
comprises of 4 main parts
%1st part: Find range of S*+K, in which the optimal number of base stock,
%S* lies
%2nd part: Find range of S+K*, in which the optimal number of kanbans,
%K* lies
%3rd part: Exhaustive search over the ranges of (S*,K) and (S,K*) to obtain
%the lowest ETC (S*, K*)
%4th part: Plot the ETC Curve against S and K

%-----------------------------------------------------------------------------
--%
% Request User Input: Lambda, Miu, Cs, Ch
%-----------------------------------------------------------------------------
--%
clear %clear off all previous values stored in computer

%Use bank format i.e. 2 dec places
format bank

%Request input from user
lambda = input ( '\n\n Enter demand arrival rate (lambda): ' );

if lambda <=0
    fprintf ( ' The demand arrival rate you entered is invalid! \n Please
try again...');
    break;
end

miu = input ( ' Enter MP processing rate (miu): ' );

if (miu <=0)
    fprintf ( ' The MP processing rate you entered is invalid! \nPlease try
again...');
    break;
end

if (lambda >= miu)
    fprintf ( ' The demand arrival rate cannot exceed or equal the MP
processing rate! \nPlease try again...');
    break;
end


p =  lambda/miu;

Cs = input ( ' Enter Shortage Costs per time (Cs): ' );
Ch = input ( ' Enter Holding Costs per unit per time(Ch): ' );

if (Cs < Ch)
    fprintf ( ' The Shortage Cost cannot be lesser than the Holding Costs!
\nPlease try again...');
```

```
    break;
end

if (Cs <=0) || (Ch <=0)
    fprintf ( ' The Shortage Cost or Holding Cost you entered is invalid!
\nPlease try again...');
    break;
end


%-----------------------------------------------------------------------%
% 1st part: Find range of S*+K, in which the optimal number of base stock,
%S* lies
%-----------------------------------------------------------------------%

x = Ch/(Cs*(p-1)^2);

syms g

%We solve for the LHS range first, where g = (S*+K)1
a = solve( x - ((p^(g+1)) / (p^(g+2)-1)*(p^(g+1)-1))  , g);

fprintf ('\n The LHS range for S*+K gives: %d', ceil(double (a)))
fprintf ('\n');

syms h

%Now we solve for the RHS range, where h = (S*+K)2
b = solve( ((p^h) / (p^h-1)*(p^(h+1)-1)) - x , h);

fprintf ('\n The RHS range for S*+K gives: %d', ceil(double (b)))
fprintf ('\n');


%-----------------------------------------------------------------------%
%2nd part: Find range of S+K*, in which the optimal number of kanbans,
%K* lies
%-----------------------------------------------------------------------
-%

y = Ch/ (2*(Cs*(p-1)^2));

syms i

%We solve for the LHS range first, where i = (S+K*)1
c = solve( ((p^(i+1)) / (p^(i+2)-1)*(p^(i+1)-1)) - y , i);

fprintf ('\n The LHS range for S+K* gives: %d', ceil(double (c)))
fprintf ('\n');


syms j

%Now we solve for the RHS range, where j = (S+K*)2
d = solve( y - ((p^j) / (p^j-1)*(p^(j+1)-1)) , j);

fprintf ('\n The RHS range for S+K* gives: %d', ceil(double (d)))
fprintf ('\n');

%-----------------------------------------------------------------------%
%3rd part: Exhaustive search over the ranges of (S*,K) and (S,K*) to obtain
```

```
%the lowest ETC (S*, K*)
%--------------------------------------------------------------------------%

 LHS = input('\n\n Enter LHS Range for (S*+K*): ');
 RHS = input('\n\n Enter RHS Range for (S*+K*): ');

 fprintf('S      K      S+K      ETC(S,K)\n');
 fprintf('----------------------------');


 LTC = 1e100000;

 for S = LHS : RHS
     for K = LHS : RHS

         U = S+K;
         ETC = Ch*(S+K/2) + Cs*( (p-1) / (p^(S+K+1)-1));

         if U <= RHS

             fprintf('\n%d     %d     %d        $%2.2f', S, K, U, ETC);

             if ETC < LTC
                 LTC = ETC;
                 V = S;
                 W = K;
             end
         end
     end
 end

 fprintf('\nThe optimal Base Stock, S* is %d', V);
 fprintf('\nThe optimal No. Of Kanbans, K* is %d', W);
 fprintf('\nThe Lowest Expected Total Cost, ETC(S*,K*) is $%2.2f\n', LTC);

%--------------------------------------------------------------------------%
 %4th part: Plot the ETC Curve against S and K
%--------------------------------------------------------------------------%

%Let F == ETC, X1 == S (no. of base stock) and X2 == K (no. of kanbans)
F = @(X1,X2) Ch*(X1+X2/2) + Cs*( (p-1) / (p^(X1+X2+1)-1));

colormap(jet)    % sets the default colors
ezmeshc(F,[0,50],[0,50]) % draws a mesh with contours underneath


xlabel('Base Stock Level');
ylabel('Number of Kanbans');
zlabel('Expected Total Cost, E(TC)')
grid on
```

I2 MATLAB Program Optimizing SS/SP/BS

```
%This program uses the method and equations of Hopp and Spearman (2008) in
their book Factory Physics to compute the Optimal Base Stock Level.
%It comprises of 3 parts:
% 1st part: Find the optimal base stock
% 2nd part: Plot the base stock level vs. the E(TC) equation
%----------------------------------------------------------------------------
----------------------------------------------------------------------------
-%

%Use bank format i.e. 2 dec places
format bank

%Request input from user
lambda = input ( '\n\nEnter demand arrival rate (lambda): ' );

if lambda <=0
    fprintf ( 'The demand arrival rate you entered is invalid! \nPlease try
again...');
    break;
end

Cb = input ( 'Enter Backorder Costs per unit per time (Cb): ' );
Ch = input ( 'Enter Holding Costs per unit per time(Ch): ' );

if (Cb < Ch)
    fprintf ( 'The Backorder Costs cannot be lesser than the Holding Costs!
\nPlease try again...');
    break;
end

if (Cb <=0) || (Ch <=0)
    fprintf ( 'The Backorder Cost or Holding Cost you entered is invalid!
\nPlease try again...');
    break;
end

p = Cb /(Cb+Ch);


%----------------------------------------------------------------------------
----------------------------------------------------------------------------
-%
```

```
% 1st part: Find the optimal base stock
%-----------------------------------------------------------------------
-----------------------------------------------------------------------
-%

if lambda >= 10
    sigma = sqrt(lambda);                                        %Find the
standard deviation
    bs = norminv(p, lambda, sigma);                              %Find the
normal inverse  i.e. bs = the optimal base stock level
    fprintf ( '\n\n The optimal Base Stock Level is: %6.0f\n\n',
bs);     %Display the optimal Base Stock if demand arrival rate is more
than or equal to 10 --> use Normal Distribution

else
    bs = poissinv(p, lambda);
    fprintf ( '\n\n The optimal Base Stock Level is: %6.0f\n\n',
bs);     %Display the optimal Base Stock if demand arrival rate is less
than 10 --> use Poisson Distribution

end


%-----------------------------------------------------------------------
-----------------------------------------------------------------------
-%
% 2nd part: Plot the base stock level vs. the E(TC) equation
%-----------------------------------------------------------------------
-----------------------------------------------------------------------
-%

Fs = @(Z) Ch * (Z - lambda + (lambda * poisspdf(Z,lambda) + (lambda -
Z)*(1-poisscdf(Z,lambda)))) + Cb * (lambda * poisspdf(Z,lambda) + (lambda -
Z)*(1-poisscdf(Z,lambda))); ;

ezplot (Fs,[0,100])

xlabel('Base Stock Level');
ylabel('Expected Total Cost, E(TC)');
grid;
```

I3 MATLAB Program Optimizing SS/SP/TKCS

```
%This program uses the equations in Nori and Sarker (1998), equations (10), (18) and (19), specifically, to find the
%optimal no. of kanbans of a single stage manufacturing process.
%It comprises of 3 parts:
% 1st part: Find the optimal no. of kanbans for the case of lambda < miu
% 2nd part: Find the optimal no. of kanbans for the case of lambda == miu
% 3rd part: Plot the no. of kanbans vs. the E(TC) equation
%---------------------------------------------------------------------------------------------------------------------------------------------%

%Declare variables used
syms p Cb Ch k1 k2 S1 S2 positive

%Use bank format i.e. 2 dec places
format bank

%Request input from user
lambda = input ( '\n\nEnter demand arrival rate (lambda): ' );
miu = input ( 'Enter MP processing rate (miu): ' );

if (lambda > miu)
    fprintf ( 'The demand arrival rate cannot exceed the MP processing rate! \nPlease try again...');
    break;
end

if (lambda <=0) || (miu <=0)
    fprintf ( 'The demand arrival rate or MP processing rate you entered is invalid! \nPlease try again...');
    break;
end

p = miu / lambda;
Cb = input ( 'Enter Backorder Costs per unit time (Cb): ' );
Ch = input ( 'Enter Holding Costs per unit (Ch): ' );

if (Cb < Ch)
    fprintf ( 'The Backorder Costs cannot be lesser than the Holding Costs! \nPlease try again...');
    break;
end
```

```
if (Cb <=0) || (Ch <=0)
    fprintf ( 'The Backorder Cost or Holding Cost you entered is invalid!
\nPlease try again...');
    break;
end

%------------------------------------------------------------------------------
------------------------------------------------------------------------------
-%
%1st part: Find the no. of kanbans for the case of lambda < miu first
%------------------------------------------------------------------------------
------------------------------------------------------------------------------
-%

if lambda < miu
    %Solve for k1
    S1 = solve((p^(k1+1)) - k1*(p-1) - p - ((p-1)^2)*(Cb/Ch), k1);
    a = max (floor(double (S1))) ; %a = maximum number of kanbans and round
down a to the lowest integer

    fprintf ('\nThe max no. of kanbans is: ')
    disp(a);



    %Solve for k2
    S2 = solve ((p^(k2+1)) - k2*(p-1)/p - ((p-1)^2)*(Cb/(p*Ch)) - (2*p-1)/p,
k2);
    b = max (ceil(double (S2))); %b = minimum number of kanbans and round
up b to the highest integer

    fprintf ('\nThe min no. of kanbans is: ')
    disp(b);



    %Solve for the optimal no. of kanbans
    if a == b
        fprintf ( '\nThe optimal number of kanbans is:%6.0f', a);

        ETC = ( ((p-1)/((p^(a+1))-1))*Cb + p*Ch* (((a*(p-1)*(p^a))-((p^a)-
1))/(((p^(a+1))-1)*(p-1))) ); %Calculate Expected Total Cost (ETC)
        fprintf ( '\nThe corresponding expected total cost is: $%1.00f \n',
ETC);
```

```
    else
        LTC = ( ((p-1)/((p^(b+1))-1))*Cb + p*Ch* (((b*(p-1)*(p^b))-((p^b)-1))/(((p^(b+1))-1)*(p-1))) );
        %Initialize Lowest Total Cost (LTC) corresponding to min no. of kanbans (b)

        Ln = 1; %Initialize Lowest number (Ln) of kanbans to 1

        disp( 'No. of kanbans    Expected Total Cost (ETC)');

        for n = b:a
            ETC = ( ((p-1)/((p^(n+1))-1))*Cb + p*Ch* (((n*(p-1)*(p^n))-((p^n)-1))/(((p^(n+1))-1)*(p-1))) ); %Calculate Expected Total Cost (ETC)

            disp ([    n                   ETC]); %Display n and ETC

            if ETC < LTC
                LTC = ETC; %Assign Total Cost (TC) to Lowest Total Cost (LTC)
                Ln = n;   %Assign current number of kanbans (n) to Lowest number (Ln) of kanbans
            end
        end

        fprintf ( 'The optimal k is: %6.0f', Ln); %Display the optimal number of kanbans
        fprintf ( '\nThe corresponding Expected Total Cost (ETC) is: $%1.00f \n', LTC); %Display the Expected Lowest Total Cost (LTC)

    end


%-----------------------------------------------------------------------------------------------------------------------------------------%
%2nd part: Find the no. of kanbans for the case of lambda == miu
%-----------------------------------------------------------------------------------------------------------------------------------------%

elseif lambda == miu
    k3 = (((1+8*(Cb/Ch))^(1/2))-1)/2;

    fprintf ('\nThe max no. of kanbans is: ');
    disp(ceil(k3));
```

```
    fprintf ('\nThe min no. of kanbans is: ');
    disp(1);

    LTC1 = (Cb/2) + (Ch/2);  %Initialize Lowest Total Cost 1 (LTC1)
corresponding to min no. of kanbans of k =1, using Nori & Sarker Eqn (12)


    Ln1 = 1; %Initialize Lowest number (Ln) of kanbans to 1

        disp( 'No. of kanbans    Expected Total Cost (ETC)');

        for i = 1:k3
            ETC1 = Cb/(i+1) + (i*Ch)/2; %Calculate Expected Total Cost 1
(ETC1) when lambda == miu

            disp ([    i                   ETC1]); %Display i and ETC1

            if ETC1 < LTC1
                LTC1 = ETC1; %Assign Expected Total Cost 1 (ETC1) to Lowest
Total Cost 1 (LTC1)
                Ln1 = i;   %Assign current number of kanbans (i) to Lowest
number 1 (Ln1) of kanbans
            end
        end

        fprintf ( 'The optimal k is: %6.0f', Ln1); %Display the optimal
number of kanbans
        fprintf ( '\nThe corresponding Expected Total Cost (ETC) is:
$%1.00f \n', LTC1); %Display the Expected Lowest Total Cost (LTC)


end


%----------------------------------------------------------------------------
----------------------------------------------------------------------------
-%
%3rd part: Plot the no. of kanbans vs. the E(TC) equation
%----------------------------------------------------------------------------
----------------------------------------------------------------------------
-%

Fk = @(Y) ((p-1)/((p^(Y+1))-1))*Cb + p*Ch* (((Y*(p-1)*(p^Y))-((p^Y)-
```

```
1))/(((p^(Y+1))-1)*(p-1))) ;

ezplot (Fk,[0,50])

xlabel('Number of Kanbans');
ylabel('Expected Total Cost, E(TC)');
grid;
```

I4 MATLAB Program Optimizing SS/MP/TKCS

```
%This program finds the optimal no. of kanbans for a Single Stage -
%Two Product - TKCS.
%It comprises of 3 parts:
% 1st part: Gather inputs from the user. Namely:
%   - lambda1: demand arrival rate of product 1
%   - lambda2: demand arrival rate of product 2
%   - miu1: MP Processing Rate for product 1
%   - miu2: MP Processing Rate for product 2

% 2nd part: The program will compute the following
%   - lambdaA: Average or total demand arrival rate
%   - miuA: Average MP Processing rate
%   - L: Total number of jobs in process in system (or total WIP for all
%        product types)
%   - p: Proportion of parts processed belonging to particular product type
%        i.e. miuA / L

% 3rd part: Find the Optimal K* for each product type (i = 1, 2)
%   - Ch1: Holding Cost for product 1 ($/unit)
%   - Ch2: Holding Cost for product 2 ($/unit)
%   - Cs1: Shortage Cost for product 1 ($/time)
%   - Cs2: Shortage Cost for product 2 ($/time)
%   - By incrementing K, we obtain the Lowest Expected Total Cost (ETC)
when
%     ETC(K+1) - ETC (K) >=0 (for each product type, i = 1, 2)

%--------------------------------------------------------------------%

%Declare variables used
syms lambda1 lambda2 miu1 miu2 positive
syms Ch1 Ch2 Cs1 Cs2 positive
syms lambdaA miuA L p k1 k2 positive

%Use bank format i.e. 2 dec places
format bank

%1st part: Request input from user----------------------------------------%
lambda1 = input ( '\nEnter demand arrival rate (lambda) for Part 1: ' );
lambda2 = input ( '\nEnter demand arrival rate (lambda) for Part 2: ' );

lambdaA = lambda1 + lambda2;
```

```
fprintf ('\nThe Average Arrival Rate of 2 Independent Arrivals is %6.0f',
lambdaA);


%Request input from user for MP Processing Rates
miu1 = input ( '\nEnter MP processing rate (miu) for Part 1: ' );
miu2 = input ( '\nEnter MP processing rate (miu) for Part 2: ' );

if (lambda1 > miu1)
    fprintf ( 'The demand arrival rate cannot exceed the MP processing rate!
(For Part 1)\n Please try again...');
    break;
end

if (lambda2 > miu2)
    fprintf ( 'The demand arrival rate cannot exceed the MP processing rate!
(For Part 2)\n Please try again...');
    break;
end

if (lambda1 > lambda2)
    fprintf ( 'Part 1 demand arrival rate cannot exceed Part 2 demand
arrival rate! \n Please try again...');
    break;
end

if (lambda2 > miu1)
    fprintf ( 'Part 2 demand arrival rate cannot exceed Part 1 processing
rate! \n Please try again...');
    break;
end

if (lambda1 <=0 || lambda2 <=0 || miu1 <=0 || miu2 <=0)
    fprintf ( 'The demand arrival rate or MP processing rate you entered is
invalid! \nPlease try again...');
    break;
end

miuA = lambdaA*miu1*miu2 / (lambda1*miu2 + lambda2*miu1) ;

fprintf ('\nThe Average MP Processing Rate for Both Parts is: %6.0f\n',
miuA);

if (lambdaA > miuA)
```

```
    fprintf ('The Average Demand Arrival Rate has exceeded the Average MP
Processing Rate!\nPlease try again...');
    break;
end


%2nd part: Compute L and p------------------------------------------------%
L = (lambdaA^2/miuA^2)/(1-lambdaA/miuA) + (lambdaA/miuA);

p = miuA / L;

%3rd part: Compute ETC equation for each product type---------------------%
Ch1 = input ( 'Enter Holding Costs per unit (Ch)for Product 1: ' );
Cs1 = input ( 'Enter Shortage Costs per unit time (Cs) for Product 1: ' );

Ch2 = input ( 'Enter Holding Costs per unit (Ch)for Product 2: ' );
Cs2 = input ( 'Enter Shortage Costs per unit time (Cs) for Product 2: ' );

if (Cs1 < Ch1) || (Cs2 < Ch2)
    fprintf ( 'The Shortage Costs cannot be lesser than the Holding Costs!
\nPlease try again...');
    break;
end

if (Cs1 <=0) || (Ch1 <=0) || (Cs2 <=0) || (Ch2 <=0)
    fprintf ( 'The Shortage Cost or Holding Cost you entered is invalid!
\nPlease try again...');
    break;
end

%Obtain Optimal K1* first-------------------------------------------------%
k1 = input ( 'Enter range of search for Optimal No. of Kanbans for Product
1, K1*: ' );

disp( 'No. of kanbans   Expected Total Cost (ETC1)');
disp( 'for Product 1    for Product 1');

LTC1 = 1000000;

for n1 = 1:k1
    for x1 = 1:n1;
        %Step 1: Obtain ETC1 denominator (b1)
        b1 = 0;
```

```
        b1 = b1+factorial(n1)/factorial(n1-x1)*(lambda1^(n1-x1))*(p^x1);

        %Step 2: Obtain ETC1 numerator (a1)
        a1 = 0;
        a1 = a1+x1*(factorial(n1)/factorial(n1-x1)*(lambda1^(n1-
x1))*(p^x1));
    end

    %Step 3: Obtain ETC1
    ETC1 = (Cs1*(lambda1^n1)+Ch1*a1)/b1;


    disp ([n1             ETC1]); %Display n1 and ETC1

    if ETC1 < LTC1
        LTC1 = ETC1; %Assign Total Cost (TC) to Lowest Total Cost (LTC)
        Ln1 = n1;   %Assign current number of kanbans (n) to Lowest number
(Ln) of kanbans
    end
end

fprintf ( 'The optimal K1* is: %6.0f', Ln1); %Display the optimal number of
kanbans
fprintf ( '\nThe corresponding Expected Total Cost (ETC1) is: $%1.00f \n',
LTC1); %Display the Expected Lowest Total Cost (LTC)



%Now Obtain Optimal K2* ---------------------------------------------------%
k2 = input ( 'Enter range of search for Optimal No. of Kanbans for Product
2, K2*: ' );

disp( 'No. of kanbans   Expected Total Cost (ETC2)');
disp( 'for Product 2    for Product 2');

LTC2 = 1000000;

for n2 = 1:k2
    for x2 = 1:n2;
        %Step 1: Obtain ETC2 denominator (b2)
        b2 = 0;
        b2 = b2+factorial(n2)/factorial(n2-x2)*(lambda2^(n2-x2))*(p^x2);
```

```
        %Step 2: Obtain ETC2 numerator (a2)
        a2 = 0;
        a2 = a2+x2*(factorial(n2)/factorial(n2-x2)*(lambda2^(n2-
x2))*(p^x2));
    end

    %Step 3: Obtain ETC2
    ETC2 = (Cs2*(lambda2^n2)+Ch2*a2)/b2;



    disp ([n2             ETC2]); %Display n2 and ETC2

    if ETC2 < LTC2
        LTC2 = ETC2; %Assign Total Cost (TC) to Lowest Total Cost (LTC)
        Ln2 = n2;   %Assign current number of kanbans (n) to Lowest number
(Ln) of kanbans
    end
end

fprintf ( 'The optimal K2* is: %6.0f', Ln2); %Display the optimal number of
kanbans
fprintf ( '\nThe corresponding Expected Total Cost (ETC2) is: $%1.00f \n',
LTC2); %Display the Expected Lowest Total Cost (LTC)
```

I5 MATLAB Program Optimizing SS/MP/De/EKCS

```
%This program finds the optimal no. of kanbans for a Single Stage -
%Two Product - EKCS (Dedicated).
%It comprises of 3 parts:
% 1st part: Gather inputs from the user. Namely:
%    - lambda1: demand arrival rate of product 1
%    - lambda2: demand arrival rate of product 2
%    - miu1: MP Processing Rate for product 1
%    - miu2: MP Processing Rate for product 2

% 2nd part: The program will compute the following
%    - lambdaA: Average or total demand arrival rate
%    - miuA: Average MP Processing rate
%    - L: Total number of jobs in process in system (or total WIP for all
%         product types)
%    - p: Proportion of parts processed belonging to particular product type
%         i.e. miuA / L

% 3rd part: Compute lowest E{tstockout} - Expected Shortage Time
%    - Let Z1 = S1+K1 i.e. the Total WIP allowed
%    - Increment Z1 for E{tstockout} equation until lowest E{tstockout} is
%      obtained; hence Optimal Z1 obtained
%    - Do likewise for Optimal Z2

% 4th part: Compute lowest ETC
%    - Ch1: Holding Cost for product 1 ($/unit)
%    - Ch2: Holding Cost for product 2 ($/unit)
%    - Cs1: Shortage Cost for product 1 ($/time)
%    - Cs2: Shortage Cost for product 2 ($/time)
%    - Since Z1 = S1+K1, try different combinations of S1 and K1 that adds
up to
%      the optimal Z1
%    - With each combination, substitute into ETC1 equation
%    - Combination that gives Lowest ETC1 is Optimal S1* and K1*
%    - So likewise for lowest ETC2 and obtain Optimal S2* and K2*

%---------------------------------------------------------------------%

%Declare variables used
syms lambda1 lambda2 miu1 miu2 positive
syms Ch1 Ch2 Cs1 Cs2 positive
syms lambdaA miuA L p k1 k2 positive
```

```
%Use bank format i.e. 2 dec places
format bank

%1st part: Request input from user----------------------------------------%
lambda1 = input ( '\nEnter demand arrival rate (lambda) for Part 1: ' );
lambda2 = input ( '\nEnter demand arrival rate (lambda) for Part 2: ' );

lambdaA = lambda1 + lambda2;
fprintf ('\nThe Average Arrival Rate of 2 Independent Arrivals is %6.0f',
lambdaA);


%Request input from user for MP Processing Rates
miu1 = input ( '\nEnter MP processing rate (miu) for Part 1: ' );
miu2 = input ( '\nEnter MP processing rate (miu) for Part 2: ' );

if (lambda1 > miu1)
    fprintf ( 'The demand arrival rate cannot exceed the MP processing rate!
(For Part 1)\n Please try again...');
    break;
end

if (lambda2 > miu2)
    fprintf ( 'The demand arrival rate cannot exceed the MP processing rate!
(For Part 2)\n Please try again...');
    break;
end

if (lambda1 > lambda2)
    fprintf ( 'Part 1 demand arrival rate cannot exceed Part 2 demand
arrival rate! \n Please try again...');
    break;
end

if (miu1 > miu2)
    fprintf ( 'Part 1 processing rate cannot exceed Part 2 processing rate!
\n Please try again...');
    break;
end

if (lambda2 > miu1)
    fprintf ( 'Part 2 demand arrival rate cannot exceed Part 1 processing
rate! \n Please try again...');
```

```
    break;
end

if (lambda1 <=0 || lambda2 <=0 || miu1 <=0 || miu2 <=0)
    fprintf ( 'The demand arrival rate or MP processing rate you entered is
invalid! \nPlease try again...');
    break;
end

miuA = lambdaA*miu1*miu2 / (lambda1*miu2 + lambda2*miu1) ;

fprintf ('\nThe Average MP Processing Rate for Both Parts is: %6.0f\n',
miuA);

if (lambdaA > miuA)
    fprintf ('The Average Demand Arrival Rate has exceeded the Average MP
Processing Rate!\nPlease try again...');
    break;
end


%2nd part: Compute L and p-----------------------------------------------%
L = (lambdaA^2/miuA^2)/(1-lambdaA/miuA) + (lambdaA/miuA);

p = miuA / L;


% 3rd part: Compute lowest E{tstockout} - Expected Shortage Time---------%
% Obtain Optimal Z1* first-----------------------------------------------%

Z1 = input ( 'Enter range of search for Optimal No. of WIP for Product 1,
Z1*: ' );

disp( 'Total No. of WIP           E{tstockout}');
disp( 'for Product 1 (Z1)         for Product 1');

b1 = 0;

for n1 = 1:Z1
    for x1 = 0:n1;
        %Step 1: Obtain EST1 denominator (b1)
        b1 = b1+factorial(n1)/factorial(n1-x1)*(lambda1^(n1-x1))*(p^x1);
    end
```

```
    %Step 2: Obtain EST1
    EST1 = (lambda1^n1)/b1;

    disp ([n1              EST1]); %Display n1 and EST1

end

Z1star = input ( '\nThe optimal Z1* is: ' );
LST1 = input ('\nThe corresponding Expected Shortage Time (EST1) is: \n')


% Now obtain Optimal Z2* -------------------------------------------------%

Z2 = input ( 'Enter range of search for Optimal No. of WIP for Product 2,
Z2*: ' );

disp( 'Total No. of WIP           E{tstockout}');
disp( 'for Product 2 (Z2)         for Product 2');

b2 = 0;

for n2 = 1:Z2
    for x2 = 0:n2;
        %Step 1: Obtain EST2 denominator (b2)
        b2 = b2+factorial(n2)/factorial(n2-x2)*(lambda2^(n2-x2))*(p^x2);
    end

    %Step 2: Obtain EST2
    EST2 = (lambda2^n2)/b2;

    disp ([n2              EST2]); %Display n2 and EST2

end

Z2star = input ( '\nThe optimal Z2* is: ' );
LST2 = input ('\nThe corresponding Expected Shortage Time (EST2) is: \n')


% 4th part: Compute lowest ETC--------------------------------------------%
Ch1 = input ( 'Enter Holding Costs per unit (Ch)for Product 1: ' );
Cs1 = input ( 'Enter Shortage Costs per unit time (Cs) for Product 1: ' );
```

```
Ch2 = input ( 'Enter Holding Costs per unit (Ch)for Product 2: ' );
Cs2 = input ( 'Enter Shortage Costs per unit time (Cs) for Product 2: ' );

if (Cs1 < Ch1) || (Cs2 < Ch2)
    fprintf ( 'The Shortage Costs cannot be lesser than the Holding Costs!
\nPlease try again...');
    break;
end

if (Cs1 <=0) || (Ch1 <=0) || (Cs2 <=0) || (Ch2 <=0)
    fprintf ( 'The Shortage Cost or Holding Cost you entered is invalid!
\nPlease try again...');
    break;
end

%First Find lowest ETC1--------------------------------------------------%

fprintf('S1      K1      Z1      ETC1(S1,K1)\n');
fprintf('----------------------------------');


LTC1 = 1e100000;

 for S1 = 0 : Z1star
     for K1 = 1 : Z1star

         ETC1 = Ch1*(S1+K1/2) + Cs1*LST1;

         if (S1+K1) <= Z1star

             fprintf('\n%d      %d      %d         $%2.2f', S1, K1, (S1+K1),
ETC1);

             if ETC1 < LTC1
                 LTC1 = ETC1;
                 V1 = S1;
                 W1 = K1;
             end
         end
     end
 end
```

```
 fprintf('\nThe optimal Base Stock for Product 1, S1* is %d', V1);
 fprintf('\nThe optimal No. Of Kanbans for Product 1, K1* is %d', W1);
 fprintf('\nThe Lowest Expected Total Cost for Product 1, ETC1(S1*,K1*) is
$%2.2f\n', LTC1);


%Now find lowest ETC2--------------------------------------------------%

fprintf('S2      K2      Z2      ETC2(S2,K2)\n');
fprintf('----------------------------------');


LTC2 = 1e100000;

 for S2 = 0 : Z2star
     for K2 = 1 : Z2star

         ETC2 = Ch2*(S2+K2/2) + Cs2*LST2;

         if (S2+K2) <= Z2star

             fprintf('\n%d      %d      %d         $%2.2f', S2, K2, (S2+K2),
ETC2);

             if ETC2 < LTC2
                 LTC2 = ETC2;
                 V2 = S2;
                 W2 = K2;
             end
         end
     end
 end

 fprintf('\nThe optimal Base Stock for Product 2, S2* is %d', V2);
 fprintf('\nThe optimal No. Of Kanbans for Product 2, K2* is %d', W2);
 fprintf('\nThe Lowest Expected Total Cost for Product 2, ETC2(S2*,K2*) is
$%2.2f\n', LTC2);
```

I6 MATLAB Program Optimizing SS/MP/Sh/EKCS

```
%This program finds the optimal no. of kanbans for a Single Stage -
%Two Product - EKCS (SHARED).
%It comprises of 3 parts:
% 1st part: Gather inputs from the user. Namely:
%    - lambda1: demand arrival rate of product 1
%    - lambda2: demand arrival rate of product 2
%    - miu1: MP Processing Rate for product 1
%    - miu2: MP Processing Rate for product 2

% 2nd part: The program will compute the following
%    - lambdaA: Average or total demand arrival rate
%    - miuA: Average MP Processing rate
%    - L: Total number of jobs in process in system (or total WIP for
all
%         product types)
%    - p: Proportion of parts processed belonging to particular
product type
%         i.e. miuA / L

%    - p1: Utilization rate for product 1 i.e. lambda1 / miuA
%    - p2: Utilization rate for product 2 i.e. lambda2 / miuA

% 3rd part: Now proportion the total WIP to product 1 and 2
%    - Let Z1 = S1+K1 i.e. the Total WIP allowed
%    - Z1 = p1/(p1+p2)*L
%    - Z2 = p2/(p1+p2)*L

% 4th part: Compute E{tstockout} - Expected Shortage Time for each
product
% type using Z1 and Z2.

% 5th part: Compute lowest ETC
%    - Ch1: Holding Cost for product 1 ($/unit)
%    - Ch2: Holding Cost for product 2 ($/unit)
%    - Cs1: Shortage Cost for product 1 ($/time)
%    - Cs2: Shortage Cost for product 2 ($/time)
%    - Since Z1 = S1+K1, try different combinations of S1 and K1 that
adds up to
%      the optimal Z1
%    - With each combination, substitute into ETC1 equation
%    - Combination that gives Lowest ETC1 is Optimal S1* and K1*
%    - So likewise for lowest ETC2 and obtain Optimal S2* and K2*
```

```
% 6th part: Obtain Shared K*
%   - K* = K1* + K2*

%---------------------------------------------------------------------%

%Declare variables used
syms lambda1 lambda2 miu1 miu2 positive
syms Ch1 Ch2 Cs1 Cs2 positive
syms lambdaA miuA L p k1 k2 positive

%Use bank format i.e. 2 dec places
format bank

%1st part: Request input from user-----------------------------------
----%
lambda1 = input ( '\nEnter demand arrival rate (lambda) for Part 1:
' );
lambda2 = input ( '\nEnter demand arrival rate (lambda) for Part 2:
' );

lambdaA = lambda1 + lambda2;
fprintf ('\nThe Average Arrival Rate of 2 Independent Arrivals
is %6.0f', lambdaA);


%Request input from user for MP Processing Rates
miu1 = input ( '\nEnter MP processing rate (miu) for Part 1: ' );
miu2 = input ( '\nEnter MP processing rate (miu) for Part 2: ' );

if (lambda1 > miu1)
    fprintf ( 'The demand arrival rate cannot exceed the MP
processing rate! (For Part 1)\n Please try again...');
    break;
end

if (lambda2 > miu2)
    fprintf ( 'The demand arrival rate cannot exceed the MP
processing rate! (For Part 2)\n Please try again...');
    break;
end

if (lambda1 > lambda2)
    fprintf ( 'Part 1 demand arrival rate cannot exceed Part 2 demand
```

```
arrival rate! \n Please try again...');
    break;
end

if (miu1 > miu2)
    fprintf ( 'Part 1 processing rate cannot exceed Part 2 processing
rate! \n Please try again...');
    break;
end

if (lambda2 > miu1)
    fprintf ( 'Part 2 demand arrival rate cannot exceed Part 1
processing rate! \n Please try again...');
    break;
end

if (lambda1 <=0 || lambda2 <=0 || miu1 <=0 || miu2 <=0)
    fprintf ( 'The demand arrival rate or MP processing rate you
entered is invalid! \nPlease try again...');
    break;
end

miuA = lambdaA*miu1*miu2 / (lambda1*miu2 + lambda2*miu1) ;

fprintf ('\nThe Average MP Processing Rate for Both Parts
is: %6.0f\n', miuA);

if (lambdaA > miuA)
    fprintf ('The Average Demand Arrival Rate has exceeded the
Average MP Processing Rate!\nPlease try again...');
    break;
end



%2nd part: Compute L, p, p1 and p2-----------------------------------
-----%
L = (lambdaA^2/miuA^2)/(1-lambdaA/miuA) + (lambdaA/miuA);

p = miuA / L;

p1 = lambda1 / miuA;

p2 = lambda2 / miuA;
```

```
% 3rd part: Now proportion the total WIP to product 1 and 2
% Let Z1 = S1+K1 i.e. the Total WIP allowed

Z1 = ceil (p1/(p1+p2)*L);
Z2 = ceil (p2/(p1+p2)*L);


% 4th part: Compute E{tstockout} - Expected Shortage Time using
optimal Z1 and Z2---------%
% Obtain E{tstockout} for product 1 first----------------------------
---------------------%

% Step 1: Obtain EST1 denominator (b1)

b1 = 0;

for x1 = 0:Z1;
    b1 = b1+factorial(Z1)/factorial(Z1-x1)*(lambda1^(Z1-x1))*(p^x1);
end

% Step 2: Obtain EST1
EST1 = (lambda1^Z1)/b1;

fprintf ('\nThe Optimal Total WIP Allowed for Product 1 (S1+K1)*
is: %6.0f\n', Z1);
fprintf ('\nThe Expected Shortage Time for Product 1 is: %6.0f\n',
EST1);


% Now Obtain E{tstockout} for product 2------------------------------
-------------------%
% Step 1: Obtain EST2 denominator (b2)

b2 = 0;

for x2 = 0:Z2;
    b2 = b2+factorial(Z2)/factorial(Z2-x2)*(lambda2^(Z2-x2))*(p^x2);
end

% Step 2: Obtain EST2
EST2 = (lambda2^Z2)/b2;
```

```
fprintf ('\nThe Optimal Total WIP Allowed for Product 2 (S2+K2)*
is: %6.0f\n', Z2);
fprintf ('\nThe Expected Shortage Time for Product 2 is: %6.0f\n',
EST2);



% 5th part: Compute lowest ETC---------------------------------------
----%
Ch1 = input ( 'Enter Holding Costs per unit (Ch)for Product 1: ' );
Cs1 = input ( 'Enter Shortage Costs per unit time (Cs) for Product 1:
' );

Ch2 = input ( 'Enter Holding Costs per unit (Ch)for Product 2: ' );
Cs2 = input ( 'Enter Shortage Costs per unit time (Cs) for Product 2:
' );

if (Cs1 < Ch1) || (Cs2 < Ch2)
    fprintf ( 'The Shortage Costs cannot be lesser than the Holding
Costs! \nPlease try again...');
    break;
end

if (Cs1 <=0) || (Ch1 <=0) || (Cs2 <=0) || (Ch2 <=0)
    fprintf ( 'The Shortage Cost or Holding Cost you entered is
invalid! \nPlease try again...');
    break;
end

%First Find lowest ETC1----------------------------------------------
---%

fprintf('S1     K1     Z1     ETC1(S1,K1)\n');
fprintf('----------------------------------');


LTC1 = 1e100000;

 for S1 = 1 : Z1
     for K1 = 0 : Z1

         ETC1 = Ch1*(S1+K1/2) + Cs1*EST1;

         if (S1+K1) == Z1
```

```
            fprintf('\n%d     %d     %d         $%2.2f', S1, K1,
(S1+K1), ETC1);

            if ETC1 < LTC1
                LTC1 = ETC1;
                V1 = S1;
                W1 = K1;
            end
        end
    end
end

fprintf('\nThe optimal Base Stock for Product 1, S1* is %d', V1);
fprintf('\nThe optimal No. Of Kanbans for Product 1, K1* is %d', W1);
fprintf('\nThe Lowest Expected Total Cost for Product 1,
ETC1(S1*,K1*) is $%2.2f\n', LTC1);


%Now find lowest ETC2-----------------------------------------------
--%

fprintf('S2     K2     Z2     ETC2(S2,K2)\n');
fprintf('----------------------------------');


LTC2 = 1e100000;

for S2 = 1 : Z2
    for K2 = 1 : Z2

        ETC2 = Ch2*(S2+K2/2) + Cs2*EST2;

        if (S2+K2) == Z2

            fprintf('\n%d     %d     %d         $%2.2f', S2, K2,
(S2+K2), ETC2);

            if ETC2 < LTC2
                LTC2 = ETC2;
                V2 = S2;
                W2 = K2;
            end
```

```
        end
    end
end

fprintf('\nThe optimal Base Stock for Product 2, S2* is %d', V2);
fprintf('\nThe optimal No. Of Kanbans for Product 2, K2* is %d', W2);
fprintf('\nThe Lowest Expected Total Cost for Product 2,
ETC2(S2*,K2*) is $%2.2f\n', LTC2)
```

APPENDIX J

ARENA Snapshots

J1 ARENA Snapshots for SS/SP/KCS

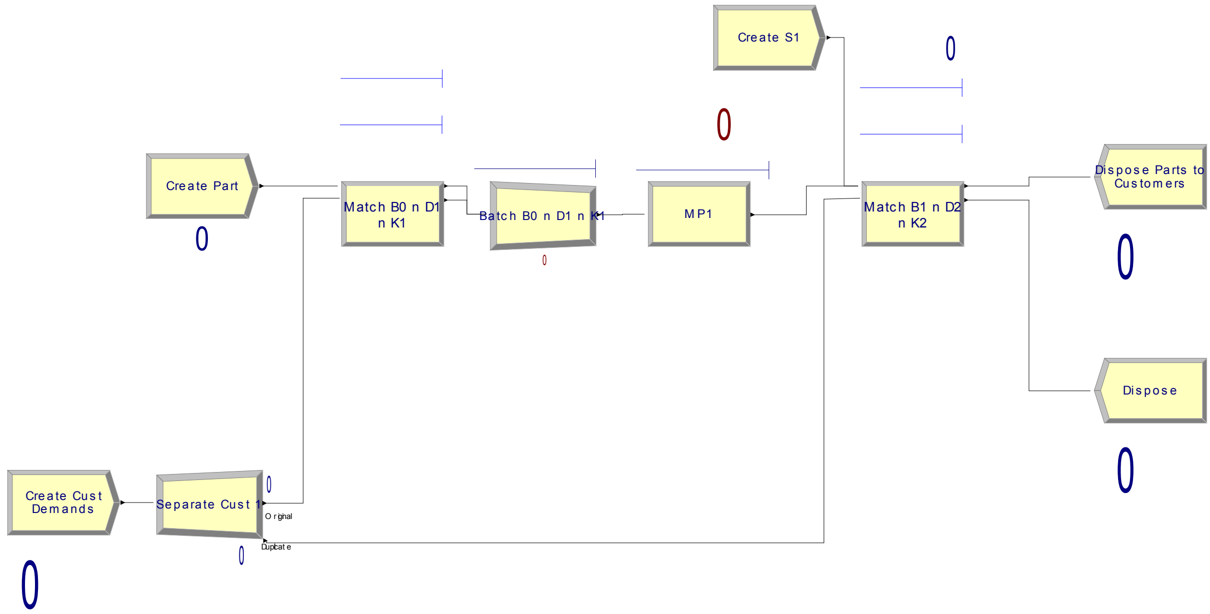

Figure J 1: ARENA Snapshot of SS/SP/BS

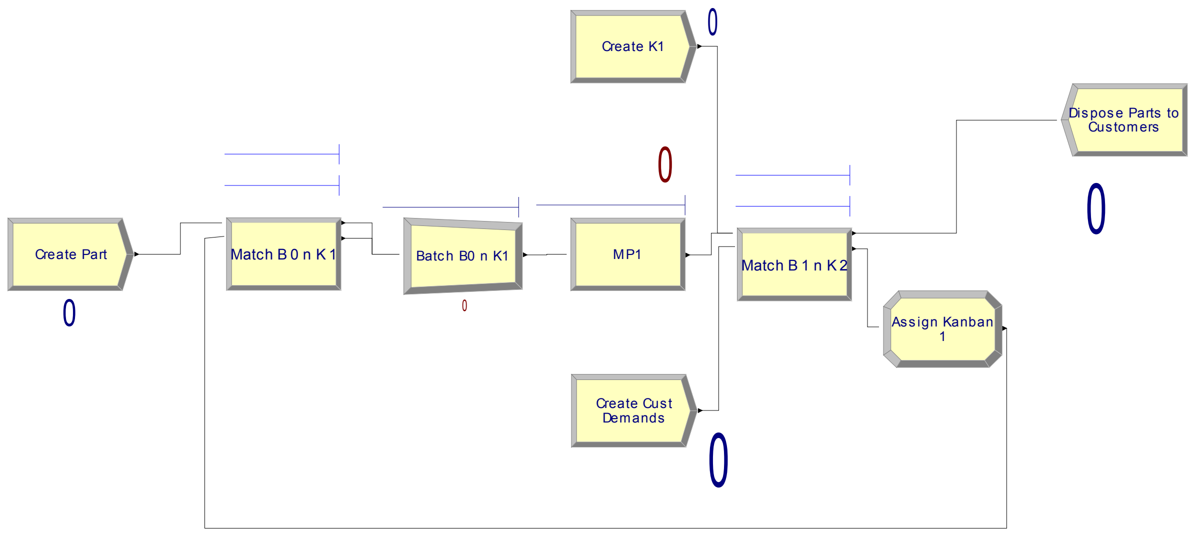

Figure J 2: ARENA Snapshot of SS/SP/TKCS

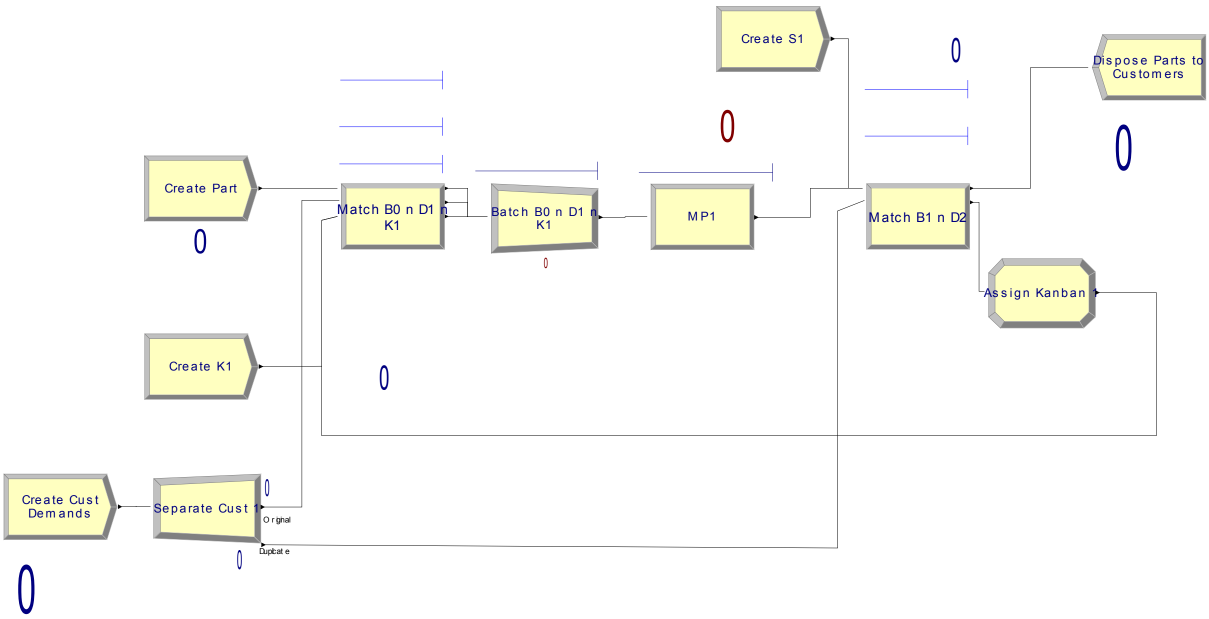

Figure J 3: ARENA Snapshot of SS/SP/EKCS

J2 ARENA Snapshots for SS/MP/KCS

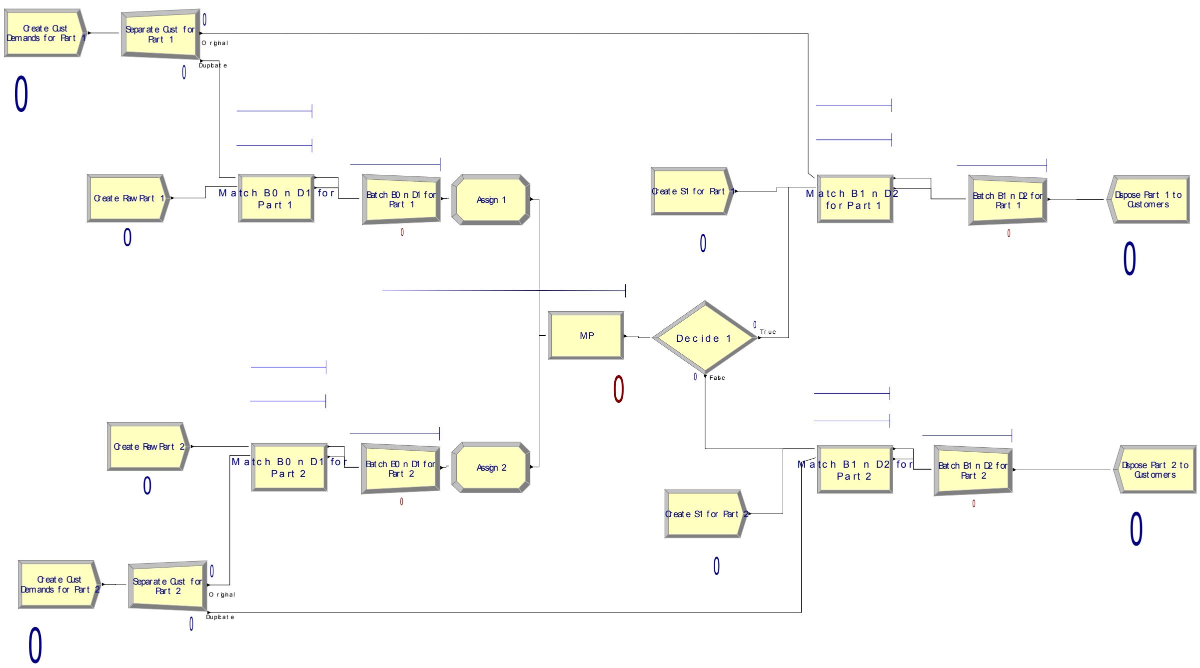

Figure J 4: Single Stage, Multiple Product Base Stock (SS/MP/BS) in ARENA

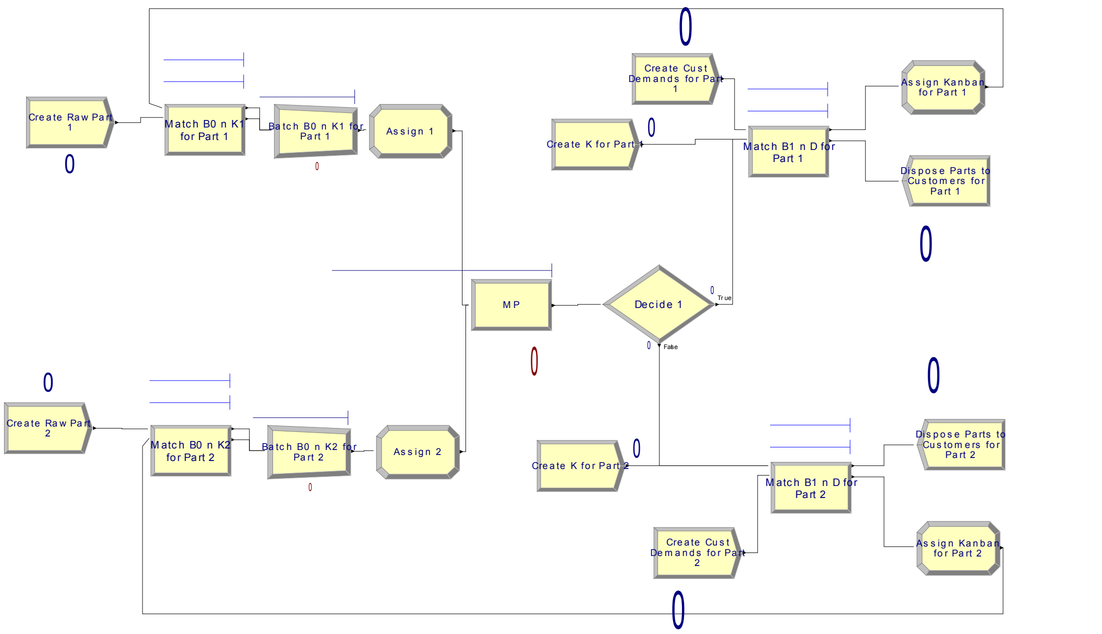

Figure J 5: Single Stage, Multiple Product Traditional Kanban Control System (SS/MP/TKCS) in ARENA

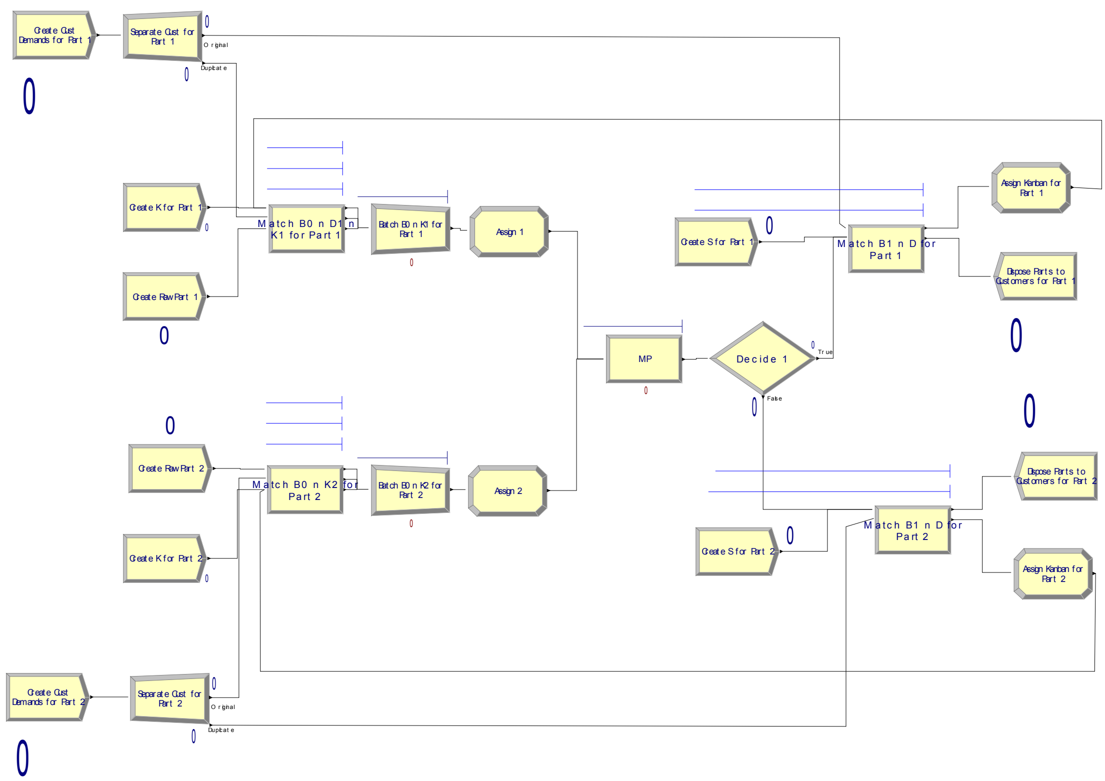

Figure J 6: Single Stage, Multiple Product, Dedicated Extended Kanban Control System (SS/MP/De-EKCS) in ARENA

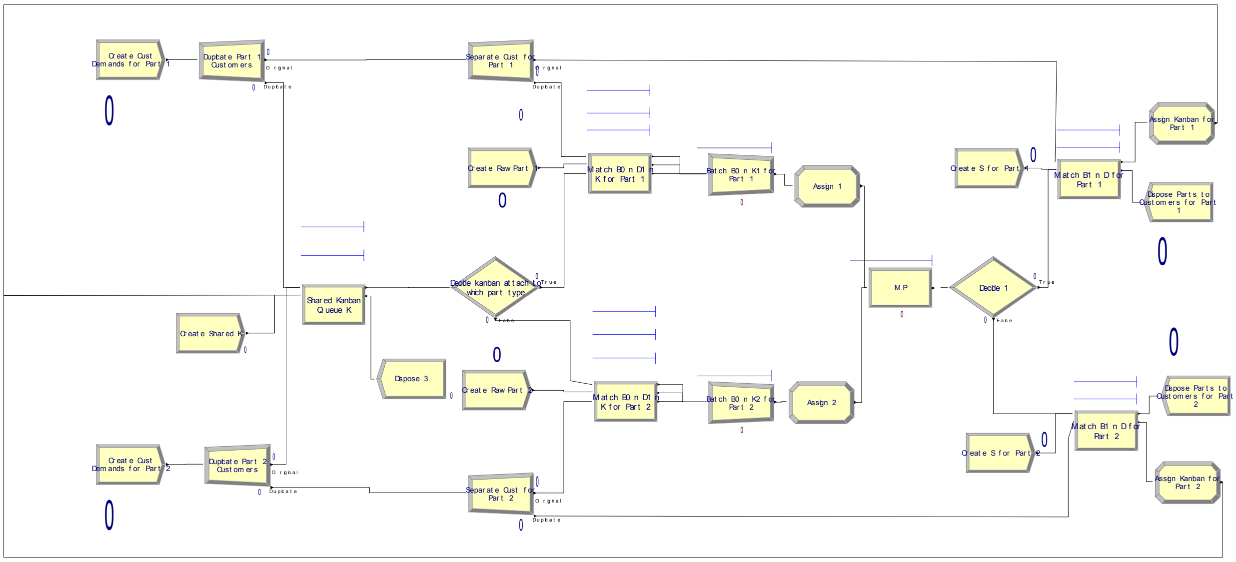

Figure J 7: Single Stage, Multiple Product, Shared Extended Kanban Control System (SS/MP/Sh-EKCS) in ARENA

APPENDIX K

Hypothesis Tests for Simulation Results

K1 Brief Conclusion of Hypothesis Tests

29 out of 30 hypothesis tests conducted on the Single Product Kanban Controlled System (SP/KCS) showed that EKCS does outperform TKCS and BS.

20 out of 20 hypothesis tests conducted on the Multiple Product Kanban Controlled System (MP/KCS) showed that EKCS does outperform TKCS and BS.

Hence, the claim that EKCS outperforms TKCS and BS is true.

K2 Steps Involved in Hypothesis Tests

Single Product Kanban Controlled System (SP/KCS)

There are three main scenarios for the simulation experiments: Low, Medium and High Backorder and Shortage Costs.

For each of these scenarios, the Actual Total Cost (ATC) was plotted against the Demand Arrival Rate (units per day). Since the comparisons done here were using the ATC mean values; but not their standard deviations, hypothesis tests were done to confirm the results. Hence, for each scenario, and for each demand arrival rate, a hypothesis test was carried out. We follow Lind, Marchal, and Wathen (2011) method of comparing population means with unknown population standard deviations (Pooled t-test).These are the assumptions:

1. The samples are independent
2. The two populations follow the normal distribution
3. The population standard deviations are unknown (thus we use the t distribution rather than the z)

Step 1: Taking Samples

For example, we take a specific case to explain. For the Low Backorder and Shortage Cost scenario, and for a demand arrival rate of 10 units per day, the following ten samples were taken:

Table K 1: Ten Samples taken for a Particular Scenario: Low Backorder and Shortage Cost, Demand Arrival Rate of 10 units per day

	Actual Total Cost (ATC) $		
Sample Number.	Extended Kanban Control System (EKCS)	Traditional Kanban Control System (TKCS)	Base Stock (BS)
1	1.91	11.17	107.89
2	2.31	12.12	76.49
3	2.14	11.85	100.68
4	1.96	11.91	98.73
5	2.30	12.19	102.03
6	2.00	11.50	100.03
7	2.09	11.20	102.94
8	2.51	9.87	90.45
9	2.47	10.57	106.78
10	1.69	11.11	108.21

Step 2: Stating the Claim

Since there are three systems but we can only do one comparison per time, we have to make two claims here

$$H_0 : \mu_{EKCS} \leq \mu_{TKCS}$$
$$H_1 : \mu_{EKCS} > \mu_{TKCS}$$

Where μ_{EKCS} : refers to mean of EKCS's Actual Total Cost (ATC)
μ_{TKCS} : refers to mean of TKCS's Actual Total Cost (ATC)

$$H_3 : \mu_{EKCS} \leq \mu_{BS}$$
$$H_4 : \mu_{EKCS} > \mu_{BS}$$

Where μ_{EKCS} : refers to mean of EKCS's Actual Total Cost (ATC)
μ_{BS} : refers to mean of BS's Actual Total Cost (ATC)

This means that if H_0 is accepted, EKCS's ATC is lower than TKCS. And if H_3 is accepted, it means that EKCS's ATC is lower than BS.

Step 3: Selecting Level of Significance

We choose a significance level of 0.05, or rather, α=0.05. In other words, if H_0 and H_3 are accepted, the claim that EKCS does outperform TKCS and BS is proven at a 95% confidence interval (this is using the t distribution for a one-tailed test).

According to Lind et al. (2011), the p-value gives the probability of observing a sample value as extreme as, or more extreme than, the value observed, given that the null hypothesis is true. A p-value is frequently compared to the significance level to evaluate the decision regarding the null hypothesis. It is a means of reporting the likelihood that H_0 is true.

If the p-value is greater than the significance level, then H_0 is not rejected. But if the p-value is less than the significance level, then H_0 is rejected.

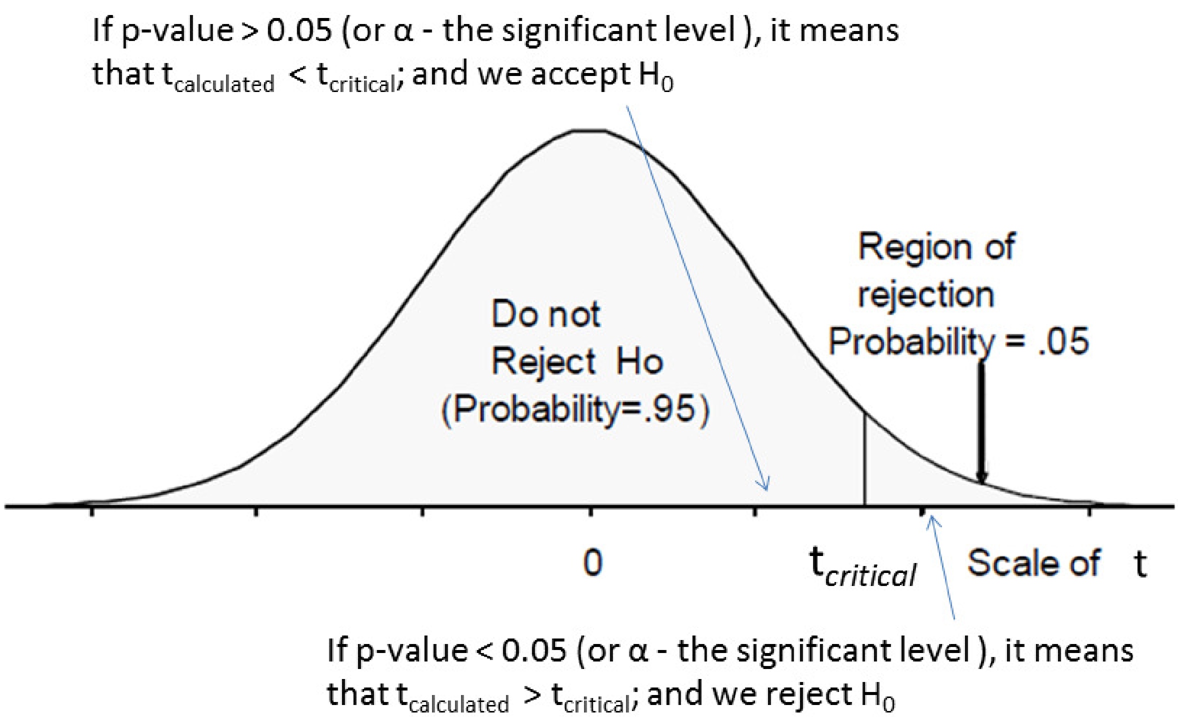

Figure K 1: Rejection region of H_0 (Lind et al., 2011)

Step 4: Perform a Two-Sample pooled t-test for Difference of Two Means using Statistical Software JMP.

We enter the above sampled data into JMP

	Actual Total Cost (ATC - $)	Type of System
1	2.13	EKCS
2	2.05	EKCS
3	1.92	EKCS
4	2.25	EKCS
5	2.01	EKCS
6	1.99	EKCS
7	1.82	EKCS
8	2	EKCS
9	2.3	EKCS
10	2.1	EKCS
11	11.3	TKCS
12	10.9	TKCS
13	10.7	TKCS
14	11.2	TKCS
15	11.7	TKCS
16	11.7	TKCS
17	11.9	TKCS
18	11.8	TKCS
19	9.41	TKCS
20	12.3	TKCS

Figure K 2: Data Entry into JMP Data Table

Then we do an Analyse > Fit Y by X

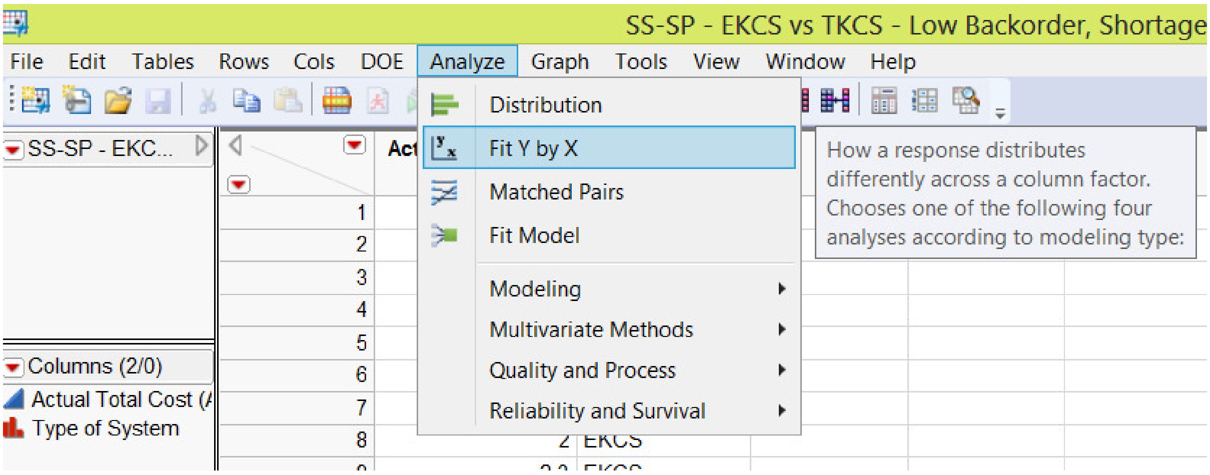

Figure K 3: Using "Fit Y by X" method on JMP

Then we drag "Actual Total Cost" to Y, Response and "Type of System" to X, Factor

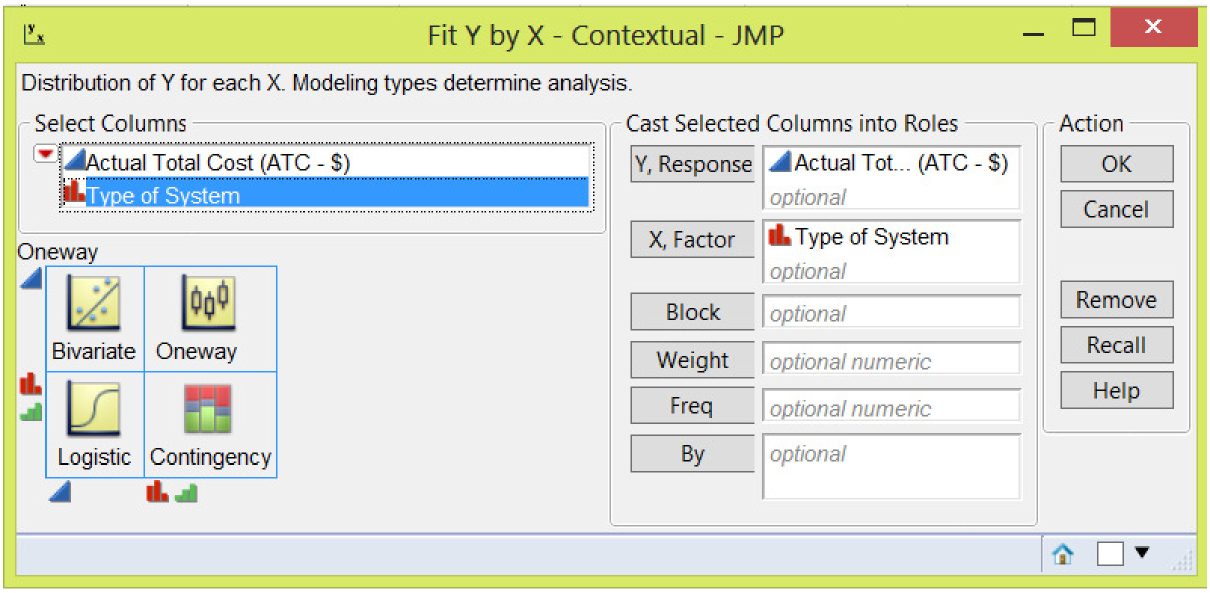

Figure K 4: Filling up the Y and X axis for JMP

After doing a t-test, we manage to obtain the following output:

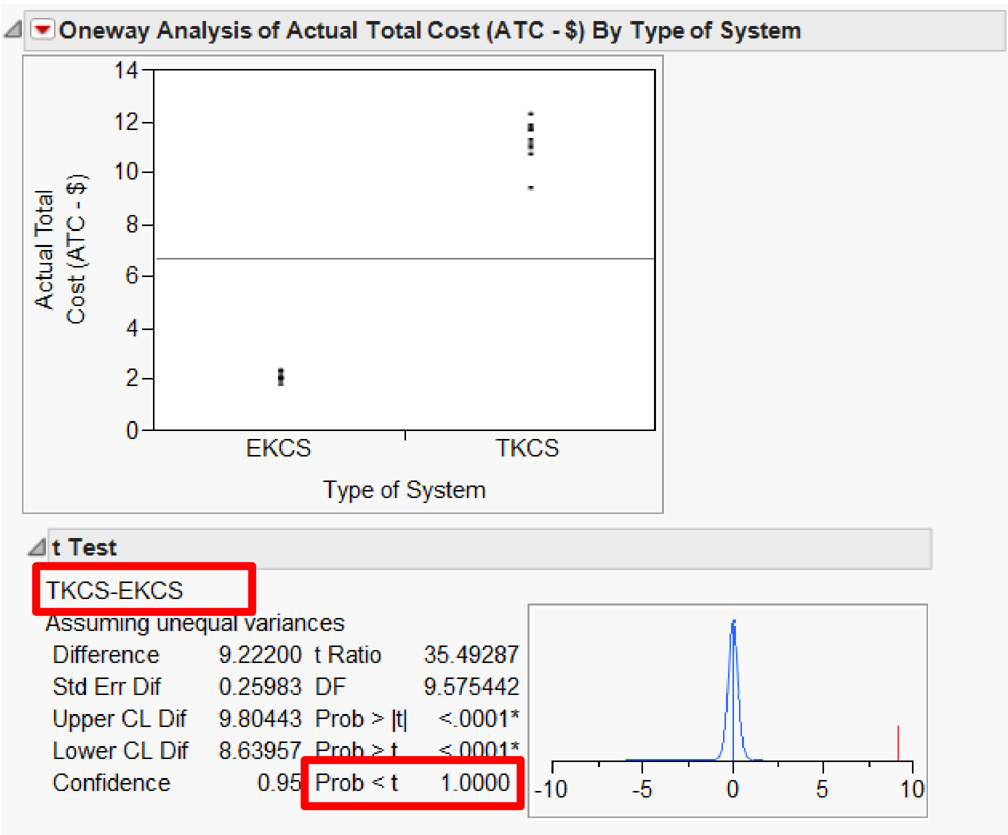

Figure K 5:JMP Output showing a Data Plot and t test of the Data Samples

Since we are investigating if mean ATC for EKCS is less than TKCS, we are performing a left tailed test. As such, the respective p-value is 1. As p-value >0.05 (or α), we accept H_0 and reject H_1. Therefore, we are 90% confident that EKCS outperforms TKCS since its ATC is lower.

The following steps above are repeated for comparing EKCS versus BS. And its p-value is also 1, which is >0.05 (or α), we accept H_0 and reject H_1. Therefore, we are 95% confident that EKCS outperforms BS since its ATC is lower.

K3 Results of Hypothesis Test

Single Product Kanban Controlled System (SP/KCS)

Thirty hypothesis tests were conducted at 95% Confidence Interval for SP/KCS systems. Ten tests were conducted for each for the three scenarios: Low, Medium and High Backorder and Shortage Cost. Overall, in almost all cases, it showed that EKCS does outperform TKCS and BS; except for a few unique cases.

Referring to Table K2 below, there are ten hypothesis tests conducted for Low Backorder, Shortage Cost scenario. Each of it showed a p-value of over 0.05 (or α); which means that the claim of EKCS Actual Total Cost (ATC) is lower than TKCS and BS is true at a 95% confidence level.

Table K 2: Results of Hypothesis Test for SS/SP/KCS: Low Backorder and Shortage Cost Scenario

Demand Arrival Rate (lambda) (units per day)	EKCS vs. TKCS p-value	EKCS vs. BS p-value	Conclusion at 95% Confidence Interval
10	1	1	EKCS outperforms TKCS and BS
12	0.3143	1	EKCS outperforms TKCS and BS
14	0.9721	1	EKCS outperforms TKCS and BS
16	0.462	1	EKCS outperforms TKCS and BS
18	0.6581	1	EKCS outperforms TKCS and BS

Referring to Table K3 below, there are ten hypothesis tests conducted for Medium Backorder, Shortage Cost scenario. Most of it showed a p-value of over 0.05 (or α);

which means that the claim of EKCS Actual Total Cost (ATC) is lower than TKCS and BS is true at a 95% confidence level.

Table K 3: Results of Hypothesis Test for SS/SP/KCS: Medium Backorder and Shortage Cost Scenario

Demand Arrival Rate (lambda) (units per day)	EKCS vs. TKCS p-value	EKCS vs. BS p-value	Conclusion at 95% Confidence Interval
10	0.5767	1	EKCS outperforms TKCS and BS
12	0.0515	1	EKCS outperforms TKCS and BS. But it only outperforms TKCS slightly since the p-value is very close to 0.05 (or alpha)
14	0.9794	1	EKCS outperforms TKCS and BS
16	0.0193	1	EKCS outperforms BS but not TKCS because its p-value is lower than alpha.
18	0.7666	1	EKCS outperforms TKCS and BS

We now examine the two special cases where EKCS does not outperform.

Case 1: For the scenario of Demand Arrival Rate of 12 units per day. We examine the JMP output:

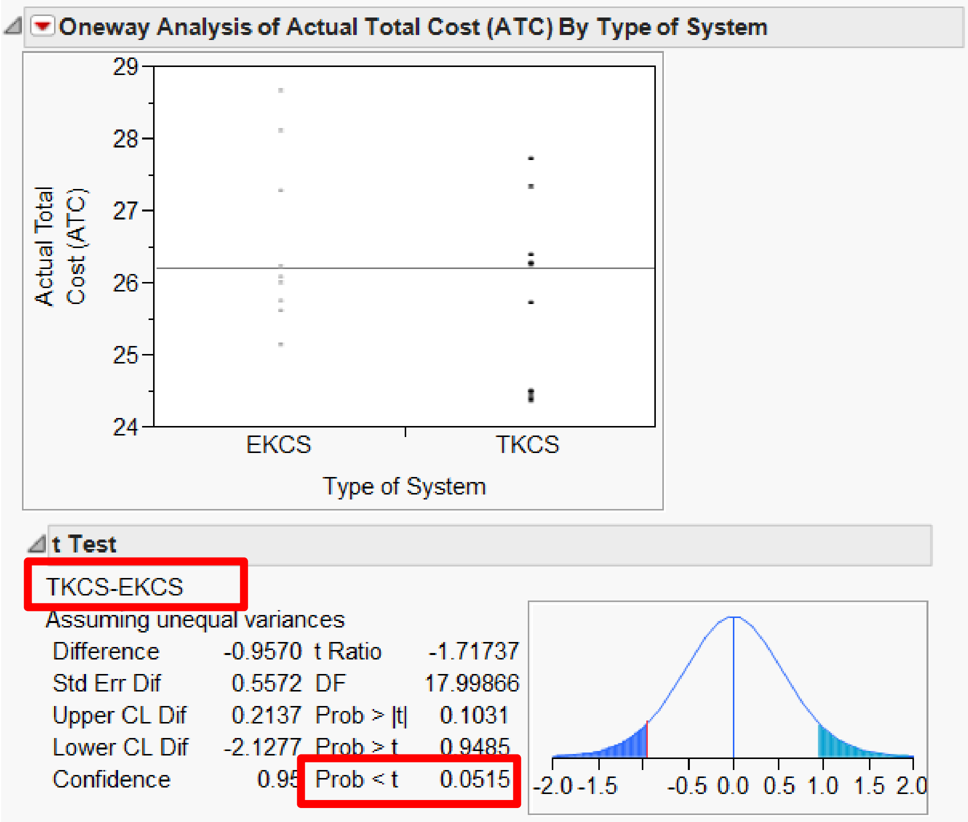

Figure K 6: Case 1: Where EKCS only outperforms TKCS by a little

Referring to Figure K6, looking at the data plot we see that the values are very close to one another. This explains why the p-value is only 0.0515, very close to α of 0.05.

Case 2: For the scenario of Demand Arrival Rate of 16 units per day. We examine the JMP output:

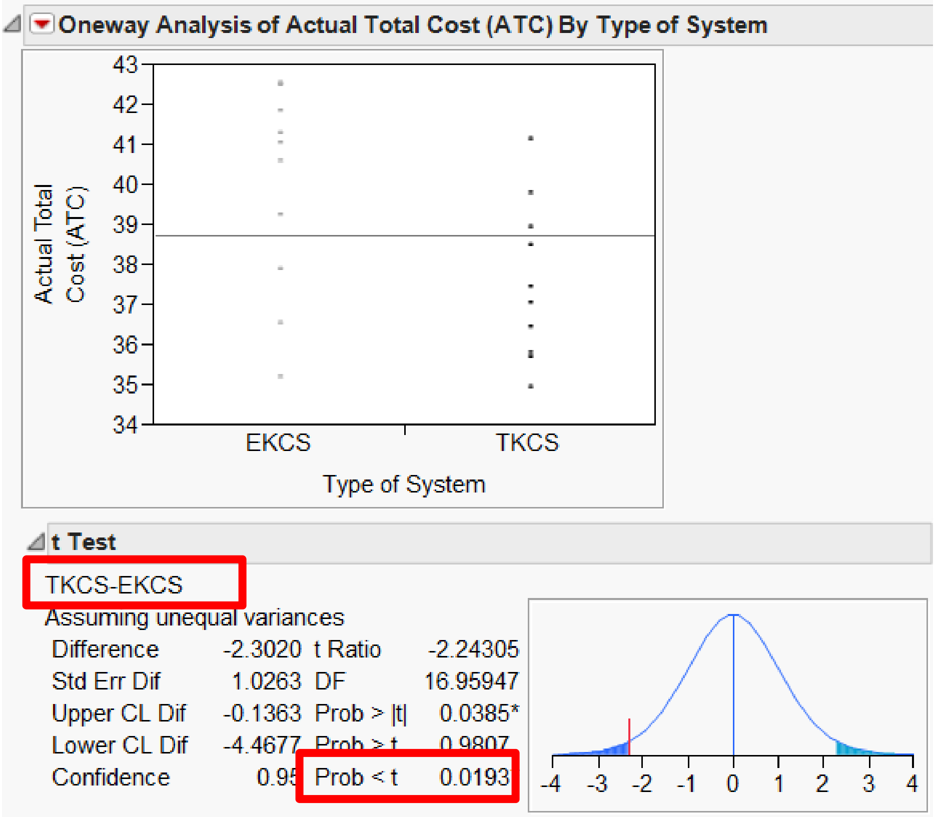

Figure K 7: **Case 2: Where EKCS does not outperform TKCS**

Referring to Figure K7, this is a rare case, and indeed the only case that the p-value (0.0193) has fallen below 0.05. Hence, only in this case we are 95% confident that EKCS does not outperform TKCS.

Referring to Table K4 below, there are ten hypothesis tests conducted for High Backorder, Shortage Cost scenario. All of them showed a p-value of over 0.05 (or α); which means that the claim of EKCS Actual Total Cost (ATC) is lower than TKCS and BS is true at a 95% confidence level.

Table K 4: **Results of Hypothesis Test for SS/SP/KCS: High Backorder and Shortage Cost Scenario**

Demand Arrival Rate (lambda) (units per day)	EKCS vs. TKCS p-value	EKCS vs. BS p-value	Conclusion at 95% Confidence Interval
10	0.9988	1	EKCS outperforms TKCS and BS
12	0.997	1	EKCS outperforms TKCS and BS

14	0.9031	1	EKCS outperforms TKCS and BS
16	0.1666	1	EKCS outperforms TKCS and BS
18	0.6768	0.2255	EKCS outperforms TKCS and BS

Multiple Product Kanban Controlled System (MP/KCS)

Twenty hypothesis tests were conducted at 95% Confidence Interval for MP/KCS systems. Ten tests were conducted for each for the two scenarios: Average MP Processing Rate of Product 1 smaller than Product 2 and vice versa. In all cases, it showed that Dedicated EKCS does outperform TKCS and BS. Since it has been proven that the Dedicated and Shared EKCS are equivalent, hypothesis tests were only run on the De-EKCS.

Referring to Table K5 below, there are ten hypothesis tests conducted for Average MP Processing Rate of Product 1 smaller than Product 2 scenario. Each of it showed a p-value of over 0.05 (or α); which means that the claim of De-EKCS Actual Total Cost (ATC) lower than TKCS and BS is true at a 95% confidence level.

Table K 5: Results of Hypothesis Test for SS/MP/KCS: Average MP Processing Rate of Product 1 smaller than Product 2

Demand Arrival Rate for Product 2 (lambda 2) (units per day)	EKCS vs. TKCS p-value	EKCS vs. BS p-value	Conclusion at 95% Confidence Interval
2	0.9971	1	EKCS outperforms TKCS and BS
3	0.9807	1	EKCS outperforms TKCS and BS
4	0.9999	1	EKCS outperforms TKCS and BS
5	1	1	EKCS outperforms TKCS and BS
6	1	1	EKCS outperforms TKCS and BS

Referring to Table K6 below, there are ten hypothesis tests conducted for Average MP Processing Rate of Product 1 greater than Product 2 scenario. Each of it showed a p-value of over 0.05 (or α); which means that the claim of De-EKCS Actual Total Cost (ATC) lower than TKCS and BS is true at a 95% confidence level.

Table K 6: Results of Hypothesis Test for SS/MP/KCS: Average MP Processing Rate of Product 1 greater than Product 2

Demand Arrival Rate for Product 2 (lambda 2) (units per day)	EKCS vs. TKCS p-value	EKCS vs. BS p-value	Conclusion at 95% Confidence Interval
2	1	1	EKCS outperforms TKCS and BS
3	1	1	EKCS outperforms TKCS and BS
4	1	1	EKCS outperforms TKCS and BS
5	1	1	EKCS outperforms TKCS and BS
6	1	1	EKCS outperforms TKCS and BS