International Journal of Production Management and Engineering

http://dx.doi.org/10.4995/ijpme.2015.3038
Received 2014-05-29 Accepted: 2014-10-22

A performance comparison of single product kanban control systems

Alvin Ang

Department of Systems and Engineering Management, Nanyang Technological University
50 Nanyang Avenue. Singapore 639798
Angw0038@ntu.edu.sg

Abstract: This paper presents a simulation experiment comparing the Single Stage, Single Product Base Stock (BS), Traditional Kanban Control System (TKCS) and Extended Kanban Control System (EKCS). The results showed that BS incurs the highest cost in all scenarios; while EKCS is found to be effective only in a very niche scenario. TKCS is still a very powerful factory management system to date; and EKCS did not perform exceptionally well. The only time EKCS did outperform TKCS was during low demand arrival rates and low Backorder (C_b) and Shortage costs (C_s). That is because during then, it holds no stock. The most important discovery made here is that EKCS becomes TKCS once it has base stock (or dispatched kanbans). The results have also evinced the strength of the pure kanban system, the TKCS over BS. Hence managers using BS should consider upgrading to TKCS to save cost.

Key words: Kanban Control System (KCS), Base Stock (BS), Extended Kanban Control System (EKCS), Markov Chain, Arena, Simulation.

1. Introduction

Kanban is a Japanese word for card. A Kanban Control System (KCS) is a production mechanism that uses kanbans, or production authorization cards, to control the Work-In-Process (WIP) of the production floor. Once a customer demand arrives, the kanban that was previously attached to the finished part is removed and sent back, upstream, to re-initiate the manufacturing production process. This happens simultaneously while the finished part is being shipped to the customer.

1.1. Traditional Kanban Control System (TKCS)

The Traditional Kanban Control System (TKCS) was first proposed by Sugimori, Kusunoki, Cho, and Uchikawa (1977). Figure 1 shows a Single Stage production line controlled by the TKCS. By Single Stage, we mean that the Manufacturing Process (MP) contains only a Single Server. The number of kanbans limit the WIP. This system is the most famous pull mechanism in the world today (Monden, 1998). It limits the amount of inventory at each stage, such that the maximum WIP is only equal to the number of kanbans circulating in that stage.

Referring to Figure 1, the TKCS operates as follows: When a customer demand arrives at the system it joins Customer Demand Queue D_1, requesting the release of a finished product from Output Buffer B_1 to the customer.

At that time there are two possibilities:

1. If a part is available in Output Buffer B_1 (which is initially the case), the finished is released to the customer. At the same time, the kanban that was attached to it will be detached. Then, this kanban is transferred upstream to the Undispatched Kanban Queue K_1, carrying with it a demand signal for the production of a new finished part.

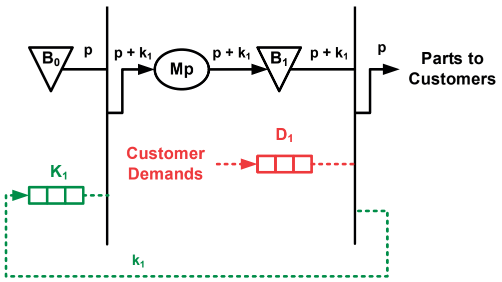

Figure 1. A Single Stage, Single Product Traditional Kanban Control System (SS/SP/TKCS) (source: Sugimori *et al.* (1977))

https://ojs.upv.es/index.php/IJPME

2. Otherwise, if no parts are available in Output Buffer B_1, the demand is backordered and the customer is left waiting in the Customer Demand Queue D_1; until such a time a new part is completed and placed in Output Buffer B_1. The newly finished part will be released to the customer instantly and the detached kanban will be transferred back to Kanban Queue K_1.

The Output Buffer B_0 represents the raw material inventory buffer, in which it is assumed to carry infinite stock, while the Manufacturing Process is denoted by MP. The advantage of the kanbans in TKCS is that it acts as a form of feedback, assisting 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 WIP for the entire production line. Lastly, this system does not respond well to long-term demand fluctuations (Monden, 1983).

1.2. Base Stock (BS)

Figure 2 shows the Base Stock (BS) System. It was first proposed by Clark and Scarf (1960). It does not limit WIP but limits inventory stored in the Output Buffer, B_1, with its Base Stock level, S. It does not use cards as feedback signals to previous stages, but uses instantaneous transmission of demand signals to all production stages. Queues D_i, where i = 1, 2, contain the customer demands.

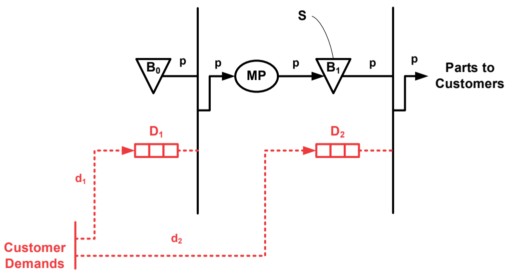

Figure 2. A Single Stage, Single Product Base Stock System (SS/SP/BS) (source: Clark and Scarf (1960)).

The Base Stock (BS) System operates as follows: When the system is in its initial state (before any demands arrive), Buffer B_1 contains S number of base stocks of finished products. Buffer B_0 is the components buffer and is assumed to contain an infinite quantity of components.

When a customer demand arrives at the system, it is replicated into multiple demand signals. These signals are immediately transmitted to its respective queues, Queues D_i, where i = 1, 2. The last demand signal joins Queue D_2, requesting the release of a finished product from Buffer B_2 to the customer.

At this point there can be two possibilities:

1. If a product is available in Output Buffer B_1, it is immediately released to the customer. Then, the previous demand signal in Customer Demand Queue D_1 will signal the MP to produce a new part to top up the Base Stock Level, S in Output Buffer B_1.
2. Otherwise, if no product is available in Output Buffer B_1, the demand is backordered and the customer waits in Queue D_2 until a new part is completed in the upstream stage.

One advantage of the Base Stock (BS) system is that it sets a target level of production in the Output Buffer B_1, by bounding it with a Base Stock level, S. That is, the MP will stop producing once the Output Buffer B_1 contains S parts. Another advantage of this system is that there is no demand information blockage, due to the assumption of instantaneous transmission of demands to all production stages.

One disadvantage of this system is that, though the output buffers are bounded by the base stock level, the WIP levels in each stage are unbounded. That is, this system does not set a limit on the WIP levels; neither in each stage nor for the entire production line. Thus, if 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 will continue to receive direct demand information and will operate to release parts as usual leading to an unbounded build-up inventory in front of the failed machine. Another disadvantage is that this system has no feedback, and hence there is no coordination between stages.

1.3. Extended Kanban Control System (EKCS)

Figure 3 shows the Extended Kanban Control System (EKCS). It is a hybrid of both the TKCS and BS.

In the initial state, the Output Buffer B_1 contains S amounts of finished parts, with a kanban attached to every part. And the raw material buffer, Buffer B_0 is assumed to contain infinite number of components.

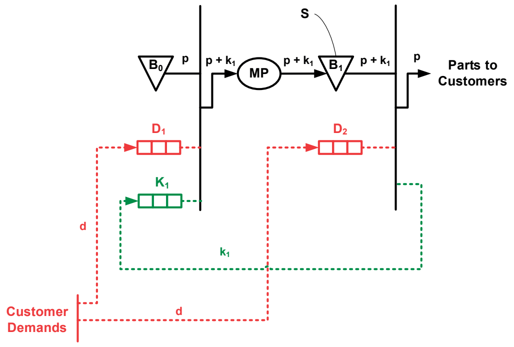

Figure 3. A Single Stage, Single Product Extended Kanban Control System (SS/SP/EKCS) (source: Dallery (2000))

The Customer Demand Queues, Queue D_i, i = 1, 2 are empty; while the Undispatched Kanban Queue, Queue K_1, contains K number of undispatched kanbans.

The EKCS operates as follows: When a customer demand arrives at the system, it is instantaneously split into multiple demands. The first demand joins Queue D_2, requesting the release of a finished product from Output Buffer B_1 to the customer.

At this point, there are two possibilities:

1. If a product is available in Output Buffer B_1, this product is released to the customer after the kanban is detached. Simultaneously, this un-detached kanban is transferred upstream to the Undispatched Kanban Queue, Queue K_1.The replicated demand joins Customer Demand Queue D_1. Since Input Buffer B_0 is assumed to contain infinite raw parts, the previous un-detached kanban in Queue K_1 is now attached to one raw part and sent into the MP for processing.
2. If there is no final product in Output Buffer B_1, the demand is backordered and has to wait in Queue D_2. However, the replicated demand that went into Queue D_1 may signal a raw part from Input Buffer B_0 into the MP for processing; presuming that there's an undispatched kanban lying in the Undispatched Kanban Queue, K_1.

One advantage of EKCS is that there is an instantaneous transmission of demand. When a demand arrives at the system, it is immediately broadcasted 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.

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).

2. Literature Review

2.1. Overview of Kanban Control Systems (KCS)

The TKCS was first proposed by Sugimori *et al.* (1977). BS was first proposed by Clark and Scarf (1960) and EKCS was first proposed by Dallery (2000). They have been described in the previous Section 1 Introduction. There are many other types of KCS and they will be briefly mentioned here.

CONWIP stands for CONstant Work–In–Process. This system was first proposed by Spearman, Woodruff, and Hopp (1990). 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 SS/TKCS is equivalent to a SS/CONWIP system. CONWIP control is executed only at the entry of the manufacturing system.

The Generalized Kanban Control System (GKCS) was first proposed by Buzacott (1989). GKCS sets a limit on WIP levels at each stage. Kanbans and demand signals are coupled together and both of them have to be present in an authorization queue before parts can move downstream. However, the presence of the authorization queue means 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 GKCS has not become popular (Boonlertvanich, 2005).

CONWIP Kanban (CK) was first proposed by Bonvik, Couch, and Gershwin (1997). It was proposed to leverage on the advantages of both the CONWIP and the TKCS. The advantage of CONWIP Kanban is that the WIP is controlled for the entire production line as well as the 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. However, having two types of cards also mean 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).

Extended CONWIP Kanban (ECK) was first proposed by (Boonlertvanich, 2005). It is a combination of BS, CONWIP, and TKCS, and is supposed to encapsulate the advantages of all three systems. The kanban is freed earlier in ECK than in other systems, since its detached right after the part leaves a 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).

The Flexible Kanban System (FKS) was first introduced by Gupta, Al-Turki, and Perry (1999). They introduced this new system to cope with uncertainties and planned/unplanned interruptions. They demonstrated FKS's superiority by conducting four case examples which covered various uncertainties. After comparing the FKS's performance with the traditional JIT system, their claim was 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 introduced a new adaptive kanban-type pull control mechanism which determines when to release or reorder raw parts based on customer demands. They claimed 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 WIP in the system. They showed that their adaptive system can outperform the traditional kanban pull control mechanism while remaining easy to implement.

The Reactive Kanban System (RKS) was proposed by 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 proposed 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 placed numerous detection points at each inventory buffer to detect these unstable changes. The performance of their RKS is finally analysed using simulation experiments.

The Knowledge Kanban (KK) was proposed by Jou Lin, Frank Chen, and Min Chen (2013). They proposed a 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 claimed that their proposed system helps employees of these R&D firms reduce the cycle time of their work. First, they created a Virtual Enterprise (VE). Then, they designed a KK model to custom fit it. Finally, in their study, they claimed that KK system they created is an effective tool to facilitate knowledge creation, storage, transmission and sharing for R&D firms.

The latest E-Kanban paper was documented by Al-Hawari and Aqlan (2012). In their paper, they developed a software application for an E-Kanban inventory control system; developed to track WIP inventory and finished goods in an aluminium factory. They claimed that after the current paper system is replaced with their E-Kanban system, data entry errors were minimized. Furthermore, their results showed that manufacturing lead time and WIP have been reduced by an average of 88% and 50%, respectively. They also built 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) studied a cellular manufacturing system controlled by kanban. In their model, they included the possibility of defective items and rework. They used a Mixed-

Integer Non Linear programming (MINLP) model to minimize total cost. Thereafter, their total cost model was used to determine the optimal number of kanbans and batch size. They also used Particle Swarm Optimization (PSO) and Simulated Annealing (SA) algorithms to lessen the large computational time for solving large MINLPs. They showed 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) studied how to synchronize the flow of materials in a kanban controlled serial production line. Their production line is described as a queuing network; which they then made 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 validated their results using the Pro Model discrete event simulator. They discovered that their performance measure had a very small error compared to their result. Thus, they claimed that their method was effective in synchronizing inventory with customer demands.

2.2. Comparison studies of Kanban Control Systems (KCS)

The true advantages of EKCS over BS and TKCS are still not properly addressed in the research literature. In this section, the literature that compares different types of KCS are reviewed.

Karaesmen and Dallery (2000) used an optimal control framework to study the BS, TKCS and GKCS. They used a two-stage production system where demands arrive according to a Poisson process with rate λ and their MP have exponentially distributed service times with rate μi (i=1,2). However, their modeling approach has made it difficult for the analyses of inventory levels in the two separate stages because they used X_1 as a random variable to represent a combination of stage 1 output buffer and stage 2 MP. Usually, in literature, X_1 should denote the WIP of the first MP plus the first output buffer while X_2 denote the WIP of the second MP plus the second output buffer. Also, they did not use EKCS in their comparison because under the state space representation approach, the EKCS is a special case of the GKCS. Hence, scenarios which EKCS outperforms BS and TKCS are not clearly highlighted. Moreover, they did not compare their performances in terms of Key Performance Indicators (KPIs).

The latest comparison of pull control policies was done by Korugan and Çadırcı (2008). They studied the four most common pull control systems: BS, TKCS, GKCS and EKCS, using a Markov Chain model to develop each of the four policies. These models were then analysed using a cost function, which was then minimized with respect to the control parameters of each control mechanism. Finally, results are obtained from numerical experiments and conclusions drawn. Even though the authors explicitly mentioned that the EKCS and GKCS displayed superior performance over the BS and TKCS, they did not show how this was done. Also, the pull models are not of standard tandem process lines. They included an additional remanufacturing process on top of the usual MP, which makes their analysis more complex.

Khuller (2006) used simulation to compare two types of kanban control systems in different manufacturing environments. Although KPIs such as Fill Rate, WIP and order fulfillment time were used as a gauge, he did not use the standard EKCS. Instead, he used the Extended Information Kanban Control System (EiKCS) i.e. EKCS with the Base Stock level equal to the maximum WIP capacity at each stage.

Deokar (2004) also used simulation to compare the TKCS, GKCS and EKCS. She assumed a multi-product system where the kanbans are either dedicated or shared. By assuming a multi-product system, she increased the complexity of the analysis. Even so, she has not specifically mentioned why, and in what scenarios, does the EKCS outperform the TKCS.

Only four references in the kanban literature exists for evaluating EKCS performance. Furthermore, these papers do not distinctly demonstrate how EKCS perform in different scenarios. This clearly shows that there is insufficient analysis on how the EKCS outperforms traditional systems like the BS and TKCS. Thus, in order for conventional factory managers to be convinced of EKCS's performance, a clear and well defined comparison in terms of KPIs is needed.

3. Kanban Controlled Systems Optimization Models

This section briefly discusses models to optimize BS, TKCS and EKCS, all of which also assume Single Stage and Single Product (SS/SP) for consistency. Having models to optimize these systems are important because a fair comparison can then take place. If non-optimal systems were compared, their performance results would be inaccurate.

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 the optimal Base Stock Level, 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)'s approach.

Many authors have proposed different techniques, mostly based on Markov Chains, to find the optimal kanban number, 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 devised 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.

Ang and Piplani (2010) have proposed a model for obtaining Optimal Base Stock, S*, and Optimal Number of Kanbans, K*, in a SS/SP/EKCS. In this research, our method will be used for optimizing the EKCS. Likewise, a Matlab program for this model has been written.

4. Simulation Experiment

4.1. Arena Simulation Model

TKCS, BS and EKCS were simulated in Arena Version 12. Figures I1 to I3 in Appendix I shows their snapshots respectively. These simulation models were developed based on their respective KCS schematics. Thus, SS/SP/BS (Figure I1) corresponds to Figure 1; SS/SP/TKCS (Figure I2) corresponds to Figure 2 and SS/SP/EKCS (Figure I3) corresponds to Figure 3. 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 Section 3, coded in Matlab and simulations were run using the parameters described in Section 4.2. These steps are repeated for SS/SP/TKCS and EKCS.

4.2. Simulation Parameters

Simulation experiments are conducted under three scenarios (Table 1): low, medium and high backorder and shortage costs, C_b and C_s. The holding cost, C_h, is kept constant. MP rate is also constant, but the demand arrival rate is varied.

Before each scenario is simulated, Matlab was 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 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,

Table 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	

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,

Actual Total Cost (ATC) = Total Backorder Cost + Total Shortage Cost + Total Holding Cost (1)

Backorder Cost, C_b, can be defined as a penalty cost, in dollars per unit, for unmet demand at the end of each period. Shortage Cost, C_s, in dollars per unit day, is defined as the penalty cost for demand kept waiting. For example, if average customer waiting time, as given by Arena, is 20 hours, it means that on average, a customer order is required to wait for 20 hours before the demand is met. This translates into a Shortage Cost of $(20/24×$C_s$), as the system is in shortage mode for 20 hours. Finally, Holding Cost, C_h, can be defined as cost (dollars per unit per day) of holding inventory in the output buffer.

4.3. Simulation Assumptions

Assumptions made in modelling the SS/SP/KCS were:

1. All systems produced only a single product.
2. One card kanban system was adopted.
3. The system produced no defective parts.
4. All systems had a single stage containing only one MP.
5. Each MP contained only one machine/server.
6. Machine setup times were zero.
7. No machine failures could occur.
8. Each machine could only process one part per unit time.
9. Parts were transported with negligible transfer time.
10. Demand signals and kanbans flowed instantaneously.
11. Parts followed a First In First Out (FIFO) dispatching policy at all machines and buffers.
12. Input material buffers had an infinite supply of component parts.
13. Demand arrivals followed a Poisson process.
14. All processing times at MPs were assumed to be exponentially distributed.
15. Each replication was run for one year.
16. Each simulation run was replicated 10 times.
17. The warm up period for each replication was three months.

The reason for using a three month warm up period was to eradicate any transient behaviour of the systems. Since each replication was chosen to run for one year, we discarded the results for the first quarter, selecting the steady-state results only at the beginning of the second quarter.

5. Results and Discussion

Figures 7 to 9 show the ATC of each system with varying demand arrival rates. 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.

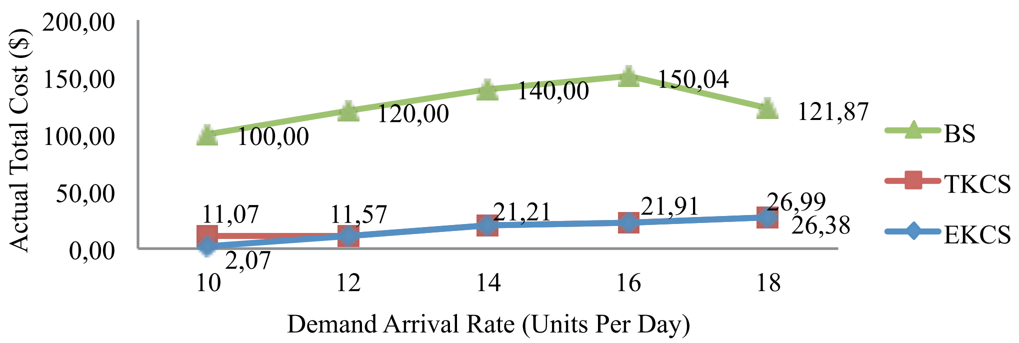

Figure 4. Comparison of SS/SP/EKCS, TKCS and BS in low backorder and shortage costs scenario

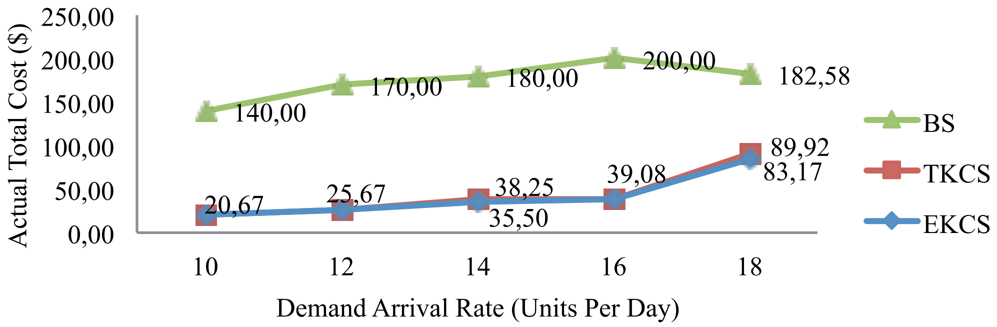

Figure 5. Comparison of SS/SP/EKCS, TKCS and BS in medium backorder and shortage costs scenario.

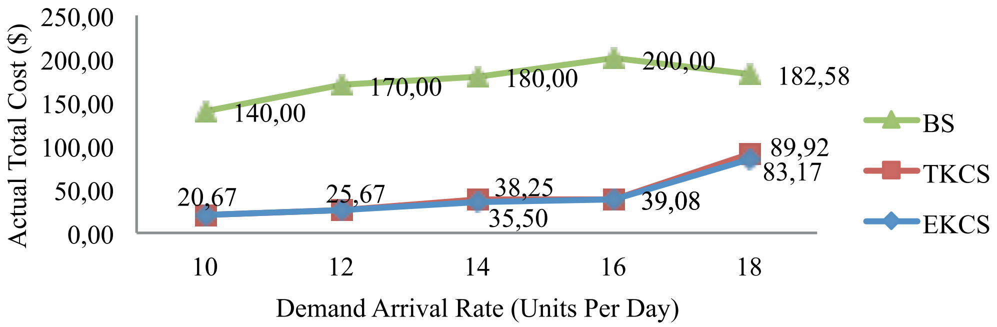

Figure 6. Comparison of SS/SP/EKCS, TKCS and BS in medium backorder and shortage costs scenario.

Referring to Figures 7 to 9, 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 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.

5.1. EKCS and TKCS Performance Similarity

Looking at Figures 4 to 6, 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. Both their average inventory levels seem to hold almost equal amount of inventory; hence, their average backorders and customer waiting times are almost equal. This leads to their 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.

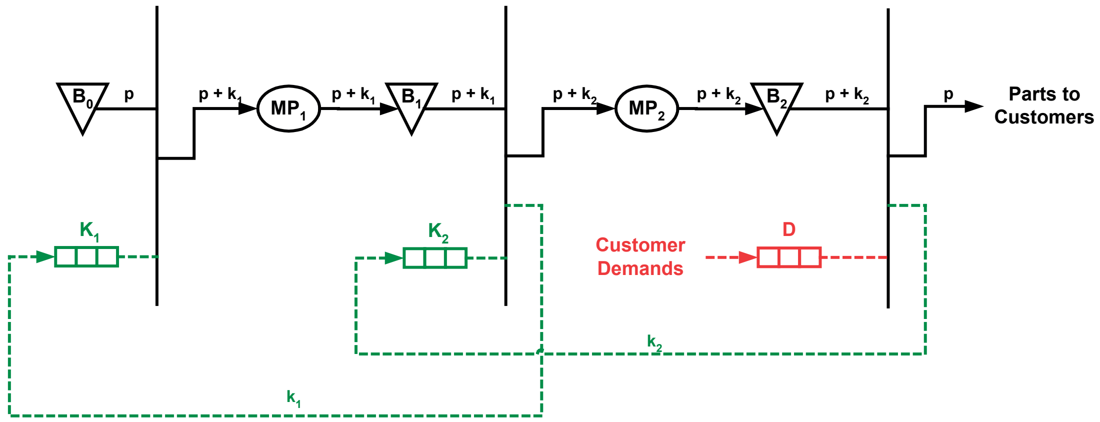

Figure 7. Two Stage, Single Product TKCS.

5.2. 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*.

5.3. 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 3, 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 1 and 3, 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

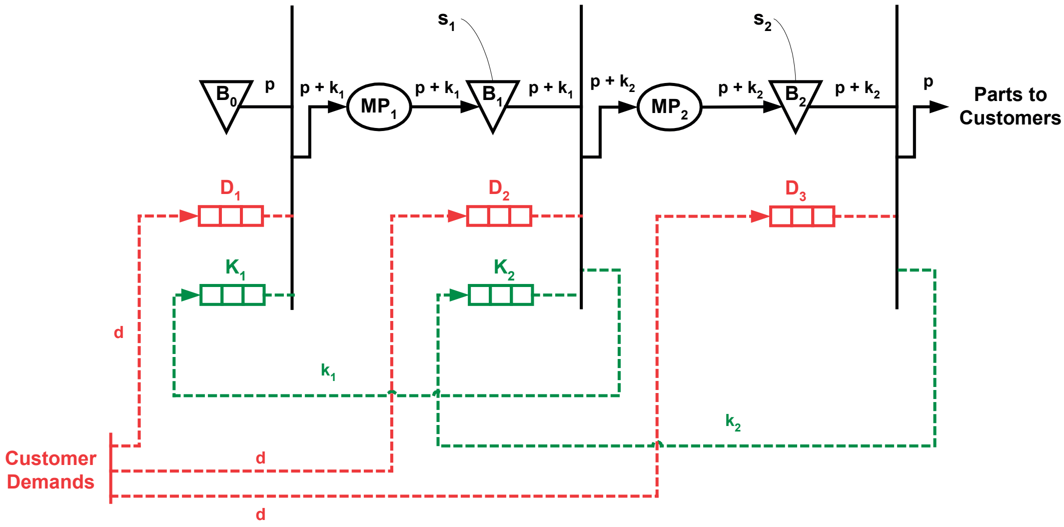

Figure 8. Two Stage, Single Product EKCS.

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.

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

In a Two Stage SP/TKCS (Figure 7), it 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 and instantaneous transmission of kanbans), even though demand arrives only at the final stage. This makes the role of instantaneous transmission of demands in a Two Stage SP/EKCS (Figure 8) redundant. That is, demand queues D_1 and D_2 are unnecessary. This is because 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 instantaneous demand queues are rendered unnecessary.

5.5. Comparison of TKCS and EKCS in Low Utilization Scenario

This section analyses scenarios under which EKCS may outperform TKCS. Further investigation (Figures 9 to 11) 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 9, the cost difference between EKCS and TKCS is quite significant and worthy of further study.

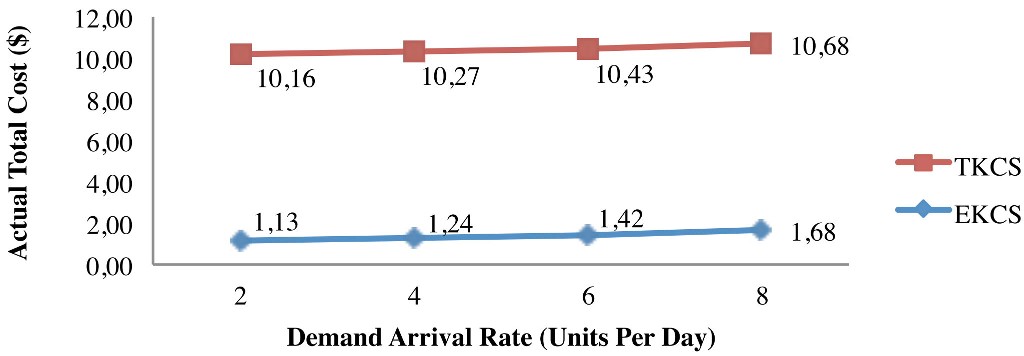

Figure 9. Comparison of EKCS and TKCS (low utilization rates and low backorder and shortage costs).

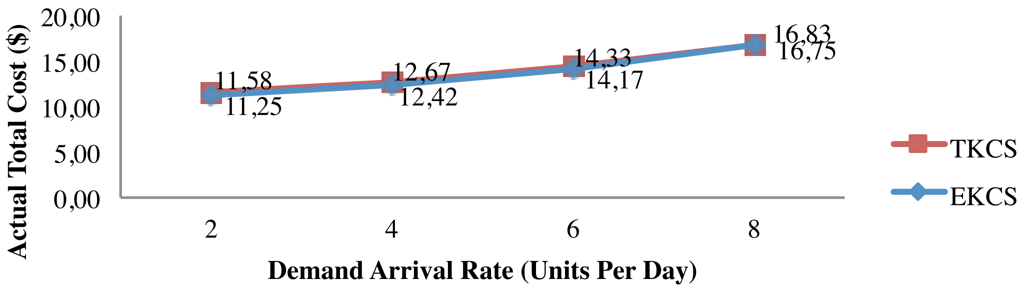

Figure 10. Comparison of EKCS and TKCS (low utilization rates and medium backorder and shortage costs)

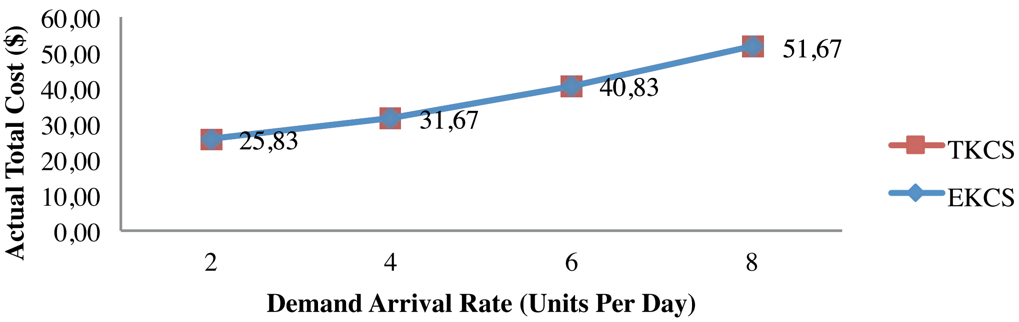

Figure 11. Comparison of EKCS and TKCS (low utilization rates and high backorder and shortage costs).

5.6. EKCS and TKCS: Some comments

The reason for the significant cost difference in Figure 9 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 7 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, 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_1 (Figure 3) in EKCS prevents undispatched kanbans from entering MP when there is no demand, thereby making EKCS a system with truly no inventory.

5.7. 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 10 and 11), 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=20×C_h), EKCS does not hold base stock, 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=200×C_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 ATC curves in Figure 11

5.8. 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 un-dispatched 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.

6. Conclusion and Future Work

This paper 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 Level, S*, and optimal number of kanbans, 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 was 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 outperformed BS, because lean production seems to work best for mass-produced products, such as cars, which is where these systems were 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, Jones Daniel T., & Roos Daniel., 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 research 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 the extra demand queue for instantaneous transmission in EKCS (Queue D_1) is ineffective. This is because, if compared to the TKCS, TKCS also has this functionality but without needing the additional queue. In other words, since we have assumed negligible kanban transfer times, TKCS' kanban queues also act to instantaneously transmit a demand signal. Thus, in this context, adding an extra demand queue for EKCS does not improve its performance at all.
5. Extra demand queues are useful only when EKCS has no stock held 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 similarly to MS/SP/TKCS, assuming negligible kanban transfer times; the optimal number of undispatched kanbans in such a case is one.

7. Implication for Practice

In practical situations, the EKCS is best applied to managing the production of niche or high net value products. Using car manufacturing as an example, niche market cars such as formulae one Indy cars, or customized luxury cars, have extremely low demands. Manufacturers (or "crafters") do not produce them until they receive orders (Sardi, 2009). These cars are only produced one unit per time and only per order. These cars also have low backorder and shortage costs as compared to holding costs (Sardi, 2009). Well-known luxury car brands such as Ferrari care more for their image than mass production. In other words, their concern is not about backorders nor long customer waiting times; but rather, having too many of their cars on the roads cheapening their image. Hence they even create waitlists for customers who wish to purchase their cars. This is very similar to how EKCS behaves – which has an optimal number of undispatched kanban only equal to one and it performs best only during low backorder and shortage costs scenarios.

As for economical cars such as Toyota or Honda, the TKCS would still be the preferred choice of managing their productions. These cars have medium to high backorder and shortage costs compared to holding costs (Womack *et al.*, 2007). That is, managers of such production floors cannot afford to keep their customer waiting, or worse, having their customers walk off. These manufacturers stand to lose out if they do not hold stock in the long run. Also, these cars usually have high demands (Monden, 1993). Similarly, the TKCS performs best during medium and high backorder and shortage cost scenarios, coupled with high utilization rates of more than 50%.

Future work for this research would be to compare Single-Stage, Multiple-Product Kanban Controlled Systems which operate and behave very differently from SP/KCS.

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

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.

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. doi:10.1080/002075497195713

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. doi:10.1016/0167-188X(89)90050-5

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

Dallery, Y. (2000). Extended kanban control system: Combining kanban and base stock. *IIE Transactions*, 32(4): 369-386. doi:10.1080/07408170008963914

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

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

Hopp, W. J., Spearman, M. L. (2004). To pull or not to pull: What is the question? *Manufacturing & service operations management*, 6(2): 133. doi:10.1287/msom.1030.0028

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

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. doi:10.1016/j.rcim.2012.04.020

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. doi:10.1016/S0925-5273(98)00246-1

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.

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.

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.

Monden, Y. (1983). *Toyota production system: practical approach to production management*. Norcross, GA: Industrial Engineering and Management Press, Institute of Industrial Engineers.

Monden, Y. (1993). *Toyota production system: an integrated approach to just-in-time* (2nd ed.). Norcross, Ga.: Industrial Engineering and Management Press. doi:10.1007/978-1-4615-9714-8

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

Nori, V. S., Sarker, B. R. (1998). Optimum number of Kanbans between two adjacent stations. *Production Planning & Control*, 9(1): 60-65. doi:10.1080/095372898234523

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

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

Spearman, M. L., Woodruff, D. L., Hopp, W. J. (1990). CONWIP. A pull alternative to kanban. *International Journal of Production Research*, 28(5): 879-894. doi:10.1080/00207549008942761

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): 553-564. doi:10.1080/00207547708943149

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. doi:10.1016/j.ijpe.2003.10.014

Tardif, V., Maaseidvaag, L. (2001). An adaptive approach to controlling kanban systems. *European Journal of Operational Research*, 132(2): 411-424. doi:10.1016/S0377-2217(00)00119-3

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.

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

Appendix I – Simulation Figures from Arena

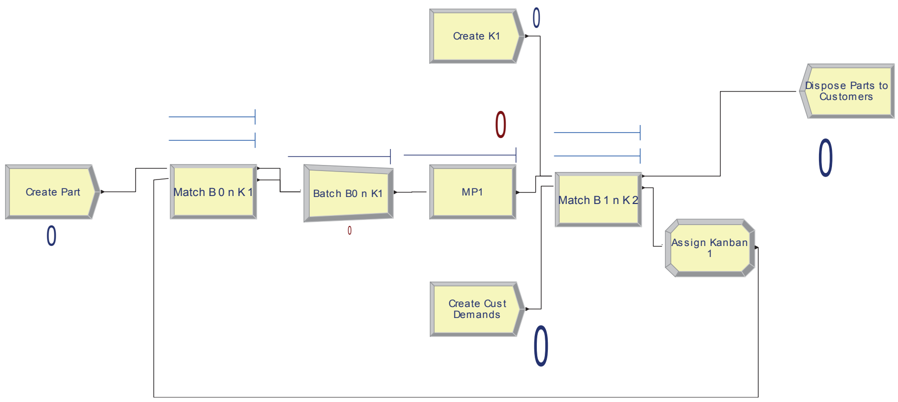

Figure I1. Arena Snapshot of a Single Stage, Single Product Traditional Kanban Control System (SS/SP/TKCS).

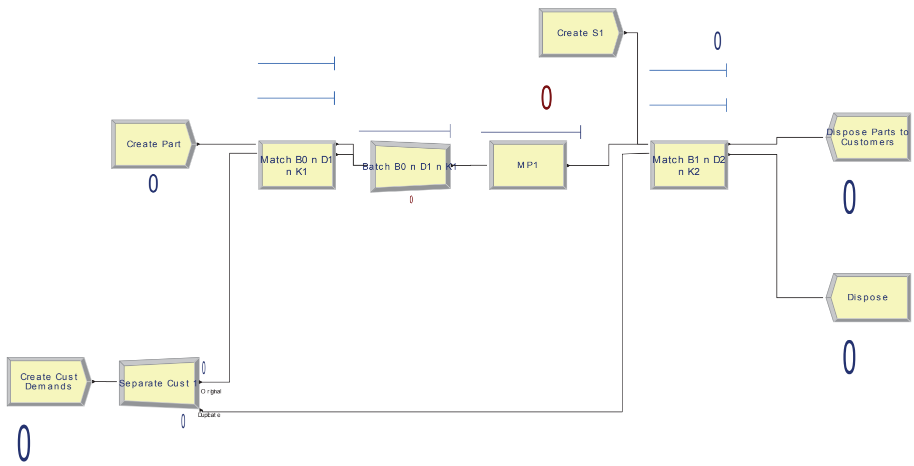

Figure I2. Arena Snapshot of a Single Stage, Single Product Base Stock (SS/SP/BS).

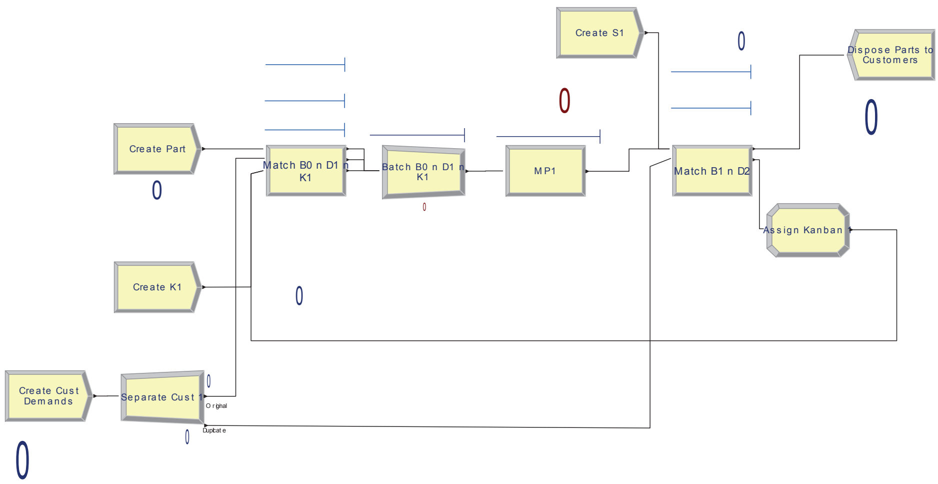

Figure I3. Arena Snapshot of a Single Stage, Single Product Extended Kanban Control System (SS/SP/EKCS).

Appendix II – Statistical Analysis of Simulation Results

We used JMP to conduct hypothesis tests for the simulation results. 29 out of 30 hypothesis tests conducted on the Single Product Kanban Controlled System (SP/KCS) showed that EKCS does outperform TKCS and BS. Hence, the claim that EKCS outperforms TKCS and BS is true.

There were 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. Since the comparisons done were using the ATC mean values; but not their standard deviations, hypothesis tests were done to confirm the results. Hence, for each scenario, 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

We take a specific case as an example. 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:

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.

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

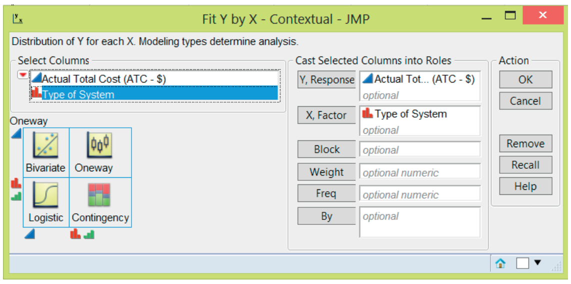

Figure II1. Filling up the Y and X axis for JMP.

Table II1. Simulation Parameters for SS/SP/KCS.

	Actual Total Cost (ATC) $		
Sample Number	Extended Kanban Control System (EKCS)	Traditional Kanban Control Sys-tem (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

After doing a t-test, we obtained:

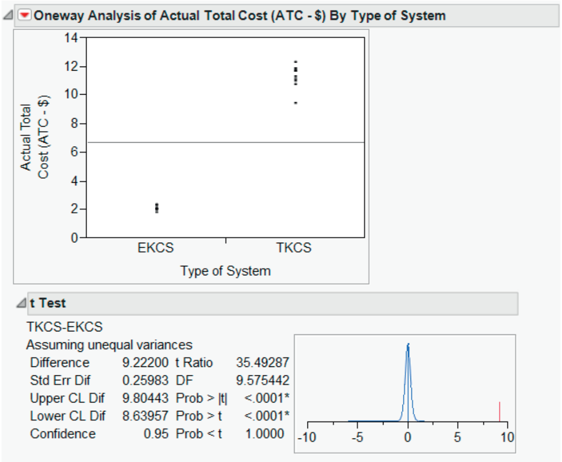

Figure II2. JMP Output showing a Data Plot and t test of the Data Samples.

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.

Results of Hypothesis Test

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 II2 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 ATC is lower than TKCS and BS is true at a 95% confidence level.

Referring to Table II3 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 ATC is lower than TKCS and BS is true at a 95% confidence level.

We now examine the two special cases where EKCS does not outperform.

Table II2. 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

Table II 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

Case 1: For the scenario of Demand Arrival Rate of 12 units per day. We examine the JMP output:

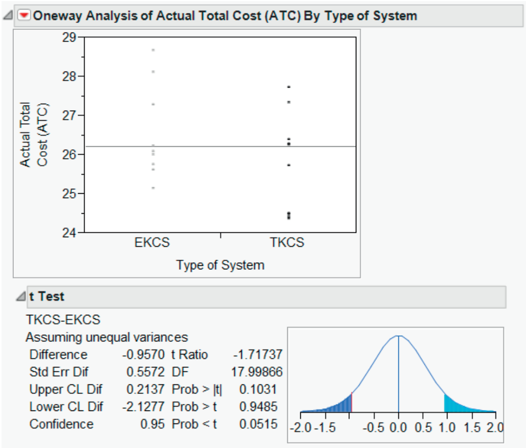

Figure II3. Case 1: Where EKCS only outperforms TKCS by a little.

Referring to Figure II3, 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:

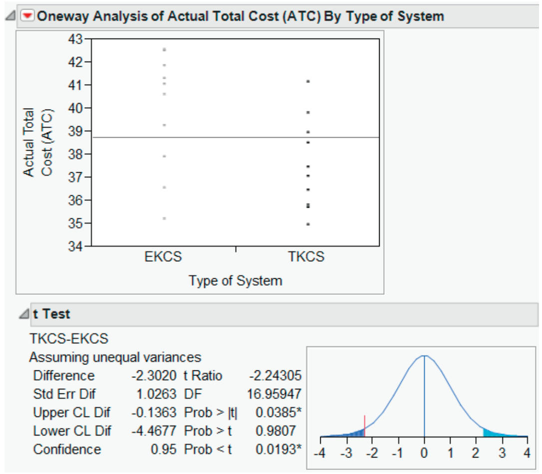

Figure II4. Case 2: Where EKCS does not outperform TKCS.

Referring to Figure II4, 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 II4 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 ATC is lower than TKCS and BS is true at a 95% confidence level.

Table II4. 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

