Department of Social Systems and Management

Discussion Paper Series

No. 1202

A Performance Evaluation of KANBAN, CONWIP and Base-stock in Serial Production Lines

by

Yaghoub Khojasteh-Ghamari, Ryo Sato

April 2008

UNIVERSITY OF TSUKUBA

Tsukuba, Ibaraki 305-8573

JAPAN

A Performance Evaluation of KANBAN, CONWIP and Base-sock in Serial Production Lines

Yaghoub Khojasteh-Ghamari[1] and Ryo Sato [2]

[1] *Graduate School of Systems and Information Engineering, University of Tsukuba, Japan*
E-mail: khojast@sk.tsukuba.ac.jp

[2] *Department of Social Systems and Management, University of Tsukuba, Japan*
E-mail: rsato@sk.tsukuba.ac.jp

Abstract

In managing a production process, choosing a proper card-based control mechanism is an important task to achieve the best performance. We present performance evaluation of the three well-known card-based production control mechanisms, KANBAN, CONWIP and Base-stock in serial production lines. We employ the theory of token transaction systems within a deterministic framework. The performance measures are system throughput and work-in-process (WIP). We compare the minimum WIP, when the system attains maximum possible throughput. Our analytical results show that CONWIP outperforms KANBAN, when the total number of cards in CONWIP is less than that in KANBAN. However, in performance evaluation of these two mechanisms with Base-stock, determining the superior system depends on a configuration of system parameters, such as processing times and number of cards employed in the system.

Keywords: Production control mechanisms; KANBAN; CONWIP; Base-stock;
Little's law; token transaction systems

1. Introduction

This paper deals with comparison of pull production control mechanisms, CONWIP, KANBAN and Base-stock in serial production lines. The most well-known pull mechanism is a KANBAN. In the KANBAN, production authorization cards, called kanbans, are used to control and limit the releases of parts into each production stage (Monden, 1998). The advantage of this mechanism is that the number of parts in every stage is limited by the number of kanbans of that stage.

CONWIP (CONstant Work In Process) proposed by Spearman *et al.* (1990) uses a single card type to control the total amount of average work-in-process (WIP, for short) permitted in the entire line. It is a generalization of KANBAN and can be viewed as a single stage

KANBAN. The primary difference between CONWIP and KANBAN is that CONWIP pulls a job into the beginning of the line and the job goes with a card between production stages, while KANBAN pulls jobs between all stages (Hopp and Spearman, 2001).

Base-stock control system limits the amount of inventory between each production stage and the demand process. The terms “base stock” is borrowed from inventory control theory. It tries to maintain a certain amount of finished parts in each output buffers, subtracting backlogged finished goods demand, if any. This amount is called the *basestock level* of each stage. In other words, the basestock level of a production stage determines the maximum planned inventory of the outputs of the stage (Lee and Zipkin, 1992). To operate a Base-stock, it is necessary to transmit demand information to all production stages as demand occurs.

In a survey paper, Framinan *et al*. (2003) reviewed comparison of CONWIP with other production control systems. Bonvik *et al.* (1997), Bonvik and Gershwin (1996), Paternina-Arboleda and Das (2001), and Yang (2000) used simulation for analysis of performance of different production control mechanisms in serial production processes. Spearman and Zazanis (1992) and Muckstadt and Tayur (1995) showed analytical result on card-based control for serial production processes. When the same number of cards is used in both CONWIP and KANBAN, Spearman and Zazanis (1992) showed that CONWIP produces a higher mean throughput than KANBAN. They pointed out that it holds true because circuits in CONWIP are virtually divided into smaller circuits in KANBAN, and then the cards in KANBAN tend to be "blocked". In the same scenario, Muckstadt and Tayur (1995) considered, simultaneously, four sources of variability in production lines - processing time variability, machine breakdowns, rework and yield loss - and showed some similarities and differences in their effects on the performance of the line. They showed that CONWIP produces a less variable throughput and a lower maximal inventory than KANBAN.

Takahashi *et al*. (2005) applied KANBAN, CONWIP and synchronized CONWIP to supply chains to determine the superior system. Their considered supply chains contain assembly stages with different lead times. Their simulation results show the superiority of both CONWIP and synchronized CONWIP over KANBAN, when all inventory levels among the stages are equally important.

According to the survey done by Framinan *et al.* (2003), in comparison of CONWIP and KANBAN, many authors pointed out that CONWIP outperforms KANBAN when processing times on component operations in production processes are variable. However, Gstettner and Kuhn (1996) arrived at the opposite conclusion. According to their results, KANBAN achieves a given throughput with less WIP than CONWIP. They showed that by choosing appropriate number of cards at each station, KANBAN can outperform CONWIP. Sato and

Khojasteh-Ghamari (2008) resolved this complicated result of comparison between CONWIP and KANBAN in a four-stage serial production line. Their analytical results showed that in a serial production process, CONWIP outperforms KANBAN, when the total number of cards in CONWIP is less than that in KANBAN. Their analysis is based on the theory of deterministic token transaction systems. They also provided analytical results for comparing these two control systems in a tree-shaped production process.

There are also some studies in the literature for comparing Base-stock with the other production control systems. Using simulation analysis, Bonvik *et al.* (1997) compared the performance of different production control policies with respect to WIP and service level in a serial production line with four workstations. They showed that CONWIP demonstrates superior performance in achieving a high service level target with minimal WIP, followed by Base-stock and KANBAN. However, as Framinan *et al.* (2003) mentioned, this result seems to be contradictory to the findings of Duenyas and Patana-anake (1998) and Paternina-Arboleda and Das (2001), which indicated that Base-stock outperforms CONWIP in a serial production process. Framinan *et al.* (2003) recommended further research to clarify these apparently contradictory results.

In this paper, we focus only on serial production processes. By means of the theory of token transaction systems, we first generalize the model considered by Sato and Khojasteh-Ghamari (2008) for an *n*-stage serial production line. We analyze CONWIP and KANBAN, and provide comparative results between the two control systems. Next, we analyze Base-stock, and present comparative results of Base-stock with both CONWIP and KANBAN in an *n*-stage serial production line. We present a performance comparison of these three production control systems in serial production lines.

The remainder of this paper is organized as follows. In Section 2, the concept of token transaction system and related definitions are briefly introduced. Some properties of token transaction systems that are used in analysis are also provided in this section. Section 3 presents comparative results between CONWIP and KANBAN. In Section 4, Base-stock is analyzed and its comparative results with both CONWIP and KANBAN are presented followed by numerical experiments. Section 5 concludes the paper and highlights some future research directions.

2. Modeling production process

Sato and Khojasteh-Ghamari (2008) proposed an integrated framework for card-based production control systems. They employed the theory of *token transaction systems* to model production processes, and used the *activity interaction diagrams* (AID, for short) to represent the considered models. Since this paper is an extension of their model, we will use the same framework for modeling and analysis of our work.

Definition 1. Activity Interaction Diagram (Sato and Praehofer, 1997)
An activity interaction diagram is a diagram that has three kinds of components. They are activities, queues, and connecting arrows. Activities should be connected with queues, and vice versa. That is, in the graph theoretic sense, an AID is a directed bipartite graph.

Figures 1, 2 and 3 are AIDs. Queues in a token transaction system are simplified as FIFO (first-in, first-out) discipline to store objects called *tokens*. Every queue can have at most one input and output arrow. Queues are also referred as *connecting queues*. In a token transaction system, tokens represent parts, products, actors, or data. The AID of a token transaction system for a simple production process is depicted in Figure 1, where activities and queues are represented by squares and ovals, respectively. It shows a serial production process with four workstations governed by CONWIP. The purchased material m, is processed by operations p_1 through p_4 to be a product which is stored in the place b. Output parts of operations p_1, p_2 and p_3 are stored in b_1, b_2 and b_3, respectively. The workers for operations are represented by tokens in w_i (i = 1, 2, 3, 4). The queue C represents the storage place of cards. Figures 2 and 3 show the same production process controlled by KANBAN and Base-stock, respectively.

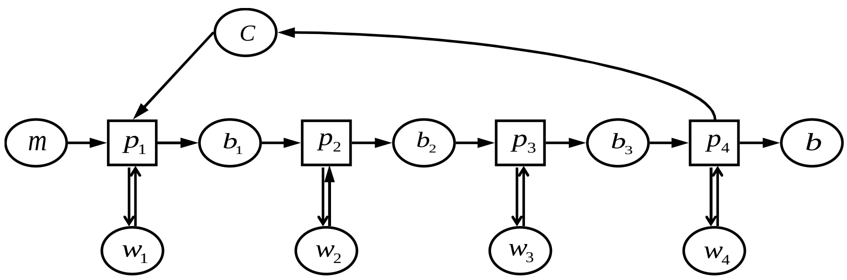

Figure 1. A serial production line with CONWIP

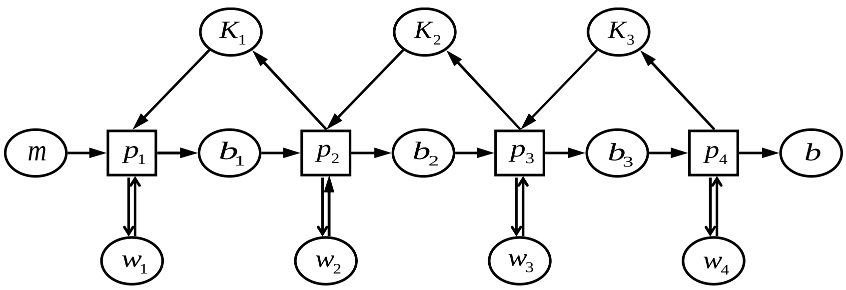

Figure 2. A serial production line with KANBAN

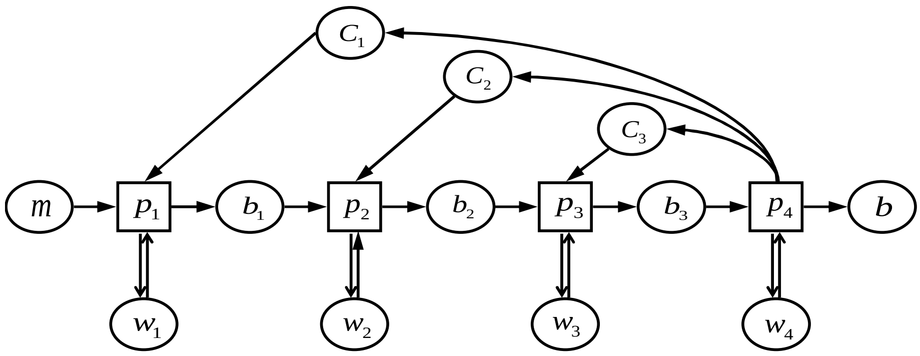

Figure 3. A serial production line with Base-stock

Let A be the set of internal activities, and Q the set of queues. There are two types of output queues for an activity. An output queue of a type gets one token from the activity when it starts, while the other type queue gets a token when the activity finishes. The former queues are called ones of Q_S type, and the others are Q_F type. An activity can have both types output.

In an activity interaction diagram of a token transaction system, a *path* is a series of activities and queues that follows the direction of connecting arrows among them. A path with a coincident start and end node is called a *cycle*, or *circuit*. If a circuit contains different activities and queues (except the start and end), then it is called an *elementary circuit*. When a circuit contains Q_S type queues, then the activities whose outputs are the queues can be eliminated to form the (shorter) circuit. For example, $p_1 b_1 p_2 b_2 p_3 b_3 p_4 C p_1$ in Figure 1 is a circuit and $p_1 b_1 p_2 b_2 p_3 b_3 C p_1$ is also a circuit, because C is a Q_S type output queue of p_4. For a circuit C, the set of activities in C is denoted by $A(C)$. The *cycle mean* of a circuit is defined as the sum of the holding time of the activities of the circuit, divided by the number of tokens in the circuit. The *maximum cycle mean,* λ, of an AID is the maximum value of all cycle means (Baccelli *et al.*, 1992) and is given by

$$\lambda = \max_{\zeta} \frac{|\zeta|_h}{|\zeta|_t},$$

where, ζ ranges over the set of elementary circuits of the AID, $|\zeta|_h$ denotes the sum of the holding times of the activities in the circuit, and $|\zeta|_t$ denotes the number of tokens in the circuit. It is clear that any non-elementary circuit has the cycle mean which is less than or equal to the maximum cycle mean. All the circuits that have maximum value of cycle mean are called *critical circuits*. A circuit consists of an activity and its actors is called an *activity circuit* (for example, $p_1 w_1 p_1$).

Definition 2. Strong connectivity of AID（Sato and Kawai，2007）

Consider an AID of a token transaction system. Let A and $Q = Q_S \cup Q_F$ be the sets of activities and queues, respectively. Let $a \in A$ and $q \in Q$ be arbitrary. If there exist a path from a to q and one from q to a, then the AID and the token transaction system are said to be strongly connected.

For example, in each of Figures 1 to 3, after removing both queues *m* and *b*, the remaining diagrams are strongly connected. The time evolution of a token transaction system can be represented by the *state transition table*. In a state transition table, an activity is said to be "imminent" if its holding time had elapsed from its starting time. There might be several activities which are imminent at a time. When imminent activities finish, the output queues of Q_F type of each imminent activity get respective tokens. When an activity can start, it must start. Once it starts, one token is removed from each input of the activity, one token is held in the activity during the processing time, and one token is added to the outputs of the Q_S type. If no activity can start, then the placement of tokens in the whole process remains the same until the next event comes. The time instant of the next event is defined as the minimum of the due times of activities in operation. So, the next event will become the next "current time" in the state transition table, and then it continues.

Table 1 shows a part of the state transition table of the CONWIP system depicted in Figure 1, where four cards are initially assigned into the system, and the process p_2 has two actors, while each of the others has only one actor. Also, initial inventory in each of b_1, b_2, b_3 and b is set to 0. We assume that enough raw material *m* is always available. In this table, "----" represents that there is no token being processed. That is, the corresponding worker is idle. "1(3)", for example, shows that one token is being processed and it will finish (be imminent) after 3 minutes. As like the p_2 column, two tokens can be processed each of which will be imminent independently.

Table 1. State transition of the CONWIP for a period

time	C	p_1	w_1	b_1	p_2	w_2	b_2	p_3	w_3	b_3	p_4	w_4	b
812	0	1(5)	0	1	1(2),1(9)	0	0	----	1	0	1(2)	0	1
814	0	1(3)	0	0	1(12),1(7)	0	0	1(3)	0	0	----	1	2
817	0	1(5)	0	1	1(9),1(4)	0	0	----	1	0	1(2)	0	2
819	0	1(3)	0	1	1(7),1(2)	0	0	----	1	0	----	1	3
821	0	1(1)	0	0	1(5),1(12)	0	0	1(3)	0	0	----	1	3
822	0	----	1	1	1(4),1(11)	0	0	1(2)	0	0	----	1	3
824	0	1(5)	0	1	1(2),1(9)	0	0	----	1	0	1(2)	0	3

The number of commencement of an activity in a period is called the *activation frequency* of the system. A token transaction system that is strongly connected and live has periodic behavior, and every activity in such a token transaction system has the same number of commencement in the period (Sato and Kawai, 2007). Notice that the numbers of commencement and finish of an activity in a period are the same so that the definition is well defined. In Table 1, for example, the activation frequency is 2, that is every activity starts and ends twice during a period. The *throughput* of a token transaction system is defined as the average value of the number of output tokens from an activity of the system. Since the activation frequency is the same for all of the activities in the system, this definition of throughput is well defined. The *cycle time* of a circuit is defined as the elapsed time for a token to go round on the circuit in the periodic behavior.

2.1. **Little's law** (Little, 1961)

The Little's law can be applied to the processes that show periodic behavior. The law says rigorous relation among cycle time, WIP, and throughput. Let us denote the period by L, and the inventory level at time t by $w(t)$. Then, the average WIP is calculated as follows.

$$WIP = \lim_{T\to\infty}\frac{1}{T}\int_0^T w(t)dt = \frac{1}{L}\int_0^L w(t)dt\ .$$

Thus, it suffices for calculation of average value to consider a period, instead of infinite interval. Similarly, average throughput (*TH*) and cycle time (*CT*) can be calculated for a period. WIP is usually represented as sum of safety and cycle stocks, where the former is considered as buffer for randomness. Since this paper focuses on deterministic model, WIP contains only cycle stock.

Hereafter, we simply write WIP to mean "average WIP", TU to mean "time unit", and PC to mean "pieces". TU can be interpreted as week, hour, minute, and so on.

Theorem 1 (Sato and Khojasteh-Ghamari, 2008). Consider a strongly connected and live token transaction system. Let TH be the throughput of the system in the periodic behavior, C a circuit of the system, w_C the average WIP of tokens on C, and CT_C the cycle time of C. Then, the Little's law holds on C. That is, $CT_C = \dfrac{w_C}{TH}$ holds true.

This theorem shows that the Little's law holds only for circuits. In other words, the average number of total inventories of a production system does not work as the WIP term in the law.

Proposition 1 (Sato and Khojasteh-Ghamari, 2008). Consider a strongly connected and live token transaction system. Let λ be its maximum cycle mean, and *TH* the throughput. Then, $TH = \frac{1}{\lambda}$.

In order to increase the throughput of the system, the maximum cycle mean should be decreased. It means that the structure or WIP placement should be changed. If either factor changes, then another circuit can become critical. This makes situation complicated so that every circuit should be considered and that focusing on the current critical circuit is not enough to improve the performance of a production system.

The sum of WIPs in a KANBAN or CONWIP system is focused on sometimes. It has practical significance. The sum of WIPs is called the *system WIP* in this paper. As Sato and Khojasteh-Ghamari (2008) showed, in a live and strongly connected token transaction system, there exists the least system WIP that attains the throughput of the system.

3. Analysis of comparison between CONWIP and KANBAN

Using analytical queuing network models, Gstettner and Kuhn (1996) provided a quantitative comparison between CONWIP and KANBAN with respect to WIP and throughput in a serial production line including six workstations with exponentially distributed processing times. Contrary to the comparative conclusions in the literature, they showed that KANBAN can achieve a given production rate with a less average WIP level than CONWIP, if the card distribution in the KANBAN is chosen appropriately. They defined the average number of finished parts in the output buffers as the average WIP. Concerning the comparison of the two control systems, Sato and Khojasteh-Ghamari (2008) resolved the complicated result in a serial production line with four workstations. Their analytical result showed that CONWIP outperforms KANBAN, when the total number of cards in CONWIP is less than that in KANBAN. In the following, we generalize their model for a serial production line with *n* workstations.

Proposition 2. Consider a serial production process with *n* workstations controlled by CONWIP and KANBAN. Assume that both systems have the same actors for respective processes, the same activation frequency, and the same throughput. Let *N* and *K* be the total number of cards in CONWIP and KANBAN, respectively. Then, we have the followings.

(i) $N < K$ if and only if $W_C < W_K$,

(ii) $N = K$ if and only if $W_C = W_K$,

where W_C and W_K are the average system WIP for CONWIP and KANBAN, respectively.

Proof. Consider the serial production lines controlled by CONWIP and KANBAN as depicted in Figures 4 and 5, respectively. Let the outmost circuit in the CONWIP be $\bar{C}$ with N tokens, i.e., $\bar{C}$ is $Cp_1b_1p_2b_2...p_{n-1}b_{n-1}C$. Also, let $\bar{K}$ be the set of all non-activity circuits in KANBAN. That is, $\bar{K}$ is $\{A_1, A_2, ..., A_{n-1}\}$, where $A_i (i = 1, 2, ..., n-1)$ is the circuit $K_i p_i b_i K_i$ with k_i tokens. Now, apply the Little's law on each of $\bar{C}$ and $\bar{K}$, and compare the system WIPs. This completes the proof.

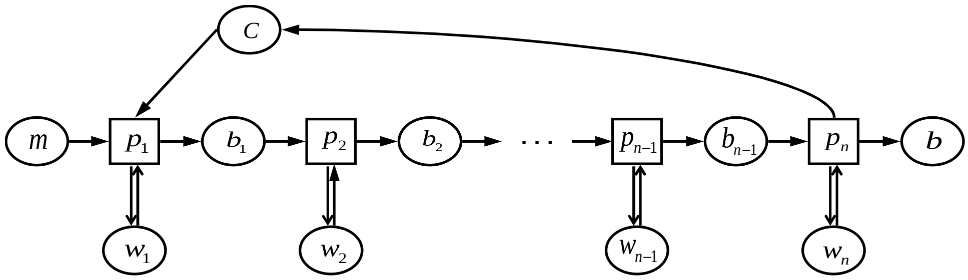

Figure 4. A serial production line with *n* workstations controlled by CONWIP

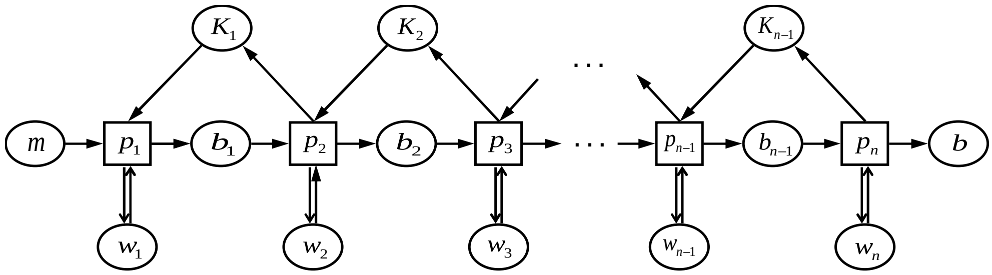

Figure 5. A serial production line with *n* workstations controlled by KANBAN

4. Comparison of Base-stock with CONWIP and KANBAN

Using simulation analysis, Bonvik *et al.* (1997) compared the performance of different production control policies with respect to WIP and service level in a serial production line with four workstations. They showed that CONWIP demonstrates superior performance in achieving a high service level target with minimal WIP, followed by Base-stock and KANBAN. But, as we mentioned in Section 1, their result seems to be contradictory to the other results. In this section, we analyze the Base-stock, and provide an analytical comparison of that with both KANBAN and CONWIP in serial production lines, followed by numerical experiments.

Proposition 3. Consider a serial production process with *n* workstations controlled by KANBAN and Base-stock. Assume that both systems have the same actors for respective processes, the same activation frequency, and the same throughput. Let K and B be the total number of cards in KANBAN and Base-stock, respectively. Then, we have the followings.

(i) if $B - K \le \frac{1}{\lambda}(\sum_{i=1}^{n-2} ih_{i+1})$, then $W_B \le W_K$,

(ii) if $B = K$, then $W_B < W_K$,

where λ is the maximum cycle mean, h_i the processing time of workstation i, and W_K and W_B are the average system WIP for KANBAN and Base-stock, respectively.

Proof. Consider the serial production lines controlled by KANBAN and Base-stock as depicted in Figures 5 and 6, respectively. Let $\bar{K}$ be the set of all non-activity circuits in KANBAN. That is, $\bar{K}$ is $\{A_1, A_2, ..., A_{n-1}\}$, where A_i (i=1, 2, …, $n-1$) is the circuit $K_i p_i b_i K_i$ with k_i tokens. In the Base-stock, let $\bar{B}$ be the set of all non-activity circuits, i.e., $\bar{B}$ is $\{H_1, H_2, ..., H_{n-1}\}$, where $H_i (i = 1, 2, ..., n-1)$ is the circuit $C_i p_i b_i p_{i+1} b_{i+1} ... p_{n-1} b_{n-1} C_i$ with m_i tokens. Now, apply the Little's law on each of $\bar{K}$ and $\bar{B}$, and compare the system WIPs. This completes the proof.

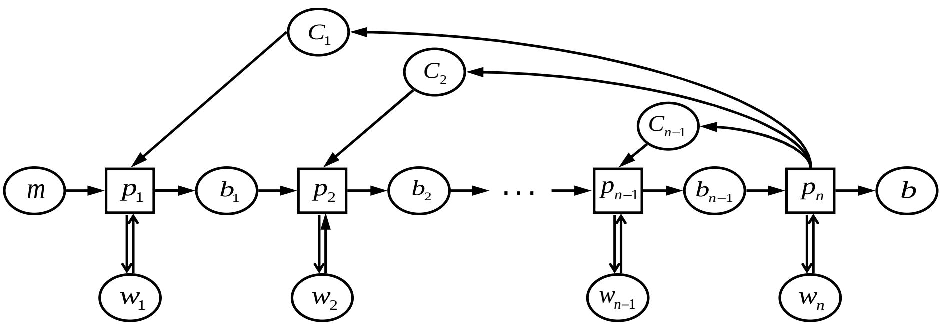

Figure 6. A serial production line with n workstations controlled by Base-stock

Proposition 3 is one of the best possible forms in the sense that the respective converses do not hold true. It suffices to show that there exists at least an example for the converse. The following example shows that each converse implications of (i) and (ii) in Proposition 3 does not hold.

Example 1. Consider a serial production line including four workstations with KANBAN and Base-stock as depicted in Figures 2 and 3, respectively. Processing times at p_1, p_2, p_3 and p_4 are set as 5, 12, 10 and 7 [TU], respectively. That is, $h_1 = 5$, $h_2 = 12$, $h_3 = 10$ and $h_4 = 7$. The process p_2 has two actors, while each of the others has only one actor. Also, initial inventory for every part is set to 0. We assume that enough raw material m is always available. Cases 1-BAS and 1-KAN below show the periodic behavior of Base-stock and

KANBAN, respectively.

Case **1-BAS.** Table 2 shows the state transition table for the production process with Base-stock. Initial cards are set as $C_1 = 4$, $C_2 = 3$ and $C_3 = 2$, which are the minimum number of cards to attain the maximum possible throughput. The system shows a periodic behavior every 10 [TU]. Activity circuit $p_3 w_3 p_3$ is critical with maximum cycle mean $\lambda = 10$. Each activity starts once in a period. The throughput is 1/10, and the system WIP is equal to 5.90. That is, $W_B = 5.90$. It can be verified that the amount of system WIP is minimum to attain the throughput 1/10.

Case **1-KAN.** Table 3 gives the state transition table for the same production process with KANBAN. Initial cards are set as $k_1 = k_3 = 1$ and $k_2 = 2$, that is $K = 4$. The system shows a periodic behavior every 10 [TU]. Circuit $p_3 w_3 p_3$ is critical with $\lambda = 10$. Each activity starts once in a period. The throughput is 1/10, and the system WIP is equal to 6.30 ($W_K = 6.30$), which is the minimum value to attain the throughput.

Table 2. State transition of 1-BAS for a period

time	C_1	p_1	w_1	b_1	C_2	p_2	w_2	b_2	C_3	p_3	w_3	b_3	p_4	w_4	b
986	0	1(3)	0	0	0	---- ,1(10)	1	1	1	1(1)	0	0	----	1	1
987	0	1(2)	0	0	0	---- ,1(9)	1	0	0	1(10)	0	0	1(7)	0	1
989	0	----	1	1	0	---- ,1(7)	1	0	0	1(8)	0	0	1(5)	0	1
994	0	1(5)	0	0	0	1(12),1(2)	0	0	1	1(3)	0	0	----	1	2
996	0	1(3)	0	0	0	1(10), ----	1	1	1	1(1)	0	0	----	1	2

Table 3. State transition of 1-KAN for a period

time	K_1	p_1	w_1	b_1	K_2	p_2	w_2	b_2	K_3	p_3	w_3	b_3	p_4	w_4	b
927	0	1(5)	0	0	0	1(2),1(12)	0	0	0	1(10)	0	0	1(7)	0	1
929	0	1(3)	0	0	0	---- ,1(10)	1	1	0	1(8)	0	0	1(5)	0	1
932	0	----	1	1	0	---- ,1(7)	1	1	0	1(5)	0	0	1(2)	0	1
934	0	----	1	1	0	---- ,1(5)	1	1	0	1(3)	0	0	----	1	2
937	0	1(5)	0	0	0	1(12),1(2)	0	0	0	1(10)	0	0	1(7)	0	2

By using Case 1-KAN as an example, we show how to calculate the system WIP based on the state transition table. Consider Table 3. By observing the state transition table for a period, at time 927, five tokens (all of them are being processed at p_1 through p_4) remain in the system for 2 time unit (929-927=2). At the next event time (i.e. 929), six tokens remain in the system, but for 3 time unit (932-929=3). Similarly, 7 tokens remain in the system for the next

2 and 3 [TU]. This yields (2*5)+(3*6)+(2*7)+(3*7)=63 [PC*TU] as the total holding and waiting times for a period. Since the period is 10 [TU], the system WIP is 63/10 [PC]. That is, $W_K = 6.3$.

Case 1-BAS and 1-KAN show that each converse implications of (i) and (ii) in Proposition 3 does not hold. Because it can be simply verified that $W_B < W_K$, however, $B \neq K$, and also $B - K = 9 - 4 > \frac{h_2 + 2h_3}{\lambda} = \frac{12 + 2(10)}{10}$, in the same notation of Proposition 3.

Proposition 4. Consider a serial production process with n workstations controlled by CONWIP and Base-stock. Assume that both systems have the same actors for respective processes, the same activation frequency, and the same throughput. Let N and B be the total number of cards in CONWIP and Base-stock, respectively. Then, we have the followings.

(i) if $B - N \le \frac{1}{\lambda}(\sum_{i=1}^{n-2} i h_{i+1})$, then $W_B \le W_C$,

(ii) if $B = N$, then $W_B < W_C$,

where λ is the maximum cycle mean, h_i the processing time of workstation i, and W_C and W_B are the average system WIP for CONWIP and Base-stock, respectively.

Proof. The proof is similar to the proof of Proposition 3.

The following example shows that the converse implication of (i) in the above proposition does not hold.

Example 2. This example shows that the converse implication of (i) in Proposition 4 does not hold. Consider a serial production line including four workstations with CONWIP and Base-stock as depicted in Figures 1 and 3, respectively. Processing times at p_1, p_2, p_3 and p_4 are set as 5, 12, 3 and 2 [TU], respectively. That is, $h_1 = 5$, $h_2 = 12$, $h_3 = 3$ and $h_4 = 2$. Same as the previous example, the process p_2 has two actors, while each of the others has only one actor. Also, initial inventory for every part is set to 0, and it is assumed that enough raw material m is always available. Cases 2-BAS and 2-CON below show the results for Base-stock and CONWIP, respectively.

Case **2-BAS.** Table 4 shows the state transition table for the production process with Base-stock. Initial cards are set as $C_1 = 4$, $C_2 = 3$ and $C_3 = 1$, which are the minimum number of cards to attain the maximum possible throughput ($B = 8$). The system shows a periodic behavior every 12 [TU]. Activity circuit $p_2 w_2 p_2$ is critical with $\lambda = 6$. Each activity starts twice in a period. The throughput is 2/12, and the system WIP is equal to 5.67. That is $W_B = 5.67$.

Case **2-CON**. The state transition table for the same production process with CONWIP has been given in Table 1. Four cards are initially assigned in the system ($N=4$), which is the minimum number of cards to attain the maximum possible throughput. The system shows a periodic behavior every 12 [TU]. Circuit $p_2 w_2 p_2$ is critical with $\lambda=6$. Each activity starts twice in a period. The throughput is 2/12, and the system WIP is equal to 5.67 ($W_C = 5.67$).

Table 4. State transition of 2-BAS for a period

time	C_1	p_1	w_1	b_1	C_2	p_2	w_2	b_2	C_3	p_3	w_3	b_3	p_4	w_4	b
934	0	1(5)	0	0	0	1(7),1(12)	0	0	0	1(3)	0	0	----	1	1
937	0	1(2)	0	0	0	1(4),1(9)	0	0	0	----	1	0	1(2)	0	1
939	0	1(5)	0	1	1	1(2),1(7)	0	0	1	----	1	0	----	1	2
941	0	1(3)	0	0	0	1(12),1(5)	0	0	0	1(3)	0	0	----	1	2
944	0	----	1	1	0	1(9),1(2)	0	0	0	----	1	0	1(2)	0	2
946	0	1(5)	0	0	0	1(7),1(12)	0	0	0	1(3)	0	0	----	1	3

Case 2-BAS and 2-CON show that the converse implication of (i) does not hold. Because $W_B \le W_C$, but $B-N=8-4>\dfrac{h_2+2h_3}{\lambda}=\dfrac{12+2(3)}{6}$, in the same notation of Proposition 4.

However, we do not have examples for the converse of (ii) yet. It seems that in Proposition 4 with $n>2$, when the both systems perform optimally, the if-condition of part (ii) cannot be satisfied under any circumstance. Optimality here refers the fact that the system attains maximum possible throughput by employing the least number of cards, and hence has a minimum amount of system WIP. As it can be seen in Figures 4 and 6, the outmost circuits in the both systems (i.e., circuit $Cp_1b_1p_2b_2...p_{n-1}b_{n-1}C$ in the CONWIP, and $C_1p_1b_1p_2b_2...p_{n-1}b_{n-1}C_1$ in the Base-stock) have the same components. Therefore, in order for both systems to attain the maximum rate of throughput with minimum amount of WIP, the same number of cards is required to assign initially into each of C and C_1. In fact, the number of cards assigned into C is the total number of cards in the CONWIP, denoted by N. However, the Base-stock needs more cards in the other non-activity circuits (i.e., circuits $C_ip_ib_ip_{i+1}b_{i+1}...p_{n-1}b_{n-1}C_i$, $1<i<n$) to operate. Thus, $B>N$. This would be a reason that we failed to find an example to show whether the converse implication of (ii) holds true.

For a serial production process in the Proposition 3, many KANBAN and Base-stock cases, which have the same level of throughput satisfy the if-condition of (i), and then $W_B \le W_K$ certainly holds. However, in the following, we give an example that dissatisfy the if-condition, and then $W_B > W_K$. In a similar way, in Proposition 4 (for case CONWIP and

Base-stock), this example satisfies $B-N>\frac{1}{\lambda}(\sum_{i=1}^{n-2} ih_{i+1})$, and then $W_B>W_C$. This implies that Base-stock is not always superior to both KANBAN and CONWIP.

Example 3. Concerning Propositions 3 and 4, this example shows that Base-stock does not necessarily outperform KANBAN and CONWIP. Consider the production process shown in Figures 1, 2 and 3 with CONWIP, KANBAN and Base-stock, respectively. We set the processing times at p_1, p_2, p_3 and p_4 as 5, 12, 10 and 1 [TU], respectively. That is, $h_1=5$, $h_2=12$, $h_3=10$ and $h_4=1$. Same as the previous examples, the process p_2 has two actors, while each of the others has only one actor. Also, initial inventory for every part is set to 0. Cases 3-BAS, 3-CON and 3-KAN below show the results for Base-stock, CONWIP and KANBAN, respectively.

Case **3-BAS.** The state transition table for Base-stock is given in Table 5. Initial cards are set as $C_1=3$, $C_2=3$ and $C_3=2$, which are the minimum number of cards to attain the maximum possible throughput. The system shows a periodic behavior every 10 [TU], and each activity starts once in a period. Activity circuit $p_3w_3p_3$ is critical with maximum cycle mean $\lambda=10$. The throughput is 1/10, and the system WIP is equal to 6.60. That is, $W_B=6.60$.

Case **3-CON**. Table 6 shows the state transition table for the production process with CONWIP. Three cards are initially assigned into the system ($N=3$), which is the minimum number of cards to attain the maximum possible throughput. Circuit $p_3w_3p_3$ is critical with $\lambda=10$. The throughput is 1/10, and $W_C=5.30$.

Case **3-KAN.** The state transition table is given in Table 7. Initial cards are set as $k_1=k_3=1$ and $k_2=2$, namely, $K=4$. Circuit $p_3w_3p_3$ is critical with $\lambda=10$. Each activity starts once in a period, the period is 10 [TU], the throughput is 1/10, and $W_K=6.30$.

Table 5. State transition of 3-BAS for a period

time	C_1	p_1	w_1	b_1	C_2	p_2	w_2	b_2	C_3	p_3	w_3	b_3	p_4	w_4	b
808	0	1(5)	0	0	1	---- ,1(7)	1	0	1	1(9)	0	0	----	1	1
813	0	----	1	0	0	1(12),1(2)	0	0	1	1(4)	0	0	----	1	1
815	0	----	1	0	0	1(10), ----	1	1	1	1(2)	0	0	----	1	1
817	0	----	1	0	0	1(8), ----	1	0	0	1(10)	0	0	1(1)	0	1
818	0	1(5)	0	0	1	1(7), ----	1	0	1	1(9)	0	0	----	1	2

Table 6. State transition of 3-CON for a period

time	C	p_1	w_1	b_1	p_2	w_2	b_2	p_3	w_3	b_3	p_4	w_4	b
797	0	1(5)	0	0	---- ,1(7)	1	0	1(10)	0	0	1(1)	0	1
798	0	1(4)	0	0	---- ,1(6)	1	0	1(9)	0	0	----	1	2
802	0	----	1	0	1(12),1(2)	0	0	1(5)	0	0	----	1	2
804	0	----	1	0	1(10), ----	1	1	1(3)	0	0	----	1	2
807	0	1(5)	0	0	1(7), ----	1	0	1(10)	0	0	1(1)	0	2

Table 7. State transition of 3-KAN for a period

time	K_1	p_1	w_1	b_1	K_2	p_2	w_2	b_2	K_3	p_3	w_3	b_3	p_4	w_4	b
788	0	1(4)	0	0	0	1(1),1(11)	0	0	0	1(9)	0	0	----	1	1
789	0	1(3)	0	0	0	---- ,1(10)	1	1	0	1(8)	0	0	----	1	1
792	0	----	1	1	0	---- ,1(7)	1	1	0	1(5)	0	0	----	1	1
797	0	1(5)	0	0	0	1(12),1(2)	0	0	0	1(10)	0	0	1(1)	0	1
798	0	1(4)	0	0	0	1(11),1(1)	0	0	0	1(9)	0	0	----	1	2

Comparison of the system WIPs in cases 3-BAS and 3-CON reveals the fact that Base-stock does not necessarily outperform CONWIP. Because it can be easily verified that $B-N=8-3>\frac{h_2+2h_3}{\lambda}=\frac{12+2(10)}{10}$, and $W_B>W_C$. This example also shows that Base-stock does not necessarily outperform KANBAN, either. Because in cases 3-BAS and 3-KAN, one can see that $B-K=8-4>\frac{h_2+2h_3}{\lambda}=\frac{12+2(10)}{10}$, and $W_B>W_K$. Therefore, the if-conditions of Propositions 3 and 4 are meaningful, and as a consequence, Base-stock is not always superior to either CONWIP or KANBAN. In fact, a configuration of parameters, such as processing time of activities, number of workers, and number of cards employed in the line decides the superior system in certain situation.

5. Conclusions

In this paper, by employing the framework proposed by Sato and Khojasteh-Ghamari (2008), we have compared the performance of three production control mechanisms, CONWIP, KANBAN and Base-stock in serial production lines. Using the theory of token transaction system and within the same framework, we extended their model to a serial production line with n workstations.

In comparison of CONWIP and KANBAN in serial production lines, Proposition 2 has given a complete characterization. That is, CONWIP is superior to KANBAN, if and only if, the total number of cards in CONWIP is less than that in KANBAN. Superiority here refers the fact that the minimum system WIP is smaller than the other to attain the same rate of throughput.

In comparison of Base-stock with KANBAN and CONWIP, the situation is complicated, so that we cannot completely characterize it. Base-stock outperforms KANBAN in some cases, while it does not in other cases. This happens in the comparison with CONWIP, too. We have clarified that the superiority of one over another is determined by a configuration of parameters, such as processing time of activities, number of workers for activities, and number of cards employed in the line (Propositions 3 and 4). In a certain production line with a configuration, for example, Base-stock is superior to KANBAN. Therefore, if a research focused on a line with such a certain configuration, then it could result in the superiority of Base-stock to KANBAN.

There are some related topics remained. Effect of randomness needs to be considered. Original idea of CONWIP does not restrict to FIFO policy. Sophisticated policy may lead the process to different performance. Such policies should be sought in future research.

References

Baccelli, F.L., Cohen, G., Olsder, G.J. and Quadrat, J.P. (1992) *Synchronization and Linearity -- An algebra for discrete event systems.* John Wiley.

Bonvik, A.M. and Gershwin, S.B. (1996) Beyond Kanban: Creating and analyzing lean shop floor control policies. *Manufacturing and Service Operations Management Conference Proceeding*, Durtmouth College, The Amos Tuck School, Hannover, New Hampshire, 46-51.

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

Duenyas, I. and Patana-anake, P. (1998) Base-stock control for single product tandem make-to-stock systems. *IIE Transactions*, 30, 31-39.

Framinan, J.M., Gonzalez, P.L. and Ruiz-Usano, R. (2003) The CONWIP production control system: Review and research issues. *Production Planning and Control*, 14, 255–265.

Gstettner, S. and Kuhn, H. (1996) Analysis of production control systems kanban and CONWIP. *International Journal of Production Research*, 34 (11), 3253–3274.

Hopp, W.J. and Spearman, M.L. (2001) *Factory physics: foundations of manufacturing management.* Irwin/McGraw-Hill.

Lee, Y.-J. and Zipkin, P. (1992) Tandem queues with planned inventories. *Operations Research*, 40(5), 936-947.

Little. J.D.C. (1961) A proof for the queuing formula: $L = \lambda W$. *Operations Research*, 9, 383-387.

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

Muckstadt, J.A. and Tayur, S.R. (1995) A comparison of alternative Kanban control mechanisms: I, background and structural results. *IIE Transactions*, 27 (1), 140–150.

Paternina-Arboleda C.D. and Das, T.K. (2001) Intelligent dynamic control policies for serial production lines. *IIE Transactions*, 33(1), 65–77.

Sato, R. and Kawai, A. (2007) Organizing a business process that realizes required throughput: the principle and an application to information systems for SCM. Department of Social Systems and Management, Discussion Paper Series No.1184, University of Tsukuba.

Sato, R. and Khojasteh-Ghamari, Y. (2008) Managing assembly production processes with KANBAN and CONWIP. Department of Social Systems and Management, Discussion Paper Series No.1191, University of Tsukuba.

Sato, R. and Praehofer H. (1997) A discrete event model of business system - A systems theoretic foundation for information systems analysis: Part 1. *IEEE Transactions on Systems, Man, and Cybernetics*, 27(1), 1-10.

Spearman, M.L. and Zazanis, M.A. (1992) Push and pull production systems: issues and comparisons. *Operations Research*, 40, 521–532.

Spearman, M.L., Woodruff, D.L. and Hopp, W.J. (1990) CONWIP: A pull alternative to Kanban. *International Journal of Production Research*, 23, 879–894.

Takahashi, K., Myreshka, and Hirotani, D. (2005) Comparing CONWIP, synchronized CONWIP, and Kanban in complex supply chains. *International Journal of Production Economics*, 93-94, 25-40.

Yang, K.K. (2000) Managing a flow line with single-Kanban, dual-Kanban or CONWIP. *Production and Operations Management*, 9(4), 349-366.