Developing a unified framework for performance analysis of production control mechanisms

Yaghoub Khojasteh-Ghamari†* and Ryo Sato‡

† Doctoral program, Graduate School of Systems and Information Engineering, University of Tsukuba, Japan
Email: khojast@sk.tsukuba.ac.jp
Tel: +81-80-5521-1062

‡ Department of Social Systems and Management, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
Email: rsato@sk.tsukuba.ac.jp

POMS 19th Annual Conference
La Jolla, California, U.S.A.
May 9 to May 12, 2008

*: Corresponding author
Abstract

There exists controversy on the superiority of logistics control systems. In this paper, we propose a unified framework for analyzing and comparing of production control mechanisms. KANBAN, CONWIP and Base-stock control systems are focused on and analyzed as three well-known examples of production control systems. The periodic behavior of a token transaction system is used. By using the theory of token transaction systems, and employing the Little’s law, we show how minimum WIP (Work-In-Process) of a system can be calculated that allows the system to have maximum possible throughput. We also provide a performance comparison for the three production control systems, KANBAN, CONWIP and Base-stock in serial production processes. We compare the minimum WIP of the system for the three policies, when the system attains maximum possible throughput.

Keywords: production control systems; KANBAN; CONWIP; Base-stock, Little’s law; critical circuit

1. Introduction

Over the years many production control systems have been proposed. Some are push,
some pull (e.g., KANBAN system) and some hybrid (CONWIP system). (Hereafter, we simply write KANBAN, CONWIP and Base-stock to mean respective KANBAN, CONWIP and Base-stock controlled production processes.) The best 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. The advantage of this mechanism is that the number of parts in every stage is limited by the number of kanbans of that stage. In a KANBAN, instead of directly controlling the throughput, kanbans are used to authorize production or transportation of materials such that the parts are pulled and WIP (work-in-process) is visualized and controlled.

CONWIP (CONstant Work In Process) proposed by Spearman et al. (1990) uses a single card type to control the total amount of WIP permitted in the entire line. It is a generalization of KANBAN and can be viewed as a single stage KANBAN. A CONWIP behaves as follow: when a job order arrives to a CONWIP line, a card is attached to the job, provided cards are available at the beginning of the line; otherwise, the job must wait in a backlog. When a job is processed at the final station, the card is removed and sent back to the beginning of the line, where it might be attached to the next job waiting in the backlog. No order can enter the line without its corresponding card. 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 workstations, while KANBAN pulls jobs between all stations (Hopp and Spearman, 2001).

Base-stock limits the amount of inventory between each production stage and the demand process. 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. This can be done by using a card-based system similar to KANBAN: attach a set of cards to each unit of finished goods, using as many cards as there are production stages. When a unit is delivered to satisfy demand, remove its cards, and distribute one card to each production stage. This authorizes production in the same way as in KANBAN. When an operation is completed, the card is attached to the product, and will follow the product until it leaves the finished goods buffer. If there is only one production stage, this control is identical to KANBAN, and hence CONWIP. The difference for longer lines is in the flow of information, since Base-stock distributes demand information to all stages simultaneously. Different basestock levels are attained by sending out an appropriate number of initialization cards to the various machines to fill buffers to the basestock levels.

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. Spearman and Zazanis (1992) and Muckstadt and Tayur (1995) showed analytical result on card-based control for serial production processes. Spearman and Zazanis (1992) showed that CONWIP produces a higher mean throughput than KANBAN. 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 showed the superiority of both CONWIP and synchronized CONWIP over KANBAN, when all inventory levels among the stages are equally important.

According to the survey 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 at finished part buffer. They showed that by choosing appropriate number of cards at each station, KANBAN can outperform CONWIP. Also, CONWIP has been compared against Base-stock. While Duenyas and Patana-anake (1998) and Paternina-Arboleda and Das (2001) indicated that Base-stock outperforms CONWIP, Bonvik et al. (1997) reached to the opposite conclusion.

This paper proposes a unified framework for analyzing and comparing of production control mechanisms by developing the theory of token transaction systems. The theory shows how the three indices represented in Little's law (Little, 1961) are decided by the structure of a production process with control-cards and deployment of WIP. In other word, we show how the minimum WIP of a system can be calculated that allows the system to have maximum possible throughput. As an application of the theory, we resolve complicated result of comparison among CONWIP, KANBAN and Base-stock in serial production processes.

The reminder of this paper is organized as follows. In Section 2, the concept of token transaction system and related definitions are introduced. In Section 3, CONWIP, KANBAN and Base-stock are analyzed. Section 4 concludes the paper.
2. Modeling production process

In modeling production processes with control mechanisms, this paper employs the concept of business transaction system (Sato and Praehofer, 1997) that is based on the DEVS formalism for discrete-event systems (Zeigler, 1976). We consider specific type of business transaction systems, where queues are simplified as usual FIFO (first-in, first-out) to store objects called tokens, and every queue can have at most one input and output arrow. We call such system a token transaction system. In a token transaction system, tokens represent parts, products, servers (machine or operator), or data. Queues are also referred as connecting queues. In a business transaction system, the components and connecting structure are represented by activity interaction diagrams (AID, for short).

**Definition.** Activity Interaction Diagram (AID) (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.

![Figure 1. Components of activity interaction diagrams](image-url)
There are several ways to draw an AID as shown in Figure 1. For example, the AID of a token transaction system for a simple production process that is controlled by CONWIP is shown in Figure 2, where activities and queues are represented by squares and ovals, respectively. Queue $b_i (i = 1, 2, 3)$ is the output buffer of operation $p_i$, and $b$ is the finished product buffer. We imagine an unlimited store of raw materials $m$, before the first station. The workers of operation $p_i (i = 1, 2, 3, 4)$ are represented by tokens in $w_i$. The queue $C$ represents the storage place of cards.

![Figure 2. A serial line with CONWIP](image)

Let $A$ be the set of internal activities, and $Q$ the set of queues. The output queues of an activity are specified as one of two types. 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.
Rule of transfer of tokens in a token transition system:

An activity starts when its starting condition is met. That condition is defined by relation between the input queues. Once started, an activity will finish after prescribed processing time (or holding time) for a token. When an activity 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 in the outputs of the $Q_s$ type. When an activity finishes, one token is added to each of the output queues of $Q_f$ type of the activity.

Now we define the dynamics of a token transaction system. The time evolution of a token transaction system is defined by the state transition function, which is originally defined by the set theoretic notation by Sato and Praehofer (1997). For a token transaction system we can use state transition table, because the content of queue variables are simple FIFO tokens. With the above rule, the state transition of the process depicted in Figure 2 is defined, and generates the state transition table, Table 1.

The time evolution of CONWIP in Figure 2 is determined by specifying the starting condition of activities and movement of tokens. By assuming an unlimited store of raw material, $p_1$ starts its processing if more than one card exists in $C$ and if the worker is available (that is, if the worker is not busy). When $p_1$ finishes, it outputs a token to each of
$b_i$ and $w_i$. One token in $b_i$ represents combination of a part and a card. Each of the operations $p_2$, $p_3$ and $p_4$ will start its operation if more than one pair of part and an attached card exist in the respective input buffer, and if the respective actor is available.

When $p_4$ starts, it also outputs a card to $C$. As a whole, Table 1 will come out. In Table 1, "---" 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 after 3 minutes. As like the column $p_2$, two tokens can be processed each of which will finish independently.

Table 1. State transition table of a CONWIP (Initial condition: $C$ has 4 tokens. Each of $p_1$, $p_3$ and $p_4$ has one actor, while $p_2$ has two. Each activity is idle, and inventory in each of $b_1$, $b_2$, $b_3$ and $b$ is 0.)

<table>
<thead>
<tr>
<th>time</th>
<th>C</th>
<th>p1</th>
<th>w1</th>
<th>b1</th>
<th>p2</th>
<th>w2</th>
<th>b2</th>
<th>p3</th>
<th>w3</th>
<th>b3</th>
<th>p4</th>
<th>w4</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>187</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>0</td>
<td>1(11),1(8)</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>0</td>
<td>1(6)</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>190</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>1(8),1(5)</td>
<td>0</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>191</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>1(7),1(4)</td>
<td>0</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>1(2)</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>193</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>1</td>
<td>1(5),1(2)</td>
<td>0</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>0</td>
<td>1(6)</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>195</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>0</td>
<td>1(3),1(12)</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>196</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>1(2),1(11)</td>
<td>0</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>198</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>0</td>
<td>1(12),1(9)</td>
<td>0</td>
<td>1</td>
<td>1(1)</td>
<td>0</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>199</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>0</td>
<td>1(11),1(8)</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>0</td>
<td>1(6)</td>
<td>0</td>
<td>30</td>
</tr>
</tbody>
</table>

In an activity interaction diagram of a business 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, $p_1b_1p_2b_2p_3b_3p_4Cp_1$ in Figure 2 is a circuit and $p_1b_1p_2b_2p_3b_3Cp_1$ 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, $\lambda$, 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, $\zeta$ ranges over the set of elementary circuits of the AID, and $|\zeta|_h$ 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.

The number of commencement of an activity in a period is called the activation frequency of the system. 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 the previous example (Table 1), 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 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 shows rigorous relation among cycle time, WIP, and throughput. In a
steady state queuing system, the law states that the average number of units in the system is
equal to the product of average arrival rate of units and the average waiting time of the units
in the system. If $W$ denotes the average number of the units in the system, $L$ the average time
spent by a unit in the system, and $T$ the average arrival rate, then, $L = TW$.

3. Analysis of production control systems

By applying the theory of token transaction system, we compare CONWIP, KANBAN,
and Base-stock in a serial production line. In comparison of different control schemes, as
Framinan et al. (2003) pointed out, optimized parameters should be used, i.e., the minimum
WIP, which attains maximum possible throughput for a token transaction system.

The sum of WIPs is called the system WIP in this paper. Notice that tokens in a token transaction system correspond to cards, parts, or actors. Since tokens decide whether an activity can start processing, any of the three should be considered in analysis and design of dynamic behavior. Deployment of tokens in a system decides the throughput and WIP, and hence the cycle time.

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. According to their results, KANBAN can result in a lower average WIP level than CONWIP for a given production rate if the card distribution in the KANBAN is chosen appropriately. They defined the average number of finished parts in the output buffer of a station as the average WIP. In the following, we present comparative analysis of the performance measures between CONWIP, KANBAN, and Base-stock in serial production processes.

The CONWIP process in Figure 2 is composed of four processes $p_1$ through $p_4$, and respective actors $w_1$ through $w_4$, and each process has output $b_1$ or $b$. The corresponding
KANBAN process for the same serial production line is specified as Figure 3. The first process $p_1$ starts when more than one token is available in each of its inputs, $w_1$ and $k_1$. We assume that enough material $m$ is always available. When $p_1$ finishes, a token will be added into each of $b_1$ and $w_1$. The process $p_2$ starts when more than one token is available in each of its inputs $w_2$, $k_2$ and $b_1$. A token is produced in $k_1$ and $p_2$ when $p_2$ starts. The outputs of $p_2$ are $b_2$ and $w_2$. The processes $p_3$ and $p_4$ work similarly.

Figure 4 shows the same serial production line governed by Base-stock. The first process $p_1$ starts when more than one token is available in each of its inputs, $w_1$ and $C_1$. When it finishes, a token is added into each of $b_1$ and $w_1$. The processes $p_2$ and $p_3$ work similarly. However, the process $p_4$ starts when more than one token exist in each of $b_3$ and $w_4$. It outputs one token into each of $C_1$, $C_2$ and $C_3$ at commencement. When it finishes, one token is added into $w_4$ and $b$.  

![Figure 3. A serial line with KANBAN](image-url)
In the following, we show an example of a serial production line controlled by KANBAN and Base-stock, with respective state transition tables.

Case **KANBAN.** Table 2 gives the state transition table for the production process shown in Figure 3. Initial inventory for every part is set to 0, and initial cards are set as $k_1 = k_2 = 1$ and $k_2 = 2$. Each of $p_1$, $p_3$ and $p_4$ has one actor, while $p_2$ has 2. The system shows a periodic behavior every 12 minutes. Both circuits $p_2 w_2 p_2$ and $p_4 w_4 p_4$ are critical, with maximum cycle mean $\lambda = 6$. Each activity starts twice in a period. The throughput is $2/12$, and the system WIP is equal to 5.83. It can be verified that the number of system WIP is minimum to attain the throughput $2/12$.

Case **Base-stock.** The state transition table for the same process under Base-stock is given in Table 3. Initial cards are set as $C_1 = 5$, $C_2 = 4$ and $C_3 = 2$. Initial inventories as well as the respective number of actors are the same as the previous case. Each activity starts twice in
a period. Circuits $p_2w_1p_2$ and $p_4w_4p_4$ are critical with $\lambda = 6$. The period is 12 minutes, the throughput is $2/12$, and the system WIP is 6.17, which is the minimum value to attain the throughput.

Case **CONWIP**. The state transition table for the same process under CONWIP has been given in Table 1. Four cards are initially assigned in the system. Initial inventories as well as the respective number of actors are the same as the above cases. Circuits $p_2w_2p_2$ and $p_4w_4p_4$ are critical with $\lambda = 6$. The period is 12 minutes, the throughput is $2/12$, and the system WIP is 5.83, which is the minimum value to attain the throughput.

### Table 2. State transition of KANBAN for a period

<table>
<thead>
<tr>
<th>time</th>
<th>k1</th>
<th>p1</th>
<th>w1</th>
<th>b1</th>
<th>k2</th>
<th>p2</th>
<th>w2</th>
<th>b2</th>
<th>k3</th>
<th>p3</th>
<th>w3</th>
<th>b3</th>
<th>p4</th>
<th>w4</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>186</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1(9),1(1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>187</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(8),1(12)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>0</td>
<td>1(6)</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>190</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1(5),1(9)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>191</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1(4),1(8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>1(2)</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>193</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1(2),1(6)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>---</td>
<td>1</td>
<td>0</td>
<td>1(6)</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>195</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(12),1(4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>198</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1(9),1(1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>29</td>
</tr>
</tbody>
</table>

### Table 3. State transition of Base-stock for a period

<table>
<thead>
<tr>
<th>time</th>
<th>C1</th>
<th>p1</th>
<th>w1</th>
<th>b1</th>
<th>C2</th>
<th>p2</th>
<th>w2</th>
<th>b2</th>
<th>C3</th>
<th>p3</th>
<th>w3</th>
<th>b3</th>
<th>p4</th>
<th>w4</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>183</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(12),1(4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>184</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1(11),1(3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>187</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(8),1(12)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>0</td>
<td>1(6)</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>190</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1(5),1(9)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>191</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1(4),1(8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>---</td>
<td>1</td>
<td>1</td>
<td>1(2)</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>193</td>
<td>0</td>
<td>1(3)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1(2),1(6)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>---</td>
<td>1</td>
<td>0</td>
<td>1(6)</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>195</td>
<td>0</td>
<td>1(1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(12),1(4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>0</td>
<td>1(4)</td>
<td>0</td>
<td>29</td>
</tr>
</tbody>
</table>
In the Base-stock case, for example, we show how to calculate the system WIP by using the state transition table. Consider Table 3. By observing the state transition table for a period, at time 183, five tokens (all are being processed) remain in the system for 1 minute (184-183=1). At the next event time (i.e. 184), six tokens remain in the system for 3 minutes (187-184=3). Similarly, 5, 6, 7 and 8 tokens remain in the system for the next 3, 1, 2, and 2 minutes, respectively. This yields \( (1*5)+(3*6)+(3*5)+(1*6)+(2*7)+(2*8)=74 \) as the total holding and waiting times for a period. Since the period is 12 minutes, the system WIP is \( 74/12=6.17 \) pieces.

In the both CONWIP and KANBAN cases, the optimum system WIPs to attain the same level of throughput are the same. In Proposition 1, we show that this statement holds true when the same total number of cards is employed in the both systems. However, in Base-Stock case, the optimum system WIP to attain the same level of throughput is larger than that in both CONWIP and KANBAN, because as we will show in Proposition 2, \( B - K = B - N = 11 - 4 > \frac{h_2 + 2h_3}{\lambda} = \frac{12 + 2(4)}{6} \), where \( B, K \) and \( N \) are the total number of cards in Base-stock, KANBAN and CONWIP, respectively, and \( h_i (i = 2, 3) \) is the holding time of process \( p_i \).

**Proposition 1.** Consider a serial production process with CONWIP and KANBAN. Let \( N \)
and $K$ be the total number of cards in CONWIP and KANBAN, respectively. Then, we have the following.

(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:** A proof is given in Sato and Khojasteh-Ghamari (2008).

**Proposition 2.** Consider the serial production process shown in Figures 2, 3 and 4 with CONWIP, KANBAN and Base-stock, respectively. Let $N$, $K$ and $B$ be the total number of cards in CONWIP, KANBAN and Base-stock, respectively. Then, we have the following.

(i) if $B - N \leq \frac{h_i + 2h_3}{\lambda}$, then $W_B \leq W_C$,

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

(iii) if $B - K \leq \frac{h_i + 2h_3}{\lambda}$, then $W_B \leq W_K$,

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

where $h_i$ ($i=2, 3$) is the processing time at workstation $i$, and $W_C$, $W_K$ and $W_B$ are the average WIP for CONWIP, KANBAN and Base-stock, respectively.

**Proof:** Consider Figures 2, 3 and 4. Let the outmost circuits in the both CONWIP and
Base-stock be $C$ and $B$, respectively. Also, let $K$ be the set of all elementary circuits in KANBAN. Apply the Little’s law on each of $C$, $B$ and $K$, and compare the system WIP of each pair. This completes the proof.

4. Conclusion

By employing the theory of token transaction systems, this paper showed how the three indices represented in Little's law (average WIP, average cycle time, and average throughput) are decided by the structure of a production process with control-cards and deployment of WIP. In order to apply the framework, the target production control systems should have deterministic processing time, and connecting queues with FIFO control policy. Since such token transaction systems show periodic behavior, we can design dynamic properties of production processes, which are related to the Little's law.

We showed that CONWIP is superior to KANBAN, when 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 throughput by deploying suitable number of cards. When the both systems have the same number of cards, they have the same performance. The superiority between Base-stock and KANBAN/CONWIP depends on the
number of cards employed in the systems. As shown in Proposition 2, appropriate design of
the whole system decides the superior one in certain situation.

References

-- An algebra for discrete event systems*. John Wiley.

floor control policies. *Manufacturing and Service Operations Management Conference


Gstettner, S. and Kuhn, H., 1996. Analysis of production control systems kanban and


Spearman, M.L. and Zazanis, M.A., 1992. Push and pull production systems: issues and


