International Journal of Management Science
Vol 6, No 1, May 2000

Asynchronous Waste: An Alternative Performance Measure for Pull Production Control Systems*

Il-Hyung Kim

Associate professor of School of Business Administration, Ajou University
San 5 Wonchon-Dong, Paldal-Gu, Suwon City, 442-749, Korea

(Received February 2000 , revision received April 2000)

ABSTRACT

An important objective of pull-based production control is to achieve synchronized and smooth production flow in a multi-stage system that is subject to uncertainty. To our knowledge, previous research has not generated a performance measure that captures this objective of pull-based production control systems. In this paper, we present an alternative performance measure for pull-based production control systems. This performance measure is called *asynchronous waste* which is the total expected earliness and lateness of the material with respect to the instant when the operation is required. We examine the issue of asynchronous waste in a two-stage kanban control system.

1. INTRODUCTION

Pull-based production control systems have received considerable attention from practitioners and researchers during the last decade because of their superior performance over push-based production control systems. Since pull-based systems trigger production at the time needed and in quantity required, they have less congestion and are easier to control [19]. It is often reported that the successful implementation of such systems has greatly reduced both inventory level and lead time [7].

Recently, researchers have developed formal models to analyze the performance of pull-based production control systems. The most commonly used performance measures for analyzing these systems can be classified into the following four categories: (a) capacity related such as throughput rate, (b) inventory

* This research is partially supported by BK 21 Supply Chain Management Team at Ajou University.

related such as work-in-process and finished goods inventories, (c) customer service related such as backorders, fill-rate and customer waiting time, and (d) lead time related such as production lead time and cycle time [3]. Most models have combined capacity related measures with inventory related measures; i.e., maximizing throughput rate for a given inventory level or minimizing average inventories for a given throughput rate [2, 5, 12, 13, 20, 21]. Others have considered the trade-off between the inventory level and the service level; i.e., minimizing average inventories while satisfying a certain service level (in terms of the average number of backorders, the average waiting time of customers, or the percentage of customers backordered) or minimizing the sum of inventory holding cost and backorder cost using some estimated cost coefficients [9, 18, 20, 23]. Some others have focused solely on the lead time related measures which are used in practice at Toyota Motor company to determine the number of kanban cards required in the system [15, 16]. Some non-optimization models have examined individual performance measures separately, instead of integrating multiple measures, to analyze the effects of the system parameters on the performance of the system [1, 4, 11].

An important objective of pull-based production control is to achieve synchronized and smooth production flow in a multi-stage system that is subject to uncertainty [14]. To our knowledge, previous research has not generated a performance measure that sufficiently captures this objective of pull-based production control systems. As indicated earlier, most models in the literature utilize performance measures such as throughput rate and work-in-process inventory. However, these measures, on their own, do not capture the performance of pull-based systems. For example, performance measures based on throughput rate are appropriate for traditional push-based manufacturing environment characterized by less-competitive and stable markets where a firm may sell as much as it can produce and where conventional unit cost-based accounting measures apply. If an objective of a pull-based control system is to maximize the throughput rate, the system should be considered to be inefficient whenever it is idle, which is not consistent with the basic philosophy of pull-based control systems. Consider another example where the objective is to minimize work-in-process inventory. Since this objective is unbounded, most models have incorporated certain arbitrary constraints such as exogenously-required minimum service levels or throughput rates. Actually, low level of work-in-process inventory is not an objective of pull-based control systems but a consequence of implementing such systems. These observations motivated us to develop an integrated measure that captures an important objective of pull-based production control systems.

Our new measure - *asynchronous waste*, can be described as follows: Consider

an instant when a particular operation may need material from a preceding station in which no material is available. This situation causes the operation to be delayed, an event we call *lateness*. Consider a different situation; an operation may not require certain material from a preceding station in which the material is already available. In this situation, the material waits at the in-buffer of the station, an event we call *earliness*. Asynchronous waste is simply the total expected earliness and lateness of the system, and it can be considered as an integrated measure in the following senses: The impact of earliness on the system is holding stock on hand, hiding some production and quality problems, and reducing flexibility to market fluctuations. Although the waste associated with lateness is often less emphasized than the waste associated with earliness, it may significantly affect downstream operations especially when the traffic intensity of the system is high The impact of lateness on the system is delaying subsequent operations, incurring opportunity loss, and forcing other parts or materials to wait in assembly operations.

In this paper, we examine the issue of asynchronous waste in a kanban control system (the most commonly-used pull-based production control system). The paper is organized as follows: We define the waste associated with earliness and lateness in a kanban control system and present a way to measure such waste in section 2. In section 3, we provide a two-stage model in which the inter-related effects of earliness and lateness are analyzed. In addition, an approximation scheme is presented in order to develop expressions for the performance measures. In section 4, we report some computational experiments that analyze the behavior of the system. Finally we conclude the paper with some future research in this area.

2. ASYNCHRONOUS WASTE IN A KANBAN SYSTEM

We now define the waste associated with earliness and lateness in the context of a kanban control system. In this system, the production order can be triggered only by the kanban card. Thus, the arrival of a kanban card at a manufacturing facility should be one of the necessary conditions for the operation. However, if the manufacturing facility is busy, then the kanban card has to wait until the facility becomes idle. Therefore, the arrival of a kanban card alone is not sufficient to trigger production. In order to trigger production, the facility has to be idle (ready to produce) and at least one kanban card is at the facility. Earliness

and lateness of the material can be defined with respect to the instant when the operation is required.

Table 1 describes eight possible events that may occur at an instant for a given operation keeping in view of the status of the station and material availability. The waste associated with earliness can be measured in terms of a period of time that the material has to wait until it is required. The waste associated with lateness can be measured in terms of a period of time that the operation has to wait until the material is available.

Table 1. Asynchronous waste in a kanban control system

status of the station	*material availability*	
	available	not available
1. kanban card arrived, facility idle	synchronized	lateness
2. kanban card arrived, facility busy	earliness	synchronized
3. kanban card not arrived, facility idle	earliness	synchronized
4. kanban card not arrived, facility busy	earliness	synchronized

As an initial step to analyze asynchronous waste in a pull-based production control system, we shall restrict our attention to a two-stage kanban system. In this simple setting, we analyze the impact of the various system parameters on the asynchronous waste and examine the behavior of the optimal kanban system In addition, we compare the performance of a conventional formulation, which maximizes throughput, to the performance of our formulation, which minimizes asynchronous waste.

3. THE MODEL: A TWO-STAGE KANBAN SYSTEM

3.1 Description of the Model

In this paper, we consider a two-stage kanban control system which enables us to analyze the interactions between preceding and succeeding stages while we can enjoy the mathematical tractability. As shown in Figure 1, each stage consists of a manufacturing facility (MF), two types of inventory buffer (IB. OB), and a kanban board (KB). We assume that there is an infinite and instant (no lead time) supply of raw materials in front of stage one. Demand arrives in single units. Inter-arrival times of the demand and the processing times of each stage are as-

sumed to be independent random variables. In stage i there is a fixed number of kanban cards, K_i. Production at stage i is carried out in a batch with size Q_i. We assume that the set-up time is incurred when the processor begins its production for each order (batch). The batch size of stage one is assumed to be an integer multiple of the batch size of stage two; i.e., $Q_1 = mQ_2, m \in I^+$. This assumption is quite reasonable because having m less than one increases the number of set-ups and the number of units waiting at the upstream stage, and aggravates asynchronous behavior accordingly. This is exploited in deterministic multi-stage systems in the form of nested policies [17]. The finished products at stage i are stored in the containers, each of the containers holds exactly Q_i units. There is a kanban card attached to each container. Hence, the number of the containers at stage i is same as the number of kanban cards at stage i, K_i.

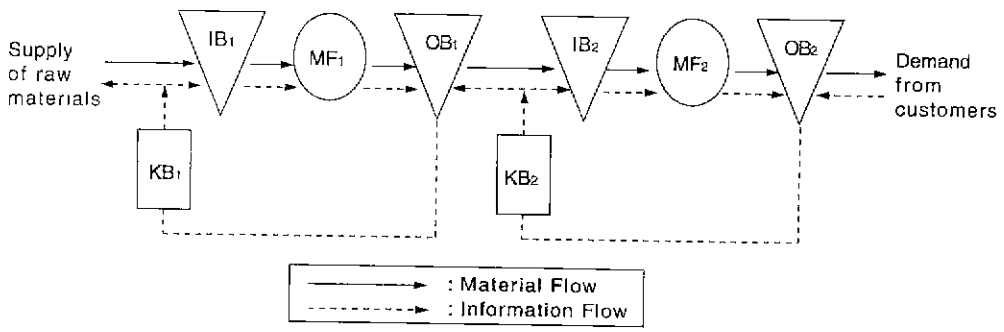

Figure 1 Schematic representation of a two-stage kanban system

The kanban system operates in the following way (our mechanism is similar to that of [12]): Whenever Q_i units are depleted from a container in out-buffer i, the corresponding kanban card is detached from the empty container and is transported to kanban board i located in front of stage i. At this point, there are two possible courses of action depending on the state of out-buffer $i-1$ (out-buffer "0" means the supply of raw materials): (a) If out-buffer $i-1$ is empty, then the kanban card has to wait at kanban board i. (b) Otherwise, Q_i units are depleted from the out-buffer and transported with the kanban card to in-buffer i. The items transported to in-buffer i are processed at processor i on a first-come-first-served basis. Once the processor produces Q_i units, the kanban card which ordered the full container is attached to the container and is sent to out-buffer i. In the event that a customer places an order and there is no finished

goods inventory available in out-buffer two. we assume that this customer is willing to wait until the finished goods become available.

There are five important observations regarding the kanban system described above. First, there are K_i kanban cards circulating at stage i at any point in time. Hence, the maximum inventory level in out-buffer i is equal to K_iQ_i. Second, a kanban card is sent to the processor whenever a full container (Q_i units) is depleted. Therefore, both the number of kanban cards K_i and the container size dictate the arrival process of the kanban cards at processor i. Third, the container size Q_i affects the traffic intensity of stage i. Smaller batches cause the workload on the facility to increase due to increased number of set-ups. As the batch size increases, the traffic intensity decreases and the effect of set-up times diminishes. Fourth, the number of kanban cards at stage one, K_1, affects not only the inventory level at the out-buffer of stage one but also the time that the kanban cards wait at the kanban board of stage two. This waiting time may cause starvation of the processor at stage two, which in turn increases the traffic intensity of the stage. Finally, the number of kanban cards at stage two, K_2, affects the inventory level at the out-buffer and the number of backorders. It also affects the time that the kanban cards wait at the kanban board since it restricts the maximum number of kanban cards waiting at the kanban board.

3.2 Analysis: Determining Asynchronous Waste

Even for a two-stage system, the exact expressions for the performance measures are mathematically intractable. This is because of blocking and starvation that may occur at the place where the two stages join together. To obtain approximated expressions for the performance measures, first we shall decompose the system into two independent subsystems by ignoring the impact of the blocking and the starvation. For each of the two subsystems we develop expressions for the performance measures. Then, we shall consider interactions between the two stages.

3.2.1 Decomposition: An Independent Kanban Cell

Under the decomposition scheme no kanban card needs to wait at the kanban board of stage two. Thus, the state of each independent kanban system can be specified by (a) the number of items waiting at the in-buffer, (b) the number of finished items in the out-buffer, and (c) the number of backorders. Let TI_i be the time for an item waiting at the in-buffer of stage i. We shall refer TI_i as the *in-buffer lead time* hereafter. Let TO_i be the time for an item waiting at the out-

buffer of stage i. We shall refer TO_i as the *out-buffer lead time* hereafter. Let TB_i denote the *order response time*: i.e., the time that a demand (kanban card) from a customer (succeeding stage) waits for an order.

In order to make the model tractable, we further assume that the arrival process of the demand (customer order) is a Poisson process and the time needed to produce a batch including a set-up time at each stage is exponentially distributed. In addition, the following notation is used in our model:

K_i : number of kanban cards at stage i
Q_i : size of the container at stage i
d : demand rate of the product
λ_i : arrival rate of kanban cards (batches) at the processor of stage $i = d/Q_i$
τ_i : set-up time for each order (batch) at stage i
ρ_i : unit production rate of the processor at stage i
δ_i : mean time to process an order (batch) at stage $i = \tau_i + Q_i/p_i$
μ_i : processing rate at the processor of stage $i = 1/\delta_i = p_i/(p_i\tau_i + Q_i)$
ρ_i : traffic intensity of stage $i = \lambda_i/\mu_i = d\tau_i/Q_i + d/p_i$
m : ratio of batch size of stage one to that of stage two, $m \in I^+$
r_i : root of the characteristic equation of $E_{Q_i}/M/1$ queue at stage i
NI_i : number of items waiting at the in-buffer of stage i
NO_i : number of items waiting at the out-buffer of stage i
NB_i : number of backorders at stage i

We now turn our attention to developing expressions for TI, TO, and TB for each stage. Since a kanban card (that triggers production) arrives at the kanban board after Q_i units are depleted at the out-buffer, the inter-arrival times of the order at stage i have an Erlang distribution of order Q_i and mean Q_i/d. In this case, stage i can be modeled as an $E_{Q_i}/M/1$ queuing system. For stage two, we have the following expressions for the performance measures (Readers are referred to the Appendix A for details):

$$\overline{TI}_2 = \frac{r_2^{Q_2+1}(1-r_2^{(K_2-1)Q_2})}{d(1-r_2)} \tag{1}$$

$$\overline{TO}_2 = \frac{K_2Q_2}{d} - \frac{Q_2-1}{2d} - \frac{r_2(1-\rho_2 r_2^{(K-2-1)Q_2})}{d(1-r_2^{Q_2})} \tag{2}$$

$$\overline{TB_2} = \frac{\rho_2 r_2^{(K_2-1)Q_2+1}}{d(1-r_2)} \tag{3}$$

For stage one, we have the following expressions (Readers are referred to the Appendix B for details):

$$\overline{TI_1} = \frac{r_1^{mQ_2+1}(1-r_2^{(K_1-1)mQ_2})}{d(1-r_1)} \tag{4}$$

$$\overline{TO_1} = \frac{K_2 mQ_2}{d} - \frac{(m-1)Q_2}{2d} - \frac{r_1}{d(1-r_1)} + \frac{\rho_1 Q_2 r_1^{((K_1-1)m+1)Q_2}}{d(1-r_1^{Q_2})} \tag{5}$$

$$\overline{TB_1} = \frac{Q_2\rho_1 r_1^{((K_1-1)m+1)Q_2}(1-r_1^{K_2Q_2})}{d(1-r_1^{Q_2})} \tag{6}$$

3.2.2 Asynchronous Waste of the Complete System

Asynchronous waste of the system may occur at the places where the two stages join together and they join supply or demand. At the place where stage one joins supply there will be no waste associated with earliness or lateness since we assume infinite and instant supply of raw materials. At the place where the two stages join together both types of waste may occur. To analyze the waste we shall examine an instant when a kanban card arrives at the kanban board of stage two. Let X be the number of kanban cards waiting at the kanban board when the card arrives. If the number of kanban cards circulating at stage two is fairly large, finished items at stage one would not be blocked by stage two and would be immediately transported to the out-buffer of stage two. In this case, X is identical to the number of backorders that would occur at stage one of the decomposed subsystems. Thus, the probability distribution associated with X when K_2 goes to infinity is given by:

$$a_x \equiv \lim_{K_2\to\infty} P(X=x) = \begin{cases} \rho_1(1-r_1^{Q_2})r_1^{((K_1-1)\,m+x)Q_2} & \text{if } x>0 \\ \rho_1(1-r_1^{Q_2})r_1^{(K_1-1)\,mQ_2} & \text{if } x=0^+ (\text{when } OB_1=0) \\ 1-\rho_1 r_1^{(K_1-1)\,mQ_2} & \text{if } x=0^- (\text{when } OB_1=0) \end{cases} \tag{7}$$

Note that, in equation (7), the probability that X will be zero is divided into two cases: the case when the out-buffer of stage one is empty ($x=0^+$) and the case when it is not empty ($x=0^-$). Thus, the probability that the arriving kanban card

will be immediately served from the out-buffer of stage one is determined by $1-\sum_{x=0}^{\infty} a_x$.

Now, consider the impact of blocking which may occur when the number of kanban cards circulating at stage two is limited by K_2. In this case, X can not be larger than K_2-1. When $X=K_2+1+j,\ j\geq 0$ in equation (7), the number of kanban cards actually waiting at the kanban board is K_2-1, and the remaining $j+1$ kanban cards are waiting at the out-buffer of stage two as backorders (The actual number of backorders is $j+1$ multiplied by Q_2 or $(j+1)Q_2$). In order to make the kanban card which, under the assumption of unlimited K_2, sees K_2+j cards actually arrive at the kanban board, $j+1$ kanban cards should be processed at stage two. By considering all possible cases which may occur during the period of processing $j+1$ kanban cards, we can assign the probability of the cases when X is greater than or equal to K_2 into the cases when X is less than K_2. Thus, the probability distribution associated with X is given by:

$$a_x \equiv P(X=x) = a_x + \sum_{k=k_0}^{\infty} P(x+km \rightarrow x)\cdot a_{x+km} \tag{8}$$

where $P(x+km\rightarrow x)$ is the transition probability from state $(x+km)$ to state x (Readers are referred to Appendix C for details about determining the transition probability).

Next, we shall examine an instant when a kanban card departs from the out-buffer of stage two. Let Y be the number of kanban cards remaining at the out-buffer of stage two when the card departs, and b_y be its departure rate when $Y=y$. By considering the limiting probability of each state at the out-buffer of stage two, we can determine b_y as follows (For the definition of the state and the detailed calculation of the limiting probabilities, readers are referred to Appendix A and Figure 8):

$$b_k = \begin{cases} dP_{(K_2-1)Q_2+1} \\ dP_{yQ_2+1} \\ dP_1+\sum_{i=0}^{-\infty}\mu P_{i-Q_2} \end{cases} = \begin{cases} \dfrac{d}{Q_2}(1-r_2^{Q_2}) & \text{if } y=K_2-1 \\ d\rho_2(1-r_2)r_2^{(K_2-1-y)Q_2-1} & \text{if } 1\leq y<K_2-1 \\ \rho_2 r_2^{(K_2-1)Q_2-1}(d(1-r_2)+\mu r_2) & \text{if } y=0 \end{cases} \tag{9}$$

The probability associated with Y can be determined by:

$$\beta_y \equiv P(Y = y) = \frac{b_y}{\sum_{k=0}^{K_2-1} b_k} \tag{10}$$

Consider the waste associated with earliness and lateness occurring at the place where the two stages join together. When a kanban card arrives at the kanban board of stage two, there are two possibilities: It will be either immediately served from the out-buffer of stage one or waiting at the kanban board until the next item is available. (a) If it is immediately served, then Q_2 units are transported from the out-buffer of stage one to the in-buffer of stage two. At the in-buffer, these items will be either immediately processed or waiting at the in-buffer depending on the status of the processor. In both cases, the waste associated with earliness occurs. This waste can be determined by the sum of the waiting time at the out-buffer of stage one and the waiting time at the in-buffer of stage two. (b) If the kanban card which arrives at the kanban board of stage two has to wait (It is the case of out of stock at the out-buffer of stage one), either type of the waste may occur. It depends on the status of the processor. Suppose $X = x, 0 \le x \le K_2 - 1$ and $Y = y, 0 \le y \le K_2 - x - 1$ when the kanban card arrives at the kanban board of stage two. In this case, the number of orders (kanban cards) waiting at the in-buffer and being processed at stage two is $K_2 - x - y - 1$. When a batch is finished at stage one, at most m kanban cards which wait at the kanban board will receive the finished items. Thus, the number of batches that should be finished at stage one before the kanban card under consideration (the card which has just arrived at the kanban board of stage two) receives the finished items, n_Q, is

$$n_Q = \left\lceil \frac{x+1}{m} \right\rceil \tag{11}$$

where $\lceil (x+1)/m \rceil$ is the smallest integer which is greater than or equal to $(x+1)/m$. From the first to the $(n_Q - 1)^{th}$ batch, each batch will serve m kanban cards, and the $n_Q{}^{th}$ batch will serve n_R kanban cards, where

$$n_R = x + 1 - (n_Q - 1)\, m \tag{12}$$

When the $n_Q{}^{th}$ batch is finished, two courses of action are possible depend-

ing on the status of the processor. If the processor is busy, then the finished items are still not late. The waste associated with earliness occurs in this case, and it is measured by the sum of remaining processing time of the items which are currently being processed at stage two and the processing time of the items waiting at the in-buffer of stage two. If the processor is idle, the waste associated with lateness occurs and it is measured by the waiting time of the processor for the orders (idle time of the processor) after the kanban card under consideration arrives at the kanban board. However, if n_R is greater than one, the kanban card under consideration will be transported to the in-buffer together with $n_R - 1$ kanban cards in front of it. In this case, only the waste associated with earliness occurs regardless of the status of the processor.

To develop expressions for the waste, we assume that the processing time distribution of n batches at stage two is approximated by an exponential distribution (Note that the actual distribution is an Erlang distribution of order n). Let $RT(n)$ be the sum, measured at an instant when the n^{th} batch is finished at stage one, of the remaining processing time of the items which are processed at stage two and the processing time of the items waiting at the in-buffer of stage two. The above approximation enables us to determine the expected value of $RT(n)$ using the following recursive formula:

$$\begin{aligned} E(RT(n)) &= E(RT(n-1) + \text{processing time of } m \text{ batches at stage two} \\ &\quad - \text{processing time of } n^{th} \text{ batch at stage one})^+ \\ &= \frac{(E(RT(n-1)) + m\delta_2)^2}{\delta_1 + E(RT(n-1)) + m\delta_2}, \qquad 2 \le n \le n_Q \end{aligned} \tag{13}$$

$$E(RT(1)) = \frac{((K_2 - x - y - 1)\,\delta_2)^2}{\delta_1 + (K_2 - x - y - 1)\,\delta_2} \tag{14}$$

The waste associated with earliness and lateness can be determined as follows:

$$\begin{aligned} WE_{12} &= (1 - P(\text{out of stock at } OB_1))\, E\,(TO_1 + TI_2 \mid \text{out of stock at } OB_1) \\ &\quad + P(\text{out of stock at } OB_1)\, E\,(RT(n_Q) + \text{processing time of} \\ &\quad (n_R - 1) \text{ batches at stage two}) \\ &= \overline{TO}_1 + \Big(1 - \sum_{x=0^+}^{K_2-1} \alpha_x\Big)\,\overline{TI}_2 \\ &\quad + \sum_{x=0^+}^{K_2-1} \alpha_x \sum_{y=0^+}^{K_2-x-1} \frac{\beta_y}{\sum_{k=0}^{K_2-x-1} \beta_k} \left(E(RT(n_Q)) + (n_R - 1)\delta_2\right) \end{aligned} \tag{15}$$

$$WL_{12} = P(\text{out of stock at } OB_1)\, E(\text{processing time of } n_Q^{th} \text{ batch at stage one } - RT(n_Q - 1) - \text{processing time of } m \text{ batches at stage two})^{-}$$

$$= \sum_{x \in \{x \mid n_{\beta}=1, 0^{+} \le x \le K_2 - 1\}} \alpha_x \sum_{y=0}^{K_2 - x - 1} \frac{\beta_y}{\sum_{k=0}^{K_2 - x - 1} \beta_k} \left(\frac{\delta_1^2}{\delta_1 + E(RT(n_Q - 1)) + m\delta_2} \right) \quad (16)$$

Note that β_y is normalized for each x such that all the possible β_y, $0 \le y \le K_2 - x - 1$, are summed to 1.

Among the above two types of waste, it can be easily understood that the waste associated with lateness, WL_{12}, has certain impact on the performance of stage two while the waste associated with earliness, WE_{12}, has no impact on the performance. Specifically, WL_{12} increases the idle time of the processor at stage two, which in turn increases virtual traffic intensity of stage two. One simple way to capture this impact is to consider WL_{12} as an additional time to process a batch at stage two. By invoking this simple approximation, we determine the waste occurring at the place where stage two joins demand as follows:

$$WE_{2D} = \overline{TO}_2 \text{ when mean processing time is } \delta_2^M \quad (17)$$

$$WL_{2D} = \overline{TB}_2 \text{ when mean processing time is } \delta_2^M \quad (18)$$

where $\delta_2^M = \delta_2 + WL_{12}$.

3.3 Analysis: Finding an Efficient Kanban System

In order to determine an efficient kanban system (specified by the number of kanban cards and the size of the container at each stage) that minimizes the total asynchronous waste, we formulate the following mathematical program:

$$(\text{P1}) \quad \min_{Q_i, K_i} \quad W = WE_{12} + WL_{12} + WE_{2D} + WL_{2D} \quad (19)$$

$$s.t. \quad Q_1 = mQ_2, \ m \in I^{+} \quad (20)$$

$$K_i \ge 1, \ i = 1, 2 \quad (21)$$

$$Q_i > \frac{d\tau_i}{1 - d/p_i} \equiv Q_i^{LB}, \ i = 1, 2 \quad (22)$$

Notice that the constraint (22) ensures that the traffic intensity of each stage is

less than one [8].

In problem (P1), we treat all four types of waste in a single horizon. In other words, we consider all of them as similar types of performance measures which can be added together with same amount of weight. However, in some situations, we may need to treat WL_{2D} differently. Specifically, WL_{2D} can be considered as an external performance measure which can be observed by the customers while the other three can be considered as an internal performance measure that monitors the efficiency of the system. One way to handle this case is to formulate the problem in a way that the trade-off between the two (internal and external) performance measures can be analyzed. In this paper, we consider the demand as an order from some other downstream operations or facilities. Thus, WL_{2D} is interpreted as a part of the internal performance, and we limit our analysis to problem (P1).

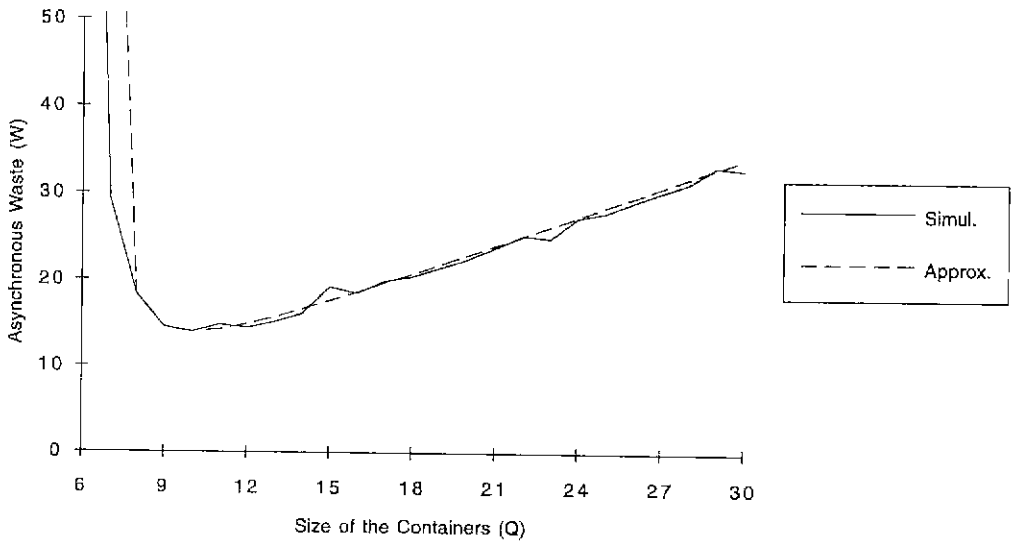

Figure 2. Asynchronous waste vs. the size of the container for the case when $d=8$, $p_1=p_2=10$, $\tau_1=\tau_2=0.1$, $K_1=8$, $K_2=4$, $m=1$

It can be easily seen from equations (15), (16), (17), and (18) that the objective function of (P1) is a complicated function of our decision variables. Since developing an efficient algorithm to solve the problem (P1) is not a main objective of this paper, we use an exhaustive enumeration approach to find a near-optimal solution by incrementing the decision variables from their minimum values to a fairly large number. We analyze, by means of numerical experiments, the sensi-

tivity of system performance with respect to changes in system parameters. Finally, we compare the performance of a conventional formulation, which maximizes throughput, to the performance of our formulation, which minimizes asynchronous waste.

4. NUMERICAL EXPERIMENTS

In this section, we report some results of numerical experiments whose purpose is twofold: (a) to examine the accuracy of approximation discussed in section 3.2.2 and (b) to analyze the behavior of the kanban system. In our 2-stage model, there are five system parameters; i.e., the mean inter-arrival time of demand, the processing rate per unit at each stage, and the set-up time at each stage. We have designed different problems by varying system parameters as follows: 17 different cases by varying d from 1.0 to 9.0 incremented by 0.5, 11 different cases by varying p_i from 7.5 to 12.5 incremented by 0.5 for each stage, and 11 different cases by varying τ_i from 0.0 to 0.5 incremented by 0.05 for each stage.

We now examine the goodness of the approximation scheme in which we approximate the waste associated with earliness and lateness. Simulation is used to determine near-exact values. Figure 2 compares the asynchronous waste determined by the approximation scheme with the simulation results. In most cases, approximated values lie within 5% variations of the simulation results. It is apparent from the figure that the difference between approximated and exact values may become larger when the traffic intensity of the system is higher (the size of the container is smaller) or the number of kanban cards is smaller (figure not shown). However, it is observed that the general behavior of the system (shape of the curves shown in the figures) determined by the approximation scheme is very similar to the simulation results although the approximated values differ substantially from the simulation results in some cases.

Next, consider the impact of the size of the container (Q_i) and the number of kanban cards (K_i) on the asynchronous waste. As the size of the container (batch size) decreases, the traffic intensity of the system approaches one, leading to the rapid increase in waste associated with lateness. As the batch size increases, the scale effect becomes predominant, leading to the linear increase in waste associated with earliness. For the batch sizes in-between, the asynchronous waste goes through a minimum [8]. In general, the asynchronous waste has a sharp minimum and is sensitive to the choice of the batch size. On the other hand, the rela-

tionship of asynchronous waste with the number of kanban cards is somewhat different: The asynchronous waste has a flat minimum and is relatively insensitive to the choice of the number of kanban cards. When the number of kanban cards at the first stage is small, we can observe the steep changes in asynchronous waste. The decrease in the number of kanban cards at the first stage will delay processing at the second stage, which in turn causes a significant increase in asynchronous waste especially when the traffic intensity of the system is high. This is simply because stage one feeds stage two.

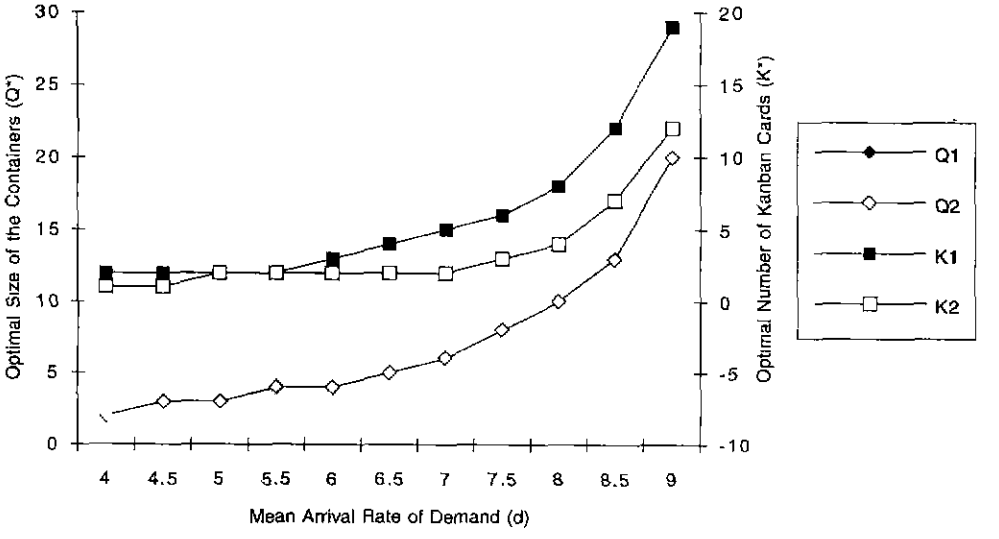

Figure 3. Optimal kanban system vs. the arrival rate of demand for the case when $p_1 = p_2 = 10$, $\tau_1 = \tau_2 = 0.1$

We now examine the sensitivity of the optimal kanban system to the system parameters. Figure 3 depicts the relationship of the kanban system with respect to the arrival rate of demand. The optimal size of the container at both stages increases with the demand rate, and they are more sensitive at higher demand rate. Note that some figures in the paper do not show points for Q_1 when Q_1 is identical to Q_2. Regarding the optimal number of kanban cards, it is observed that in the range of relatively low demand rate, the impact of increase in demand is absorbed by increasing the size of the container without increasing the number of kanban cards. In the range of high demand rate which makes the traffic intensity of the system close to one, the number of kanban cards at both stages increases with demand rate. The number of kanban cards required at stage one increases more rapidly than the number required at stage two in order to avoid

the waste associated with lateness, WL_{12}, which delays processing at stage two. When the traffic intensity is high, even a small delay may cause significant impact on the performance of the system.

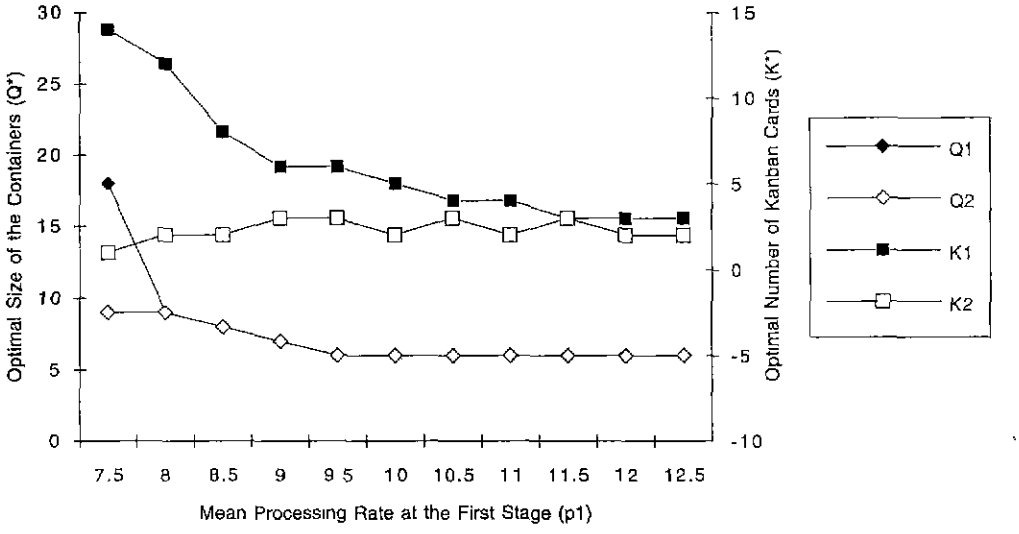

Figure 4. Optimal kanban system vs. the processing rate at stage one for the case when $d=7$, $p_2=10$, $\tau_1=\tau_2=0.1$

Figure 4 shows the relationship of the kanban system with respect to the processing rates at stage one. Since we vary the processing rate at one stage from 7.5 to 12.5 while setting the processing rate of the other stage to 10, the first half of the range (from 7.5 to 10.0) represents the case of changing the processing rate of a bottleneck machine and the second half (from 10.0 to 12.5) represents the case of a non-bottleneck machine. It is observed from our experiments that as the processing rate of a bottleneck machine increases, the optimal size of the container decreases at both stages. The processing rate of non-bottleneck machine has very little impact on the size of the container. The optimal number of kanban cards is insensitive to the processing rate of either bottleneck or non-bottleneck machines. However, when the traffic intensity of the system is high, the number of kanban cards required at stage one increases rapidly as the processing rate at the first stage decreases. This phenomenon is a mirror image of the previous case when the demand rate increases close to the processing rate, leading to high traffic intensity. It is also observed that the increase in the processing rate of a non-bottleneck machine can not improve the performance of the system in terms of the total asynchronous waste.

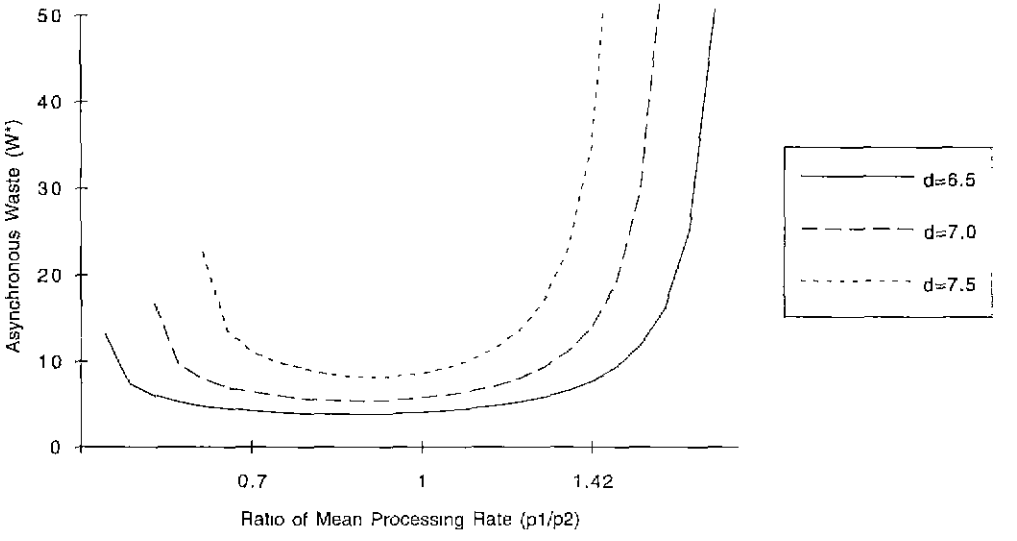

Figure 5. Asynchronous waste vs. ratio of mean processing rate when $p_1 + p_2 = 20$, $\tau_1 = \tau_2 = 0.1$

Figure 5 shows the results of the experiment in which we fix the sum of the mean processing rates of the two stages and vary the ratio of p_1 to p_2. We can observe highly skewed asymmetry on the allocation of resources. Although more intensive analysis is required to make a firm conclusion, it is suggested in this example that we need more capacity toward the end of the line to absorb higher stochastic variances at the downstream operations. A similar type of phenomenon has been shown in the simple experiment by Goldratt and Cox [6].

Figure 6 shows the behavior of the optimal kanban system with respect to the set-up times at stage two. We vary the set-up time at one stage from 0.0 to 0.5 while setting the set-up time at the other stage to 0.25. Thus, the first half of the range (from 0.0 to 0.25) represents the case of changing set-up time of a non-bottleneck machine and the second half (from 0.25 to 0.5) represents the case of a bottleneck machine. The optimal number of kanban cards is virtually independent of the set-up time at both stages. The size of the container increases with the set-up times of both bottleneck and non-bottleneck machines, and the rate of increase is gradual (approximately linear) rather than rapid (convex). The ratio of the container size of stage one to that of stage two, $Q_1 / Q_2 (= m)$, increases from one to two then to three as the set-up time of stage two decreases (or the ratio of the set-up time of stage one to that of stage two, τ_1 / τ_2, increases). It is also ob-

served that the decrease of the set-up time of non-bottleneck machines as well as bottleneck machines improves the performance of the system in terms of the total asynchronous waste.

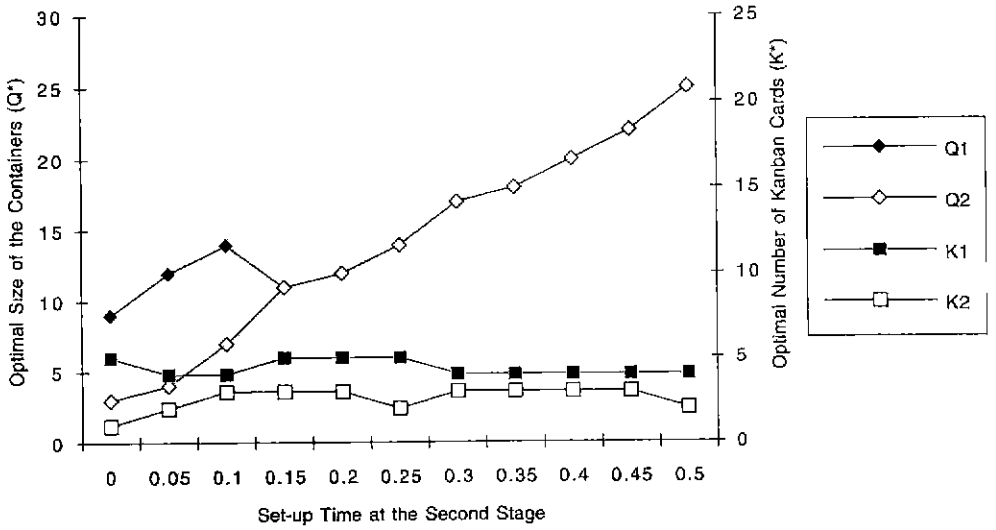

Figure 6. Optimal kanban system vs. the set-up time at stage two for the case when $d=7$, $p_1 = p_2 = 10$, $\tau_1 = 0.25$

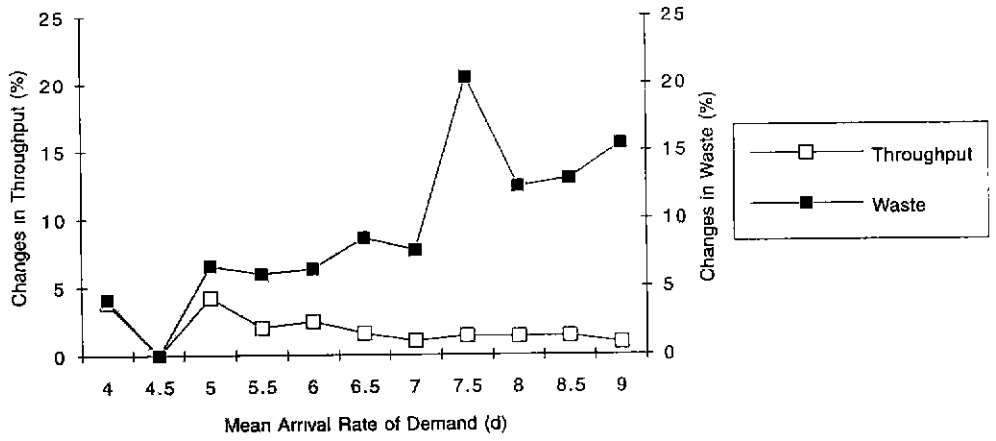

Figure 7. Changes in throughput rate and asynchronous waste when $p_1 = p_2 = 10$, $\tau_1 = \tau_2 = 0.1$

In summary, the optimal number of kanban cards is quite robust except the case of high traffic intensity. In this case, the number of kanban cards required at stage one increases rapidly as the demand rate increases or processing rate of

stage one decreases. The optimal size of the container is more sensitive to the changes of the system parameters, it increases rapidly (convex) with the traffic intensity of the system while it increases gradually (linear) with the set-up times.

Finally, we compare the performance of a conventional formulation, which maximizes throughput, to the performance of our formulation, which minimizes asynchronous waste. To compare the performance, we first determine the number of kanban cards (K_1^* / K_2^*) and the size of the container (Q_1^* / Q_2^*), which minimize the asynchronous waste. Then, we fix the total number of items circulating in the system $(Q_1^* / K_1^* + Q_2^* / K_2^*)$ and reallocate them to each stage such that the throughput of the system is maximized. In both formulations, we use an exhaustive enumeration approach to find near-optimal solutions. Figure 7 shows how the conventional formulation behaves differently from our formulation in terms of the two performance measures; asynchronous waste and throughput. Specifically, it shows the percentage changes (difference between the two formulations divided by the result of asynchronous waste formulation) realized by the reallocation with respect to the mean arrival rate of demand. When the demand rate is high (when the traffic intensity of the system is greater than 0.75), the conventional formulation increases the asynchronous waste by 10-20% while it increases the throughput only by less than 2%. In the above experiments, it is shown that minimizing asynchronous waste also results in a high throughput which is close to the theoretical maximum for a given number of items circulating in the system.

5. CONCLUSIONS

In this paper, we have presented asynchronous waste as an alternative performance measure for pull-based production control systems. This new performance measure is the total expected earliness and lateness of the material with respect to the instant when the operation is required. We have defined the waste associated with earliness and lateness in pull-based production control systems. We have shown how this new measure can be applied to a two-stage kanban control system. We have proposed an approximation scheme to find the mathematical expressions for the asynchronous waste of the system. The accuracy of approximation has been examined by simulation. The sensitivity of the kanban system with respect to changes in system parameters has been examined through numerical experiments.

While the model developed in this paper is limited, it offers some potential directions for more complicated situations such as multiple products, non exponential processing times, and more general demand processes.

REFERENCES

[1] BADINELLI, R. D., "A Model for Continuous-Review Pull Policies in Serial Inventory Systems," *Operations Research* 40 (1992), 142-156.

[2] BITRAN, G. R. and L. CHANG, "A Mathematical Programming Approach to a Deterministic Kanban System," *Management Science* 33 (1987), 427-442.

[3] BUZACOTT, J. A., "Queuing Models of Kanban and MRP Controlled Production Systems," *Engineering Costs and Production Economics* 17 (1989), 3-20.

[4] DELEERSNYDER, J., T. J. HODGSON, H. MULLER, and P. J. O'GRADY, "Kanban Controlled Pull Systems: An Analytic Approach," *Management Science* 35 (1989), 1079-1091.

[5] DENARDO, E. V. and C. S. TANG, "Linear Control of a Markov Production System," *Operations Research* 40 (1992), 259-278.

[6] GOLDRATT, E. M. and J. COX, *The Goal: A Process of Ongoing Improvement*, North River Press, Inc., New York.

[7] HAY, E. J., *The Just-In-Time Breakthrough - Implementing the New Manufacturing Basics*, John Wiley & Sons, Inc., New York 1988.

[8] KARMARKAR, U., "Lotsizes, Lead Times and In-Process Inventories," *Management Science* 33 (1987), 409-418.

[9] KARMARKAR, U. and S. KEKRE, "Batching Policy in Kanban Systems," *Journal of Manufacturing Systems* 8 (1989), 317-328.

[10] KLEINROCK, L., *QUEUEING SYSTEMS, Volume 1: Theory*, John Wiley & Sons, Inc., New York 1975.

[11] MASCOLO, M., Y. FREIN, Y. DALLERY, and R. DAVID, "A Unified Modeling of Kanban Systems Using Petri Nets," *The International Journal of Flexible Manufacturing Systems* 3 (1991), 275-307.

[12] MITRA, D. and I. MITRANI, "Analysis of A Kanban Discipline for Cell Coordination in Production Lines, I," *Management Science* 36 (1990), 1548-1566.

[13] MITRA, D. and I. MITRANI, "Analysis of A Kanban Discipline for Cell Coordination in Production Lines, II: Stochastic Demands," *Operations Research* 39 (1991), 807-823.

[14] OHNO, T., *Toyota Production System: beyond Large-Scale Production*, Pro-

ductivity Press, Cambridge 1998.

[15] PHILIPOOM, P., L. REES, B. TAYLOR III, and P. HUANG, "An Investigation of Factors Influencing the Number of Kanbans Required in the Implementation of the JIT Technique with Kanbans," *International Journal of Production Research* 25 (1987), 457-472.

[16] REES, L., P. PHILIPOOM, B. TAYLOR III, and P. HUANG, "Dynamically Adjusting the Number of Kanbans in a Just-In-Time Production System Using Estimated Values of Leadtime," *IIE Transactions* 19 (1987), 199-207.

[17] SCHWARZ, L. B., *Multi-Level Production/Inventory Control System: Theory and Practice,* North-Holland, Amsterdam 1981.

[18] SO, K. C. and S. C. PINAULT, "Allocating Buffer Storages in a Pull System," *International Journal of Production Research* 26 (1988), 1959-1980.

[19 SPEARMAN, M. and M. ZAZANIS, "Push and Pull Production Systems: Issues and Comparisons." *Operations research* 40 (1992), 521-532

[20] TANG, C. S., "The Impact of Uncertainty on a Production Line," *Management Science* 36 (1990), 1518-1531.

[21] TAYUR, S R., "Properties of Serial Kanban Lines," *Queueing Systems* 12 (1992), 297-318.

[22] TAYUR, S. R., "Structural Properties and a Heuristic for Kanban Controlled Serial Lines," *Management Science* 39 (1993), 1347-1368.

[23] WANG, H. and H. WANG, "Optimum Number of Kanbans between Two Adjacent Workstations in a JIT System," *International Journal of Production Economics* 22 (1991), 179-188.

Appendix A. Performance Measures of Stage Two

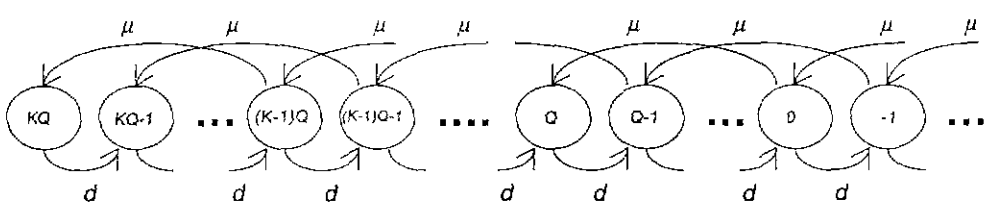

Figure 8. State transition diagram of stage two

The net inventory (on-hand - backorders) level at the out-buffer of stage two can be described by a Markovian model with the state transition diagram shown in Figure 8. The system is in state $k(k \geq 0)$ when the inventory level at the out-

buffer is equal to k. In addition, the system is in state $k(k<0)$ when the number of backorder is equal to $|k|$. For this Markovian model, we can derive expressions for the limiting probability, P_k, that a system will be in state k as follows [10]:

$$P_k = \begin{cases} \frac{1}{Q_2}(1-r_2^{K_2Q_2-k+1}) & \text{if } (K_2-1)Q_2 < k \le K_2Q_2 \\ \rho_2(1-r_2)r_2^{(K_2-1)Q_2-k} & \text{if } k \le (K_2-1)Q_2 \end{cases} \tag{23}$$

where r_2 is the root of the characteristic equation $r^{Q_2+1}-(1+\rho_2Q_2)r+\rho_2Q_2=0$. The total number of items at stage two is bounded by K_2Q_2. When $k \le 0$, there are backorders and the out-buffer is empty. Hence, $NI_2=(K_2-1)Q_2$. When $k>0$, where k satisfies $(j-1)Q_2 < k \le jQ_2, 1 \le j \le K_2-1$, it can be easily seen that there are j containers at the out-buffer. Thus, $NI_2=(K_2-j-1)Q_2$. When $(K_2-1)Q_2 < k \le K_2Q_2$, all containers are at the out-buffer and the in-buffer is empty. In summary, we have

$$NI_2 = \begin{cases} (K_2-1)Q_2 & \text{if } k \le 0 \\ (K_2-j-1)Q_2 & \text{if } (j-1)Q_2 < k \le jQ_2, 1 \le j \le K_2-1 \\ 0 & \text{if } (K_2-1)Q_2 < k \le K_2Q_2 \end{cases} \tag{24}$$

The expected number of items waiting at the in-buffer can be expressed as:

$$\overline{NI}_2 = \sum_{j=1}^{K_2-1} \sum_{k=(j-1)Q_2+1}^{jQ_2} (K_2-j-1)Q_2P_k + (K_2-1)Q_2 \sum_{k=0}^{-\infty} P_k \tag{25}$$

In this case,

$$\overline{TI}_2 = \frac{\overline{NI}_2}{d} = \frac{r_2^{Q_2+1}(1-r_2^{(K_2-1)Q_2}}{d(1-r_2)} \tag{26}$$

The expected on-hand inventory at the out-buffer can be expressed as:

$$\overline{NO}_2 = \sum_{k=0}^{K_2Q_2} k \cdot P_k \tag{27}$$

In this case,

$$\overline{TO_2} = \frac{\overline{NO_2}}{d} = \frac{K_2 Q_2}{d} - \frac{Q_2 - 1}{2d} - \frac{r_2(1 - p_2 r_2^{(K-2-1)Q_2})}{d(1 - r_2^{Q_2})} \tag{28}$$

Similarly, the expected backorder is given by $\overline{NB_2}$ and the expected order response time is given by $\overline{TB_2}$, where

$$\overline{NB_2} = \sum_{k=-1}^{-\infty} (-k) \cdot P_k$$

$$\overline{TB_2} = \frac{\overline{NB_2}}{d} = \frac{\rho_2 r_2^{(K_2-1)Q_2+1}}{d(1 - r_2)} \tag{29}$$

Appendix B. Performance Measures of Stage One

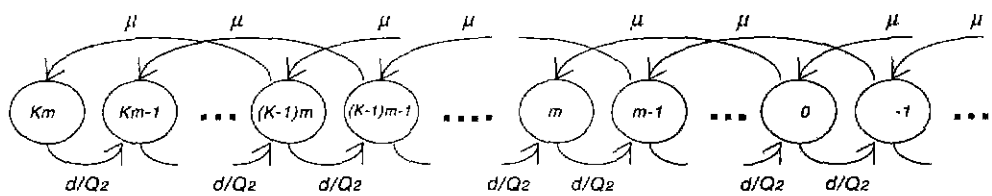

Figure 9. Condensed state transition diagram of stage one

Consider stage one of the decomposed subsystems. Since every order from stage two is Q_2 units at a time, we consider a condensed state transition diagram as shown in Figure 9, in which Q_2 numbers of states are condensed and represented by one state. In figure 9, the system is in state $k(k \geq 0)$ when the inventory level at the out-buffer is equal to kQ_2. Since the maximum number of kanban cards which could wait at the kanban board of stage two is limited by K_2, the number of backorders at stage one can not exceed $K_2 Q_2$. Thus, when the system is in state $k(-K_2 < k < 0)$, the number of backorders is equal to $|k|Q_2$. When the system is in state $k(k \leq -K_2)$, the number of backorders is equal to $K_2 Q_2$. The limiting probability, P'_k, that a system will be in state k in the condensed state transition diagram can be determined as follows:

$$P'_k = \sum_{j=(k-1)Q_2+1}^{kQ_2} P_j = \begin{cases} \dfrac{1}{m} - \dfrac{r_1^{(K_1 m-k)Q_2+1}(1-r_1^{Q_2})}{mQ_2(1-r_1)} & \text{if } (K_1-1)m < k \le K_1 m \\ \rho_1(1-r_1^{Q_2})r_1^{(K_1-1)m-k)Q_2} & \text{if } k \le (K_1-1)m \end{cases} \quad (30)$$

where r_1 is the root of the characteristic equation $r^{mQ_2+1} - (1+\rho_1 mQ_2)r + \rho_1 mQ_2 = 0$. The total number of items at stage one is bounded by $K_1 mQ_2$. When $k \le 0$, there are backorders and the out-buffer is empty. Hence, $NI_1 = (K_1 - 1)mQ_2 = 0$. When $k > 0$, where k satisfies $(j-1)m < k \le jm$, $1 \le j \le K_1 - 1$, it can be easily seen that there are j containers at the out-buffer. Thus, $NI_1 = (K_1 - j - 1)mQ_2$. When $(K_1 - 1)m < k \le K_1 m$, all containers are at the out-buffer and the in-buffer are empty. In summary, we have

$$NI_1 = \begin{cases} (K_1-1)mQ_2 & \text{if } k \le 0 \\ (K_1 - j - 1)mQ_2 & \text{if } (j-1)m < k \le jm,\ 1 \le j \le K_1 - 1 \\ 0 & \text{if } k = K_1 m \end{cases} \quad (31)$$

The expected number of items waiting at the in-buffer can be expressed as:

$$\overline{NI_1} = \sum_{j=1}^{K_1-1} \sum_{k=(j-1)m+1}^{jm} (K_1 - j - 1)mQ_2 P'_k + (K_1-1)mQ_2 \sum_{k=0}^{-\infty} P'_k \quad (32)$$

In this case,

$$\overline{NI_1} = \frac{\overline{NI_1}}{d} = \frac{r_1^{mQ_2+1}(1 - r_2^{(K_1-1)mQ_2}}{d(1-r_1)} \quad (33)$$

The expected on-hand inventory at the out-buffer can be expressed as:

$$\overline{NO_1} = \sum_{k=0}^{k_1 m} kQ_2 \cdot P'_k \quad (35)$$

In this case,

$$\overline{TO_1} = \frac{\overline{NO_1}}{d} = \frac{K_1 mQ_2}{d} - \frac{(m-1)Q_2}{2d} - \frac{r_1}{d(1-r_1)} + \frac{\rho_1 Q_2 r_1^{((K_1-1)m+1)Q_2}}{d(1-r_1^{Q_2})} \quad (35)$$

Similarly, the expected backorder (the number of items waiting at the kanban board) is given by $\overline{NB_1}$ and the expected order response time is given by $\overline{TB_1}$,

where

$$\overline{NB_1} = \sum_{k=-1}^{-K_2}(-k)Q_2\ P'_k + K_2Q_2 \sum_{k=-(k_2+1)}^{\infty-} P'_k$$

$$\overline{TB_1} = \frac{\overline{NB_1}}{d} = \frac{Q_2\rho_1 r_1^{((K_1-1)m+1)Q_2}(1-r_1^{K_2Q_2})}{d(1-r_1^{Q_2})} \tag{36}$$

Appendix C. Determination of $P(x+km \to x)$

Consider the transition from state $x+km$ to state x, where

$$k = k_0 + w, \quad w = 0, 1, 2, \cdots$$

$$k_0 = \left\lceil \frac{K_2 - x}{m} \right\rceil, \quad 0 \le x \le K_2 - 1$$

In order to make the transition, $x+km-K_2+1$ batches should be processed at stage two and k batches should be processed at stage one. Let E_1 and E_2 be the events that a batch is finished at stage one and two, respectively, N_{E_1} and N_{E_2} and be the numbers of event E_1 and E_2, respectively. Note that we subtract 1 from the total number of event E_1 which is required to make the transition since the first event has to be E_1

$$\begin{aligned} N_{E_1} &= k-1 \\ &= (k_0-1)+w \\ &= u+w, \ 0 \le u \le \left\lceil \frac{K_2}{m} \right\rceil - 1 \end{aligned} \tag{37}$$

$$\begin{aligned} N_{E_2} &= x+km-K_2+1 \\ &= (x+k_0m-K_2+1)+wm \\ &= v+wm, \quad 0 \le v \le m \end{aligned} \tag{38}$$

In this case, the total number of combinations of arranging two events, $GN(\cdot)$, can be expressed as:

$$GN(N_{E_1}, N_{E_2}, m) = \frac{(N_{E_1}+N_{E_2})!}{N_{E_1}!N_{E_2}!} \tag{39}$$

However, there are certain types of combinations which may not be possible. When K_2 kanban cards are waiting at the kanban board, the next kanban card cannot arrive until a batch is finished at stage one. Hence, the number of consecutive E_2 cannot exceed m. The number of impossible combinations, $IN(\cdot)$, can be determined by the following recursive equations:

$$IN(N_{E_1}, N_{E_2}, m) = \sum_{s=0}^{w-1} (GI(s, sm, m) - IN(s, sm, m)) \cdot GI(N_{E_1} - s,\ N_{E_2} - (s+1)m - 1, m) \tag{40}$$

$$IN(u, v, m) = 0, \quad \forall u, v, m \tag{41}$$

In addition, we need to consider some other combinations which require special attention. After m kanban cards receive finished items from stage one, the next kanban card needs to wait until a batch is finished at stage one. Thus, E_1 has to follow right after m consecutive E_2. Consider this m consecutive E_2 with one E_1 as a restricted group. The maximum number of the restricted group is w. If a certain arrangement has z restricted groups, $1 \le z \le w$, we can assign the remaining number of events to the $z+1$ positions in-between the restricted groups. Thus, the number of restricted combinations, $RN(\cdot)$, when the number of restricted group is z can be determined as follows:

$$\begin{aligned} RN(N_{E_1}, N_{E_2}, m, z) = & \sum_{g_1=0}^{w-z} \sum_{g_2=0}^{w-z-g_1} \cdots \sum_{g_z=0}^{w-z-g_1-\cdots-g_{z-1}} (GN(g_1, mg_1, m) \\ & - IN(g_1, mg_1, m) - SRN(g_1, mg_1, m)) \cdots \\ & \cdots (GN(g_z, mg_z, m) - IN(g_z, mg_z, m) - SRN(g_z, mg_z, m)) \\ & \Big(GN(N_{E_1} - z - \sum_{i=1}^{z} g_i,\ N_{E_2} - zm - \sum_{i=1}^{z} g_i m,\ m) \\ & - IN(N_{E_1} - z - \sum_{i=1}^{z} g_i,\ N_{E_2} - zm - \sum_{i=1}^{z} g_i m,\ m) \\ & - SRN(N_{E_1} - z - \sum_{i=1}^{z} g_i,\ N_{E_2} - zm - \sum_{i=1}^{z} g_i m,\ m)\Big) \end{aligned} \tag{42}$$

where $SRN(N_{E_1}, N_{E_2}, m) = \sum_{z=1}^{w} RN(N_{E_1}, N_{E_2}, m, z)$. Finally, the probability of transition from $x + km$ to x is given by:

$$
\begin{aligned}
P(x+km \rightarrow x) = \; & (GN(N_{E_1}, N_{E_2}, m) - IN(N_{E_1}, N_{E_2}, m) \\
& - SRN(N_{E_1}, N_{E_2}, m)) \cdot p(E_1)^{N_{E_1}}, p(E_2)^{N_{E_2}} \\
& + \sum_{z=1}^{w} RN(N_E, N_{E^c}, m, z)\; p(E_1)^{N_E}\, p(E_2)^{N_{E_2-z}}
\end{aligned}
\tag{43}
$$

where $p(E_i) = \dfrac{\mu_i}{\mu_1 + \mu_2}, \; i = 1, 2.$