IIE Transactions (2000) **32**, 999–1012

The extended kanban control system for production coordination of assembly manufacturing systems

CLAUDINE CHAOUIYA[1], GEORGE LIBEROPOULOS[2] and YVES DALLERY[3]

[1]*Departement Genie Bio Medical (ESIL/GBM), Ecole Superieure d'Ingenieurs de Luminy, Parc Scientifique et Technologique de Luminy, case 925, FR-13288 Marseille Cedex 9, France*
E-mail: chaouiya@esil.univ-mrs.fr
[2]*Department of Mechanical and Industrial Engineering, University of Thessaly, Pedion Areos, GR-38334, Volos, Greece*
E-mail: glib@mie.uth.gr
[3]*Laboratoire Productique et Logistique, Ecole Centrale de Paris, Grande Voie des Vignes, FR-92295 Chatenay-Malabry Cedex, France*
E-mail: dallery@pl.ecp.fr

Received June 1999 and accepted January 2000

In assembly manufacturing systems there are points in the production process where several component parts are put together in areas called assembly cells so as to form more complex parts called subassemblies. In this paper, we present and compare two variants of the Extended Kanban Control System (EKCS) – a recently developed pull production control mechanism that combines base stock and kanban control – for the production coordination of assembly manufacturing systems. In both variants, the production of a new subassembly is authorized only when an assembly kanban is available. Assembly kanbans become available when finished subassemblies are consumed. If an assembly kanban is available, in the first variant, each component part of a subassembly is released into the assembly cell as soon as *it* is available (independent release). In the second variant, however, it is released only when *all* other component parts also become available (simultaneous release). In both variants, when a component part is released into the assembly cell, it releases its kanban, thus authorizing the production of a new component part.

1. Introduction

In recent years, manufacturing management interests have turned towards cellular manufacturing. In cellular manufacturing production is divided into production *cells*. Each cell may be thought of as a production/inventory system composed of a manufacturing process and an output buffer. The manufacturing process may consist of a single machine or a subnetwork of several machines (e.g., a production line). It contains parts that are currently being processed (referred to as the *Work-In-Process* (WIP) of the cell). The output buffer is a storage area that contains parts that have completed processing in the cell (referred to as *finished parts* of the cell). The manufacturing system is fed by raw parts and releases finished parts to customers.

Each cell has a clear production task and is self managed. Often, for each customer, one needs to bring together a network of cells, where each cell may have a different owner. An important managerial concern is how to coordinate production among different self-managed cells so as to create a coherent integrated system. In many cases, coordination is achieved via a pull control policy, i.e., a policy that decides when to release parts in each cell based on when customer demands arrive to the system. A significant amount of work has been devoted to this issue for serial systems, i.e., systems consisting of cells in series. Different pull control mechanisms have been proposed in the literature, among which, the Base Stock Control System (BSCS) and the Kanban Control System (KCS) have attracted significant attention (Buzacott and Shanthikumar, 1993; Hopp and Spearman, 1996). The BSCS is a classical mechanism encountered in inventory theory, while the KCS was invented in Toyota in the 1970s and has since been widely used in industry (Monden, 1983). The main advantage of these two control policies is that they are both very simple to understand and implement as each policy depends on only one parameter per cell. Neither policy, however, always achieves a good tradeoff between inventory costs and customer service. As a result, more general pull control policies leading to better tradeoffs have been proposed. Among these, the Generalized Kanban Control System (GKCS), introduced by Buzacott (1989) and studied in detail by Buzacott and Shanthikumar (1993), and the Extended Kanban Control System (EKCS), proposed by Dallery and Liberopoulos (2000), are of special interest. Each of these policies depends on two parameters per cell. A comparison of these four control policies as well as other policies can be found in Liberopoulos and Dallery (2000).

For industrial applications, it is important to extend these control policies to systems having more general structures than "cells in series". Assembly systems are of particular interest in manufacturing. In an assembly system, a cell may have more than one immediate upstream cell. Although this situation is highly prevalent in the industry, not much work has been done in analyzing it. In Sbiti *et al.* (1999) the BSCS is extended to assembly systems, and in Di Mascolo and Dallery (1996) the KCS is extended to assembly systems. In the latter case, an important additional (with respect to the serial system case) control element arises having to do with the way in which component parts and kanbans are released prior to assembly. Two cases are considered, namely a simultaneous release mechanism and an independent release mechanism leading to the definitions of the so-called Simultaneous Kanban Control System (SKCS) and the Independent Kanban Control System (IKCS), respectively.

The purpose of this paper is to generalize the EKCS to assembly systems. As in the case of the KCS, we define two different control policies depending on whether component parts and kanbans are released simultaneously or independently prior to assembly. This leads to the definitions of the Simultaneous Extended Kanban Control System (SEKCS) and the Independent Extended Kanban Control System (IEKCS), respectively.

The rest of the paper is organized as follows. Section 2 briefly describes the EKCS defined in Dallery and Liberopoulos (2000) in the case of serial systems. In Section 3, we extend this definition to assembly systems, first describing the SEKCS and then the IEKCS. In Section 4 we show special cases in which the EKCS reduces to the classical KCS and BSCS. Properties of the two control mechanisms, the SEKCS and the IEKCS, are given in Section 5. These properties concern invariants, bounds, evolution equations, the influence of the variation of the parameters, and production capacity. Finally, the two mechanisms are compared in Section 6 and conclusions are drawn in Section 7.

2. Extended kanban control system for serial systems

The EKCS was recently introduced by Dallery and Liberopoulos (2000). This new kanban-based control system has the following advantageous features:

- Simplicity (compared to the GKCS).
- WIP limitation in each cell (unlike the BSCS).
- Immediate transfer of demands to all cells (unlike the KCS).
- Clear separation of roles of the parameters, i.e., the number of kanbans and the base stock level (unlike the GKCS).

In this section, we briefly describe the EKCS for a manufacturing system having cells in series. For a detailed description and properties of this system, refer to Dallery and Liberopoulos (2000). Figure 1 describes a manufacturing system having N cells in series. Each cell may be seen as a manufacturing process (e.g., single machine, production line, flexible manufacturing cell, job-shop, etc.) with an output buffer.

The EKCS is a pull control mechanism that can be viewed as a combination of the BSCS and the KCS (see Buzacott and Shanthikumar, 1993; Dallery and Liberopoulos, 2000; Liberopoulos and Dallery, 2000). Figure 2 shows the queueing network model of an EKCS having N cells in series.

We use the following notation that was introduced in Dallery and Liberopoulos (2000):

p_i, $i = 1, \dots, N$ = a cell-i finished part;
q_i, $i = 1, \dots, N$ = a part currently being processed in cell i;
d_i, $i = 1, \dots, N$ = a demand for the production of a new p_i;
d_{N+1} = a demand for a p_N;
a_i, $i = 1, \dots, N$ = an authorization card (kanban) for the production of a new p_i.

Table 1 describes the contents and initial state of the queues, or network of queues in the case of MP_i, in Fig. 2.

The initial number of raw parts in the raw parts buffer P_0 and the arrival process of new parts into P_0 fall outside the scope of the control mechanism and are considered as given. Each cell i has K_i kanbans a_i that authorize the production of cell i finished parts. Initially, in cell i, there are S_i kanbans a_i attached onto an equal number of parts p_i in PA_i, and therefore $K_i - S_i$ kanbans a_i in A_i (initially no part is being processed in MP_i). When a customer demand arrives to the system, it is immediately transmitted to all cells adding one demand to the contents of queues D_i ($i = 1, \dots, N+1$).

The behavior of the system can be described as follows.

Release of parts into the manufacturing cells

At the first manufacturing cell, queues P_0, A_1, and D_1 are joined in a synchronization station. Raw parts in P_0 do

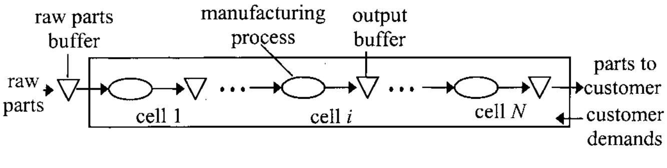

Fig. 1. Manufacturing system with N cells in series.

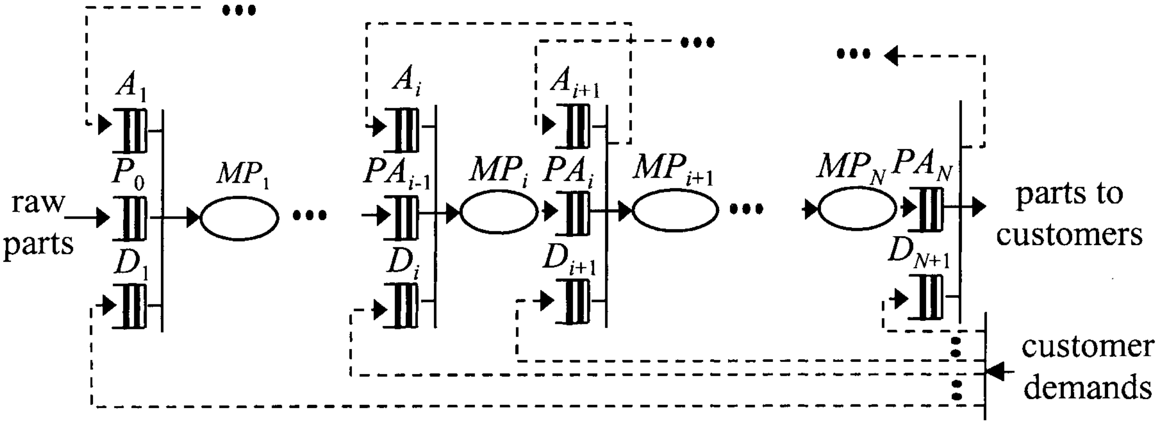

Fig. 2. Queueing network model of the EKCS.

not have any kanbans attached to them. Therefore, cell 1 can begin processing a part as soon as there is at least one part p_0 in P_0, one authorization card a_1 in A_1, and one demand d_1 in D_1. When these conditions are satisfied, then:

(1) Kanban a_1 is attached onto p_0, which is relabeled q_1, and together they are transferred downstream to MP_1 as a pair (q_1, a_1).
(2) Demand d_1 is satisfied and is therefore discarded.

The ith manufacturing cell ($i = 1, \dots, N$) can begin processing a part only when there is at least one pair (p_{i-1}, a_{i-1}) in PA_{i-1}, one authorization card a_i in A_i, and one demand d_i in D_i. When these conditions are satisfied, then:

(1) Kanban a_{i-1} is detached from p_{i-1} and is transferred upstream to A_{i-1}.
(2) Kanban a_i is attached onto p_{i-1}, which is relabeled q_i, and together they are transferred downstream to MP_i as a pair (q_i, a_i).
(3) Demand d_i is satisfied and is therefore discarded.

When part q_i ($i = 1, \dots, N$) finishes its processing in MP_i, it is relabeled p_i, and, together with kanban a_i that was attached onto it, they join PA_i as a pair (p_i, a_i).

Delivery of finished parts to customers

There is no need for an authorization to release a finished part p_N to a customer. Therefore, the delivery of a finished part can occur as soon as there is at least one pair (p_N, a_N) in PA_N and one demand d_{N+1} in D_{N+1}. When these conditions are satisfied, then:

(1) Kanban a_N is detached from p_N and is transferred upstream to A_N.
(2) Part p_N is released to the customer.
(3) Demand d_{N+1} is satisfied and is therefore discarded.

The EKCS has two parameters for every cell i, K_i and S_i. These parameters must be adjusted to achieve a good compromise between: (i) keeping a low inventory of parts in the system; and (ii) attaining a high level of immediate customer demand satisfaction. In this, their role is clearly separate. The base stock parameter basically provides a buffer against stockouts, which are associated with bad customer service. The use of kanbans basically ensures that the number of parts (WIP plus the finished parts) in each cell is bounded by the number of kanbans in that cell.

Table 1. Contents and initial state of the queues in the EKCS

Queue		*Contents*	*Initial state*
MP_i	$i = 1, \dots, N$	(q_i, a_i)	0
PA_i	$i = 1, \dots, N$	(p_i, a_i)	S_i
A_i	$i = 1, \dots, N$	a_i	$K_i - S_i$
D_i	$i = 1, \dots, N + 1$	d_i	0

3. Extended kanban control systems for assembly systems

The EKCS described for the serial system configuration in Fig. 2 can be extended to manufacturing systems having an assembly configuration. Figure 3 illustrates the tree-structured topology of a system having assembly cells (cells supplied by several raw parts buffers) and manufacturing cells (cells supplied by a single raw parts buffer).

For simplicity, we restrict our study to assembly systems having ($R - 1$) manufacturing cells supplying a single assembly cell (see Figs. 4 and 5). However, what follows can be easily extended to more general assembly topologies.

The structure that we are considering is the classical assembly structure where each item of the final product is produced by putting together one finished part from each manufacturing cell. Extending this to situations where

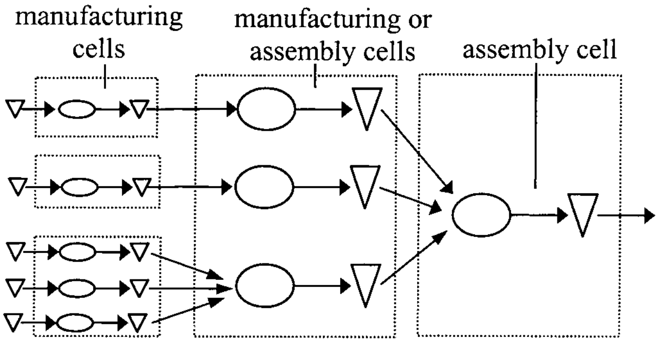

Fig. 3. General topology for assembly manufacturing systems.

more than one item of each manufacturing cell is needed to assemble a single item of the final product is straightforward and for the sake of conciseness will not be considered in this paper.

The extension of the EKCS to assembly systems leads to two kanban release mechanisms as was the case in the extension of the KCS to assembly systems (Di Mascolo and Dallery, 1996). These mechanisms are the Simultaneous EKCS (SEKCS) and the Independent EKCS (IEKCS) and are described in detail next.

3.1. *Simultaneous extended kanban control system*

Figure 4 shows the queueing network model of the SEKCS for the case of $(R-1)$ manufacturing cells supplying a single assembly cell.

We use the following notation:

$p_{0,i},\ i=1,\ldots,R-1$ = a cell-i raw part;
$p_i,\ i=1,\ldots,R-1$ = a cell-i finished part;
p_R = a cell-R assembled part;
$q_i,\ i=1,\ldots,R-1$ = a part currently being processed in cell i;
q_R = a part currently being assembled in cell R;
$d_i,\ i=1,\ldots,R-1$ = a demand for the production of a new p_i;
d_R = a demand for the assembly of a new p_R;
d_{R+1} = a demand for the delivery of an assembled part p_R;
$a_i,\ i=1,\ldots,R-1$ = an authorization card (kanban) for the production of a new p_i;
a_R = an authorization card (kanban) for the assembly of a new p_R.

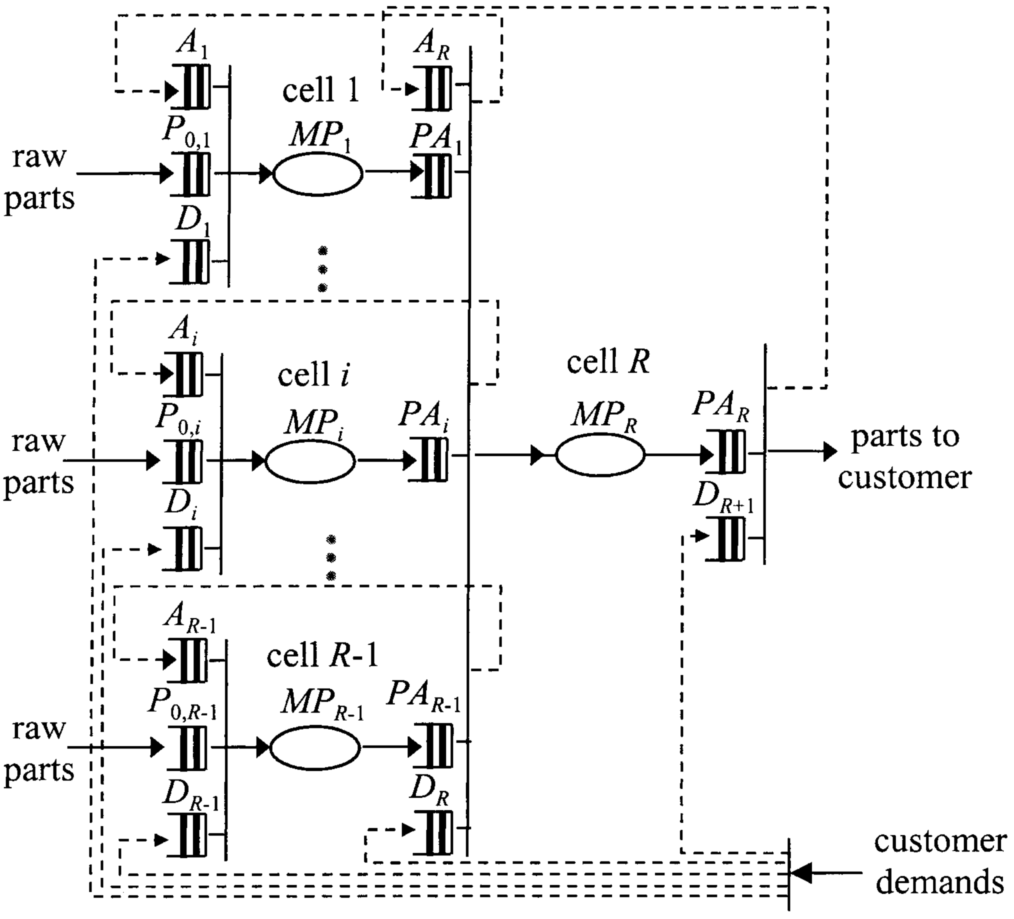

Fig. 4. Queueing network model for the SEKCS.

Table 2 shows the contents and initial state of the queues.

As was the case in the EKCS, queue $P_{0,i}$ ($i = 1, \ldots, R - 1$) represents the raw parts buffer supplying manufacturing cell i. The initial number of raw parts in $P_{0,i}$ and the arrival process of new parts into $P_{0,i}$ fall outside the scope of the control mechanism and are considered as given. When a customer demand arrives to the system, it is immediately transmitted to all cells by adding one to the contents of each of the queues D_i ($i = 1, \ldots, R + 1$).

The behavior of the SEKCS can be described as follows.

Release of parts into the manufacturing cells

At each manufacturing cell i ($i = 1, \ldots, R - 1$), queues $P_{0,i}$, A_i, and D_i are joined in a synchronization station. This means that cell i can begin the production of a part only when there is at least one part $p_{0,i}$ in $P_{0,i}$, one authorization card a_i in A_i, and one demand d_i in D_i. When these conditions are met, then:

(1) Kanban a_i is attached onto $p_{0,i}$ which is relabeled q_i, and together they are transferred downstream to MP_i as a pair (q_i, a_i).
(2) Demand d_i is satisfied and is therefore discarded.

When part q_i finishes its processing in MP_i, it is relabeled p_i, and, together with kanban a_i that was attached onto it, they join PA_i as a pair (p_i, a_i).

Release of parts into the assembly cell

At the assembly cell R, queues PA_i ($i = 1, \ldots, R - 1$), A_R, and D_R are joined in a synchronization station. This means that the assembly operation can begin only when there is at least one pair (p_i, a_i) in PA_i for every $i = 1, \ldots, R - 1$, one authorization a_R in A_R, and one demand d_{R+1} in D_{R+1}. When these conditions are satisfied, then:

(1) Kanbans a_i are *simultaneously* detached from the p_i ($i = 1, \ldots, R - 1$) and are transferred upstream to their corresponding previous cell.
(2) Kanban a_R is attached onto $(p_1, p_2, \ldots, p_{R-1})$ which is relabeled q_R, and together they are transferred downstream to MP_R as a pair (q_R, a_R).
(3) Demand d_R is satisfied and is therefore discarded.

Table 2. Contents and initial state of the queues in the SEKCS

Queue		*Contents*	*Initial state*
MP_i	$i = 1, \ldots, R - 1$	(q_i, a_i)	0
MP_R		(q_R, a_R)	0
PA_i	$i = 1, \ldots, R - 1$	(p_i, a_i)	S_i
PA_R		(p_R, a_R)	S_R
A_i	$i = 1, \ldots, R - 1$	a_i	$K_i - S_i$
A_R		a_R	$K_R - S_R$
D_i	$i = 1, \ldots, R - 1$	D_i	0
D_R		D_R	0
D_{R+1}		d_{R+1}	0

When a part q_R finishes its assembly process in MP_R, it is relabeled p_R, and, together with the kanban a_R that was attached onto it, they join PA_R as a pair (p_R, a_R).

Delivery of finished parts to the customer

At the final cell, queues PA_R and D_{R+1} are joined in a synchronization station. There is no need for an authorization to release a finished part to the customer. Therefore, the delivery of a finished assembled part can occur as soon as there is a pair (p_R, a_R) in PA_R and a demand d_{R+1} in D_{R+1}. When these conditions are satisfied, then:

(1) Kanban a_R is detached from p_R and is transferred upstream to A_R.
(2) Part p_R is released to the customer.
(3) Demand d_{R+1} is satisfied and is therefore discarded.

3.2. *Independent extended kanban control system*

Figure 5 shows the queueing network model of the IEKCS for the case of $(R - 1)$ manufacturing processes supplying a single assembly process.

We use the same notation as in the SEKCS except that d_R and a_R are each split into $R - 1$ components, $d_{R,i}$ and $a_{R,i}$, respectively, corresponding to each manufacturing cell. Thus,

$d_{R,i}$, $i = 1, \ldots, R - 1$ = a demand for the assembly of a new p_R using a cell-i finished part p_i;

$a_{R,i}$, $i = 1, \ldots, R - 1$ = an authorization card (issued from some a_R) for the assembly of a new p_R using a cell-i finished part p_i.

Table 3 describes the contents and initial states of queues $A_{R,i}$ (in place of A_R), B_i and $D_{R,i}$ (in place of D_R). The contents and initial states of all other queues are the same as those in the SEKCS (see Table 2).

As was the case in the SEKCS, queues $P_{0,i}$ ($i = 1, \ldots, R - 1$) represent the raw parts buffer supplying manufacturing cell i. The initial number of raw parts in $P_{0,i}$ and the arrival process of new parts into $P_{0,i}$ fall outside the scope of the control mechanism and are considered as given. When a customer demand arrives to the system, it is immediately transmitted to all cells by adding one to the contents of queues D_{R+1}, D_i, and $D_{R,i}$ ($i = 1, \ldots, R - 1$).

The behavior of the IEKCS can be described as follows.

Release of parts into the manufacturing cells

It is identical to the SEKCS.

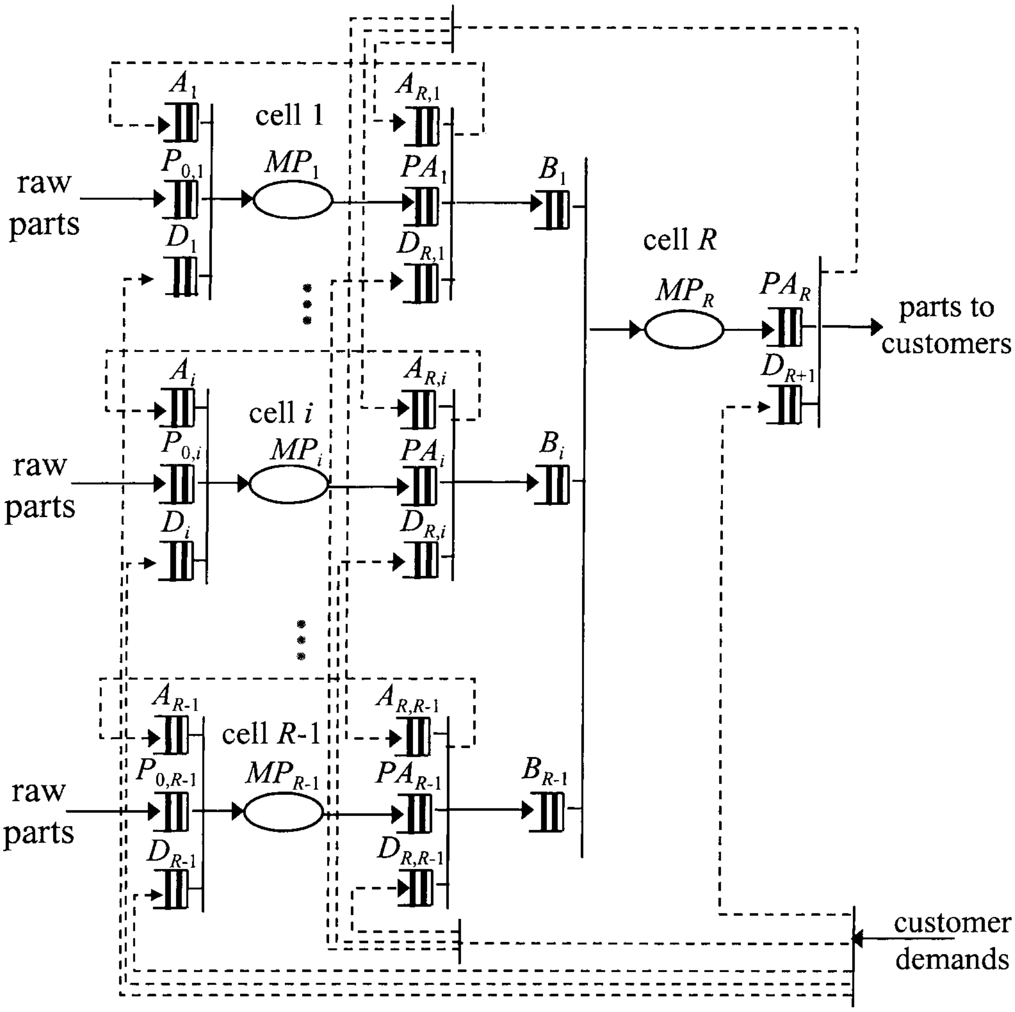

Fig. 5. Queueing network model for the IEKCS.

Release of parts into the assembly cell

The difference between the SEKCS and the IEKCS is in the way kanbans are transferred in cells $1,\dots,R-1$. In the SEKCS all kanbans are transferred simultaneously, whereas in the IEKCS they are transferred *independently* of each other. In the IEKCS, each demand for an assembly operation is split into $R-1$ demands $d_{R,i}$ $(i=1,\dots,R-1)$. Similarly, each cell-R kanban is split into $R-1$ kanbans $a_{R,i}$, upon its liberation from a finished part p_R.

Between the $R-1$ manufacturing cells $1,\dots,R-1$ and the assembly cell R there are two layers of synchronization stations. The first layer consists of $R-1$ synchronization stations in parallel, one for each manufacturing cell, and the second layer consists of a single synchronization station fed by the synchronization stations of the first layer. More precisely:

Table 3. Contents and initial state of some queues in the IEKCS

Queue		*Contents*	*Initial state*
$A_{R,i}$	$i=1,\dots,R-1$	$a_{R,i}$	K_R-S_R
B_i	$i=1,\dots,R-1$	$(p_i,a_{R,i})$	0
$D_{R,i}$	$i=1,\dots,R-1$	$d_{R,i}$	0

For each manufacturing cell i $(i=1,\dots,R-1)$, queues PA_i, $A_{R,i}$, and $D_{R,i}$ are joined in a synchronization station. The supply of cell i finished parts for the assembly operation can occur only when there is at least one pair (p_i,a_i) in PA_i, one authorization $a_{R,i}$ in $A_{R,i}$, and one demand $d_{R,i}$ in $D_{R,i}$. When these conditions are met, then:

(1) Kanban a_i is detached from p_i *independently* of what is going on in cells $1,\dots,i-1,i+1,\dots,$ $R-1$ and is transferred upstream to A_i.
(2) Kanban $a_{R,i}$ is attached onto p_i, and together they are transferred downstream to B_i as a pair $(p_i,a_{R,i})$.
(3) Demand $d_{R,i}$ is satisfied and is therefore discarded.

Queues B_i $(i=1,\dots,R-1)$, are joined in a synchronization station. When there is at least one pair $(p_i,a_{R,i})$ in each B_i, the assembly process can begin and the following happens:

(1) Pairs $(p_i, a_{R,i})$, are removed from queues B_i $(i = 1, \ldots, R-1)$, $(R-1)$-tuple $(p_1, p_2, \ldots, p_{R-1})$ is relabeled q_R, kanbans $(a_{R,1}, a_{R,2}, \ldots, a_{R,R-1})$ are merged into a kanban a_R, and pair (q_R, a_R) is transferred downstream to MP_R.

When a part q_R finishes its assembly process in MP_R, it is relabeled p_R, and, together with kanban a_R that was attached onto it, they join PA_R as a pair (p_R, a_R).

Delivery of finished parts to the customer

It is identical to the SEKCS except that when a kanban a_R is transferred back to the input of the assembly cell, it is split into $R-1$ kanbans, $a_{R,1}, a_{R,2}, \ldots, a_{R,R-1}$, and kanban $a_{R,i}$ joins queue $A_{R,i}$ $(i = 1, \ldots, R-1)$.

4. Special cases of the SEKCS and the IEKCS

Property 1 and Property 2 that follow refer to two special cases where the SEKCS (respectively the IEKCS) is equivalent to the Simultaneous Kanban Control System (SKCS) and to the Base Stock Control System (BSCS) (respectively to the Independent Kanban Control System (IKCS) and to the BSCS).

Property 1.

(1) *The SEKCS with $K_i = \infty$ and $S_i \geq 0$ $(i = 1, \ldots, R)$ is equivalent to the BSCS having a base stock of S_i finished parts in cell i $(i = 1, \ldots, R)$.*

(2) *The SEKCS with $K_i = S_i$ $(i = 1, \ldots, R)$ is equivalent to the SKCS having K_i kanbans in cell i $(i = 1, \ldots, R)$.*

The proof, which can be found in Chaouiya *et al.* (1998), is based on the elimination of the queues that play no role in the synchronization stations of the queueing network model of the SEKCS. Indeed, in case 1, queues A_i play no role in the synchronization stations they belong to since they have an infinite number of kanbans. Similarly, in case 2, queues D_i play no role in the synchronization stations they belong to since they always have a number of demands greater than the number of kanbans. Once these queues are eliminated, the remaining queueing network obtained is the same as that of the BSCS (in case 1) and of the SKCS (in case 2).

Property 2.

(1) *The IEKCS with $K_i = \infty$ and $S_i \geq 0$ $(i = 1, \ldots, R)$ is equivalent to the BSCS having a base stock of S_i finished parts in cell i $(i = 1, \ldots, R)$.*

(2) *The IEKCS with $K_i = S_i$ $(i = 1, \ldots, R)$ is equivalent to the IKCS having K_i kanbans in cell i $(i = 1, \ldots, R)$.*

The proof is similar to that of Property 1 and can be found in Chaouiya *et al.* (1998).

5. Properties of the SEKCS and the IEKCS

In this section, we present some basic properties of the SEKCS and IEKCS that are helpful in obtaining a better understanding, as well as some insights, in the behavior of the two systems. The purpose is twofold: (i) to understand the behavior of each control mechanism and in particular the influence of its parameters (the K_i's and the S_i's); and (ii) to compare the behavior of the SEKCS and the IEKCS in order to emphasize the particularity of each system and to see how the behavior of one system relates to that of the other. We will establish some relations on the population of the queues in the system, called invariants, which are valid at all times (Section 5.1). Moreover, we will derive bounds that imply a limitation on the WIP and on the number of finished parts in each cell (Section 5.2).

Both the SEKCS and the IEKCS can be modeled as Fork-Join Queuing Networks with Blocking (FJQN/B) as defined in Dallery *et al.* (1997). FJQN/Bs are queueing networks composed of a set of servers and a set of buffers, such that each buffer has exactly one upstream server and one downstream server. Each server may have several input buffers and/or several output buffers, and some servers may have no input (sources) or no output (sinks). Figures 4 and 5 show the FJQN/B models for the SEKCS and the IEKCS, respectively.

For these FJQN/Bs we will prove results concerning:

- invariance properties related to the cycles of the FJQN;
- conditions for deadlock freeness;
- recursive evolution equations using the operators '+' and 'max'.

These results are similar to those in Baccelli *et al.* (1992) and also Dallery *et al.* (1994, 1997) and are based on the equivalence of FJQN/Bs to Strongly Connected Marked Graphs (SCMGs).

We denote by $M(Q)$ the current population of any queue or network of queues, Q, in the system. $M(Q)$ varies with time and should therefore be a function of time. However, for simplicity, and since we are interested in invariants or instantaneous relations, we will omit this dependence on time. We first state some basic relations for the SEKCS and the IEKCS.

By definition of the initial state of the SEKCS and the IEKCS, queues A_i have $(K_i - S_i)$ free kanbans $(i = 1, \ldots, R-1)$, which means that $K_i - S_i \geq 0$. At the same time, queue A_R (in the case of the SEKCS) and queues $A_{R,i}$ $(i = 1, \ldots, R-1$, in the case of the IEKCS) have

$(K_R - S_R)$ free kanbans. The two parameters of cell i $(i = 1, \dots, R)$ are therefore constrained by:

$$K_i \geq S_i, \quad i = 1, \dots, R.$$

By definition of a synchronization station, at least one of the queues in a synchronization station is always empty. The mathematical expression of this is that the product of the populations of the queues in a synchronization station is zero.

For the synchronization stations in the SEKCS, this expression becomes:

$$M(A_i)M(P_{0,i})M(D_i) = 0, \quad i = 1, \dots, R-1, \tag{1}$$

$$M(A_R)M(D_R)\prod_{i=1}^{R-1} M(PA_i) = 0, \tag{2}$$

$$M(PA_R)M(D_{R+1}) = 0. \tag{3}$$

Again by definition of a synchronization station, Equations (1) and (3) that hold for the SEKCS, also hold for the IEKCS. In addition, in the IEKCS, the following expressions hold:

$$M(A_{R,i})M(PA_i)(D_{R,i}) = 0, \quad i = 1, \dots, R-1, \tag{4}$$

$$\prod_{i=1}^{R-1} M(B_i) = 0. \tag{5}$$

5.1. *Invariants*

Next, we present properties in the form of invariants pertaining to the content of various queues of the queueing network models of the SEKCS and the IEKCS. Some of these invariants express the fact that within the queueing network model of the SEKCS and the IEKCS there exist several closed subnetworks with a constant population.

Property 3. *In the SEKCS, the following holds*:

$$M(A_i) + M(MP_i) + M(PA_i) = K_i, \quad i = 1, \dots, R-1, \tag{6}$$

$$M(A_i) - M(D_i) + M(D_R) = K_i - S_i, \quad i = 1, \dots, R-1, \tag{7}$$

$$M(PA_i) - M(D_R) + M(MP_i) + M(D_i) = S_i, \quad i = 1, \dots, R-1, \tag{8}$$

$$M(A_R) + M(MP_R) + M(PA_R) = K_R, \tag{9}$$

$$M(A_R) + M(D_{R+1}) - M(D_R) = K_R - S_R, \tag{10}$$

$$M(PA_R) - M(D_{R+1}) + M(MP_R) + M(D_R) = S_R. \tag{11}$$

Proof. Equations (6) and (9) are straightforward since the total number of kanbans at each cell is constant. When the SEKCS is in its initial state, there are $(K_i - S_i)$ kanbans a_i in queue A_i $(i = 1, \dots, R-1)$ and queue D_i $(i = 1, \dots, R)$ is empty. Therefore, Equation (7) holds initially. As the SEKCS evolves starting from its initial state, Equation (7) remains true since:

(1) When a kanban is transferred to A_i, a demand leaves D_R.
(2) When a kanban leaves A_i, a demand also leaves D_i.
(3) When a demand joins D_i, a demand also joins D_R.

To prove Equation (8) it suffices to substitute $M(A_i)$ from Equation (6) into Equation (7). Equations (10) and (11) can be proved using similar arguments. ■

Property 4. *In the IEKCS, the following holds*:

$$M(A_i) + M(MP_i) + M(PA_i) = K_i, \quad i = 1, \dots, R-1, \tag{12}$$

$$M(A_i) - M(D_i) + M(D_{R,i}) = K_i - S_i, \quad i = 1, \dots, R-1, \tag{13}$$

$$M(PA_i) - M(D_{R,i}) + M(MP_i) + M(D_i) = S_i, \ i = 1, \dots, R-1, \tag{14}$$

$$M(A_{R,i}) + M(B_i) - M(A_{R,j}) - M(B_j) = 0, \quad i,j = 1, \dots, R-1, \tag{15}$$

$$M(A_{R,i}) + M(B_i) - M(MP_R) - M(PA_R) = K_R, \quad i = 1, \dots, R-1, \tag{16}$$

$$M(A_{R,i}) + M(D_{R+1}) - (D_{R,i}) = K_R - S_R, \quad i = 1, \dots, R-1, \tag{17}$$

$$M(PA_R) - M(D_{R+1}) + M(B_i) + M(MP_R) + M(D_{R,i}) = S_R, \quad i = 1, \dots, R-1. \tag{18}$$

The proof is similar to that of Property 3 and is therefore omitted.

5.2. *Bounds*

The bounds that follow, although quite obvious are important since they establish that the Work-In-Process (WIP) is limited in the SEKCS as well in the IEKCS.

Property 5. *In the SEKCS, the following holds for all the cells*:

$$\begin{array}{ll} 0 \leq M(A_i) \leq K_i, & i = 1, \dots, R, \\ 0 \leq M(PA_i) \leq K_i, & i = 1, \dots, R, \\ 0 \leq M(MP_i) \leq K_i, & i = 1, \dots, R, \\ 0 \leq M(MP_i) + M(PA_i) \leq K_i, & i = 1, \dots, R. \end{array} \tag{19}$$

Property 6. *In the IEKCS, the following holds for all the manufacturing cells*:

$$\begin{array}{ll} 0 \leq M(A_i) \leq K_i, & i = 1, \dots, R-1, \\ 0 \leq M(PA_i) \leq K_i, & i = 1, \dots, R-1, \\ 0 \leq M(MP_i) \leq K_i, & i = 1, \dots, R-1, \\ 0 \leq M(MP_i) + M(PA_i) \leq K_i, & i = 1, \dots, R-1. \end{array} \tag{20}$$

In the IEKCS, the following holds for the assembly cell:

$$\begin{aligned} &0 \le M(B_i) \le K_R, && i = 1,\dots,R-1,\\ &0 \le M(A_{R,i}) \le K_R, && i = 1,\dots,R-1,\\ &0 \le M(MP_R) \le K_R, &&\\ &0 \le M(PA_R) \le K_R, &&\\ &0 \le M(B_i) + M(MP_R) + M(D_{R,i}) \le K_R, && i = 1,\dots,R-1. \end{aligned} \tag{21}$$

Property 5 and Property 6 express the fact that, since there is no new kanban generated in the production process, the total number of parts can not exceed the number of kanbans in each cell. A formal proof can be found in Chaouiya *et al.* (1998).

Property 7 and Property 8 also present some bounds.

Property 7. *In the SEKCS the following holds*:

$$M(PA_i) - M(D_R) \le S_i, \quad i = 1,\dots,R-1, \tag{22}$$

$$M(PA_R) - M(D_{R+1}) \le S_R. \tag{23}$$

Proof. Equation (22) follows from Equation (8), and Equation (23) follows from Equation (11). ■

Property 8. *In the IEKCS, the following holds*:

$$M(PA_i) - M(D_{R,i}) \le S_i, \quad i = 1,\dots,R-1, \tag{24}$$

$$M(PA_R) - M(D_{R+1}) \le S_R. \tag{25}$$

The proof is similar to that of Property 7 and is therefore omitted.

5.3. *Evolution equations*

The purpose of this section is to provide the basic equations that describe the evolution of the SEKCS and the IEKCS. We show that the dynamics of the two systems can be described by recursive evolution equations that utilize the operators "+" and "max" only. These evolution equations are of interest because they allow us to establish some useful properties on the behavior of the SEKCS and the IEKCS (Section 5.4) as well as to compare the two systems (Section 6). Moreover, this approach is very general since it is a sample-path approach that does not require any assumption on the distributions of the random variables (processing times and interarrival times of demands).

For the sake of simplicity, we assume that there is an infinite supply of raw parts in $p_{i,0}$ ($i = 1,\dots,R-1$). The results that follow, however, could be extended to incorporate external arrival processes of parts at the expense of more tedious derivations.

5.3.1. *Evolution equations for the SEKCS*

We introduce the following notation for the times of all possible events that may take place:

$I_{i,n}$ = the time of the nth arrival in MP_i (Input), $i = 1,\dots,R$;
$I_{R+1,n}$ = the time of the nth delivery to a customer;
$O_{i,n}$ = the time of the nth departure from MP_i (Output), $i = 1,\dots,R$;
D_n = the time of the nth demand arrival (Demand).

Finally, for the sake of simplicity, we assume that MP_i consists of a single machine, and we let $\sigma_{i,n}$ be the processing time of the nth part at the machine in MP_i ($i = 1,\dots,R$).

Clearly, the following holds:

$$\begin{aligned} &I_{i,n-m} \le I_{i,n}, && i = 1,\dots,R+1 \text{ and } n,\ m = 1,2,\dots,\\ &O_{i,n-m} \le O_{i,n}, && i = 1,\dots,R \text{ and } n,\ m = 1,2,\dots,\\ &D_{n-m} \le D_n, && n,\ m = 1,2,\dots, \end{aligned}$$

where by definition $I_{i,n}$, $O_{i,n}$, and D_n are zero for $n \le 0$. We now have the proposition below, which states the evolution equations for the SEKCS.

Proposition 1. *In the SEKCS, the times of events are related by the following evolution equations*:

$$I_{i,n} = \max(D_n, I_{R,n-(K_i-S_i)}), \quad i = 1,\dots,R-1, \tag{26}$$

$$I_{R,n} = \max(D_n, \max_{i=1,\dots,R-1}(O_{i,n-S_i}), I_{R+1,\, n-(K_R-S_R)}), \tag{27}$$

$$I_{R+1,n} = \max(D_n, O_{R,n-S_R}), \tag{28}$$

$$O_{i,n} = \sigma_{i,n} + \max(I_{i,n}, O_{i,n-1}), \quad i = 1,\dots,R. \tag{29}$$

Proof. Equation (26) gives an expression for $I_{i,n}$ which, by definition, represents the time at which the nth pair (q_i, a_i) is released into MP_i ($i = 1,\dots,R-1$). Indeed, this release occurs only when two conditions are met:

(1) The nth demand d_i has arrived in D_i.
(2) The $(n-(K_i-S_i))$th kanban has arrived in A_i, since initially there are $(K_i - S_i)$ kanbans in A_i.

We recall that, in the SEKCS, a kanban a_i arrives in A_i when a pair (q_R, a_R) is released into MP_R.

Because of the assumption that there is an infinite supply of raw parts in $P_{0,i}$, no condition involving arrivals in $P_{0,i}$ appears in Equation (26).

To prove Equation (27), we will use similar arguments, only now, arrivals in queues PA_i must also be taken into account. $I_{R,n}$ is, by definition, the time at which the nth pair (q_R, a_R) is released into MP_R. Indeed, this can occur only when three conditions are met:

(1) The nth demand d_R has arrived in D_R.
(2) The $(n-S_i)$th cell-i finished part with its attached kanban has arrived in PA_i ($i = 1,\dots,R-1$), since there are initially S_i pairs (p_i, a_i) in PA_i.
(3) The $(n-(K_R-S_R))$th kanban a_R has arrived in A_R, since initially there are $(K_R - S_R)$ kanbans in A_R.

Equation (28) gives an expression for $I_{R+1,n}$, which is, by definition, the time when the nth part is delivered to the customer. This event can occur when the two following conditions are met:

(1) The nth demand has arrived in D_n.
(2) The $(n - S_R)$th finished part p_R with its attached kanban has been released in PA_R, since initially there are S_R pairs (p_R, a_R) in PA_R.

Equation (29) gives an expression for $O_{i,n}$ which is, by definition, the time at which the nth pair (p_i, a_i) has completed processing in MP_i and is released in PA_i. This time is equal to the time at which the nth pair (q_i, a_i) begins its processing in MP_i plus its processing time $\sigma_{i,n}$. The nth pair (q_i, a_i) begins its processing when the two conditions below are satisfied:

(1) The nth pair (q_i, a_i) has been released in MP_i.
(2) The $(n - 1)$th part has completed its processing at MP_i. ■

5.3.2. *Evolution equations for the IEKCS*

In addition to the previous notation, let $L_{i,n}$ be the time when the nth pair $(p_i, a_{R,i})$ arrives at queue B_i (that is, the release time of the nth kanban a_i).

The following proposition states the evolution equations for the IEKCS.

Proposition 2. *In the IEKCS, the times of events are related by the following evolution equations*:

$$I_{i,n} = \max(D_n, L_{i,n-(K_i-S_i)}), \quad i = 1, \ldots, R-1, \tag{30}$$

$$I_{R,n} = \max_{i=1,\ldots,R-1}(L_{i,n}), \tag{31}$$

$$I_{R+1,n} = \max(D_n, O_{R,n-S_R}), \tag{32}$$

$$L_{i,n} = \max(D_n, O_{i,n-S_i}, I_{R+1,n-(K_R-S_R)}), \quad i = 1, \ldots, R-1, \tag{33}$$

$$O_{i,n} = \sigma_{i,n} + \max(I_{i,n}, O_{i,n-1}), \quad i = 1, \ldots, R. \tag{34}$$

Proof. Equation (30) differs from Equation (26) only in the condition for the release of kanbans a_i. Otherwise, the argument is similar to that for the SEKCS: the nth pair (q_i, a_i) enters MP_i $(i = 1, \ldots, R-1)$ when the two conditions below are met:

(1) The nth demand d_i has arrived in D_i.
(2) The $(n - (K_i - S_i))$th kanban has arrived in A_i, since initially there are $(K_i - S_i)$ kanbans in A_i. In the IEKCS, a kanban a_i arrives in A_i when a pair $(p_i, a_{R,i})$ is transferred to B_i.

As in the SEKCS, because of the assumption of an infinite supply of raw parts in $P_{0,i}$, no condition involving arrivals in $P_{0,i}$ appears in Equation (30).

Equation (31) represents the time at which the nth pair (q_R, a_R) is transferred to MP_R. This occurs when the nth pair $(p_i, a_{R,i})$ has arrived in B_i, for all $i = 1, \ldots, R-1$.

Equation (32) is the same as Equation (28).

In Equation (33), we consider the time at which the cell-i nth finished part is transferred to B_i. This event occurs as soon as the following three conditions are met:

(1) The nth demand has arrived in $D_{R,i}$.
(2) The $(n - S_i)$th pair (p_i, a_i) has arrived in PA_i, since initially there are S_i pairs in PA_i.
(3) The $(n - (K_R - S_R))$th kanban $a_{R,i}$ has arrived in $A_{R,i}$, since initially there are $(K_R - S_R)$ kanbans in $A_{R,i}$.

Finally, Equation (34) is the same as Equation (29). ■

5.4. *Variation of parameters*

In this section we study the influence of parameters K_i and S_i $(i = 1, \ldots, R)$ on the above-mentioned event times, and we derive some monotonicity properties.

Proofs are similar to the proof of "Stochastic Monotonicity with Respect to the Initial Marking" in Baccelli *et al.* (1992) and Baccelli and Liu (1992). We will use the fact that the time when events occur in the system can be computed recursively according to the evolution equations. Therefore, there exists a total ordering on the times described by Proposition 1 and Proposition 2 allowing us to use proofs by induction for the following results.

We compare two systems: the nominal system denoted by $\mathscr{S}$ and the modified system denoted by $\tilde{\mathscr{S}}$. These two systems differ only through their parameters: the number of kanbans and the base stock level in each cell. The parameters of the original systems are K_i and S_i, the parameters of the modified system are $\tilde{K}_i$ and $\tilde{S}_i$ $(i = 1, \ldots, R)$. On the other hand, the two systems have the same sequence of customer demand times (denoted by D_n for $\mathscr{S}$, and $\tilde{D}_n$ for $\tilde{\mathscr{S}}$) and the same sequence of processing times (denoted by $\sigma_{i,n}$ and $\tilde{\sigma}_{i,n}$ $(i = 1, \ldots, R)$).

Part 1 of Property 9 and Property 10 that follow states that increasing the number of kanbans in some cell, decreases the arrival and departure times in each cell of the system. Parts 2 and 3 state that increasing the base stock in any cell q $(q = 1, \ldots, R)$, decreases the arrival and departure times of all other cells. Moreover, the arrival time (respectively the departure time) of the nth part from cell q decreases with respect to the arrival time (respectively the departure time) of the $(n + \tilde{S}_q - S_q)$th part from MP_q. This means that an increase in the base stock of cell q from S_q to $\tilde{S}_q$ has the same effect as having $(\tilde{S}_q - S_q)$ extra parts enter the cell q and receive processing before the first demand arrives.

Property 9 and Property 10 imply that increasing the number of kanbans or the base stock level increases the response time of the SEKCS and the IEKCS. We will see in Section 5.5 that the number of kanbans is mainly related to the production capacity of the system, while the base stock level is related to the customer demands satisfaction.

5.4.1. *Variation of parameters in the SEKCS*

Property 9. *Consider the two systems $\mathcal{S}$ and $\tilde{\mathcal{S}}$ under the SEKCS. Then, we have*:

(1) *If $\tilde{K}_q > K_q$ for some q in $\{1, \ldots, R\}$, $\tilde{K}_i = K_i$ for all i in $\{1, \ldots, R\} - \{q\}$, and $\tilde{S}_i = S_i$ for all i in $\{1, \ldots, R\}$, then for all n*:

$$\tilde{I}_{i,n} \leq I_{i,n}, \quad i = 1, \ldots, R+1, \tag{35}$$

$$\tilde{O}_{i,n} \leq O_{i,n}, \quad i = 1, \ldots, R. \tag{36}$$

(2) *If $\tilde{K}_i = K_i$ for all i in $\{1, \ldots, R\}$, $\tilde{S}_q > S_q$ for some q in $\{1, \ldots, R-1\}$, and $\tilde{S}_i = S_i$ for all i in $\{1, \ldots, R\} - \{q\}$, then for all n*:

$$\tilde{I}_{i,n} \leq I_{i,n}, \quad i \in \{1, \ldots, R+1\} - \{q\}, \tag{37}$$

$$\tilde{I}_{q,n} \leq I_{q,n+(\tilde{S}_q - S_q)}, \tag{38}$$

$$\tilde{O}_{i,n} \leq O_{i,n}, \quad i \in \{1, \ldots, R\} - \{q\}, \tag{39}$$

$$\tilde{O}_{q,n} \leq O_{q,n\,+\,(\tilde{S}_q - S_q)}. \tag{40}$$

(3) *If $\tilde{K}_i = K_i$ $(i = 1, \ldots, R)$, $\tilde{S}_i = S_i$ $(i = 1, \ldots, R-1)$, and $\tilde{S}_R > S_R$, then for all n*:

$$\tilde{I}_{i,n} \leq I_{i,n+(\tilde{S}_R - S_R)}, \quad i = 1, \ldots, R, \tag{41}$$

$$\tilde{I}_{R+1,n} \leq I_{R+1,n}, \tag{42}$$

$$\tilde{O}_{i,n} \leq O_{i,n+(\tilde{S}_R - S_R)}, \quad i = 1, \ldots, R. \tag{43}$$

The proof can be found in Chaouiya *et al.* (1998).

5.4.2. *Variation of parameters in the IEKCS*

The influence of the variation of the parameters upon the event times for the IEKCS is basically the same as for the SEKCS. We will only state the corresponding property. Its proof is very similar to the proof of Property 9.

Property 10. *Consider the two systems $\mathcal{S}$ and $\tilde{\mathcal{S}}$ under the IEKCS. Then, we have*:

(1) If $\tilde{K}_q > K_q$ *for some q in* $\{1, \ldots, R\}$, $\tilde{K}_i = K_i$ *for* $i \in \{1, \ldots, R\} - \{q\}$, *and* $\tilde{S}_i = S_i$ $(i = 1, \ldots, R)$, *then for all n*:

$$\tilde{I}_{i,n} \leq I_{i,n}, \quad i = 1, \ldots, R+1, \tag{44}$$

$$\tilde{O}_{i,n}, \leq O_{i,n}, \quad i = 1, \ldots, R. \tag{45}$$

(2) *If $\tilde{K}_i = K_i$ for all i in $\{1, \ldots, R\}$, $\tilde{S}_q > S_q$ for some q in $\{1, \ldots, R-1\}$, and $\tilde{S}_i = S_i$ for all i in $\{1, \ldots, R\} - \{q\}$, then for all n*:

$$\tilde{I}_{i,n} \leq I_{i,n}, \quad i \in \{1, \ldots, R+1\} - \{q\}, \tag{46}$$

$$\tilde{I}_{q,n} \leq I_{q,n+(\tilde{S}_q - S_q)}, \tag{47}$$

$$\tilde{O}_{i,n} \leq O_{i,n}, \quad i \in \{1, \ldots, R\} - \{q\}, \tag{48}$$

$$\tilde{O}_{q,n} \leq O_{q,n+(\tilde{S}_q - S_q)}. \tag{49}$$

(3) *If $\tilde{K}_i = K_i$ and $\tilde{S}_i = S_i$ $(i = 1, \ldots, R-1)$ and $\tilde{S}_R > S_R$, then for all n*:

$$\tilde{I}_{i,n} \leq I_{i,n+(\tilde{S}_R - S_R)}, \quad i = 1, \ldots, R, \tag{50}$$

$$\tilde{I}_{R+1,n} \leq I_{R+1,n}, \tag{51}$$

$$\tilde{O}_{i,n} \leq O_{i,n+(\tilde{S}_R - S_R)}, \quad i = 1, \ldots, R. \tag{52}$$

5.5. *Production capacity*

The *production capacity* of a pull control system is the maximum demand rate that the system can meet. To determine the production capacity of a pull control system, we study the *saturated* version of the system, that is the original system under the assumption that there is an infinite number of raw parts and customer demands. The production capacity of the original system is then the throughput of the saturated system.

Property 11 and Property 13 that follow are important since they state that the production capacity of the SEKCS and of the IEKCS depends only on one parameter, namely K_i. The roles of parameters S_i (base stock level) and K_i (number of kanbans) are thus clearly distinct. Parameters S_i are related to the satisfaction of demands, whereas parameters K_i are related to the production of new parts. Thus, parameters K_i could be designed first to obtain a desirable production capacity and parameters S_i could be designed subsequently to obtain a desirable customer satisfaction level (Dallery and Liberopoulos, 2000)

Property 12 and property 14 that follow state that in the saturated case, the SEKCS (respectively the IEKCS) and the SKCS (respectively the IKCS) are equivalent.

5.5.1. *Production capacity of the SEKCS*

Figure 6 shows the queueing network model of the saturated SEKCS having R cells ($R-1$ manufacturing cells and a single assembly cell). Figure 6 is obtained from Fig. 4 as follows:

- By definition of the saturated SEKCS, queues $P_{0,i}$, $(i = 1, \ldots, R-1)$ have an infinite number of raw

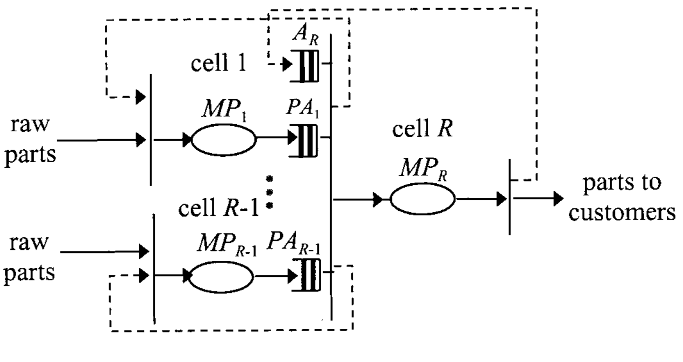

Fig. 6. Queueing network model for the saturated SEKCS.

parts, and queues $D_i, (i = 1, \ldots, R+1)$ have an infinite number of demands.
- Therefore, these queues play no role in the synchronization station they belong to since they never block the transfer of parts through that synchronization station, hence, they can be eliminated.
- Once $P_{0,i}$ and D_i have been eliminated, A_i remains as the only queue in the synchronization station at the entry of cell i $(i = 1, \ldots, R-1)$. Similarly, once D_{R+1} has been removed, PA_R remains as the only queue in the synchronization station at the output of cell R. Clearly, if a synchronization station is fed by only one queue, this queue can be removed since any customer arriving at this queue immediately goes through the synchronization station. In the saturated SEKCS, queues A_i $(i = 1, \ldots, R-1)$ and PA_R can therefore be eliminated.

The queueing network model that results after these eliminations is shown in Fig. 6. We now have the following two properties.

Property 11. *The production capacity of the SEKCS depends only on parameters K_i $(i = 1, \ldots, R)$, and is independent of S_i $(i = 1, \ldots, R)$.*

The proof is based on the following result. The throughput of a basic FJQN/B containing N elementary closed subnetworks depends only on the fixed number of customers in each closed subnetwork, and not on the initial allocation of these customers along the queues of this closed subnetwork (Dallery *et al.*, 1994, 1997). In the SEKCS, K_i is the fixed number of customers in the closed subnetwork that includes A_i, MP_i, and PA_i, whereas S_i determines the initial allocation of customers in each of the queues of the closed subnetwork.

Property 12. *The production capacity of the SEKCS, with parameters K_i and S_i $(i = 1, \ldots, R)$, is equal to the production capacity of the SKCS with the same parameters K_i $(i = 1, \ldots, R)$, as those in the SEKCS.*

Indeed, by comparing Fig. 6 and the queueing network model for the saturated SKCS, it is clear that the saturated SEKCS is equivalent to the saturated SKCS, and therefore their throughputs are equal to each other.

5.5.2. *Production capacity of the IEKCS*

Figure 7 shows the queueing network model of the saturated IEKCS having R cells ($R-1$ manufacturing cells and a single assembly cell). Figure 7 can be obtained from the queueing network model in Fig. 5 after eliminating:

- queues $P_{0,i}$ and queues D_i and $D_{R,i}$ $(i = 1, \ldots, R-1)$, using the fact that they contain an infinite number of entities;
- queues A_i $(i = 1, \ldots, R-1)$ and queue PA_R, using the fact that they are the only queues feeding a synchronization station.

We now have the following properties. The arguments for their proofs are similar to those used for the SEKCS and are therefore omitted.

Property 13. *The production capacity of the IEKCS depends only on parameters K_i $(i = 1, \ldots, R)$, and is independent of S_i $(i = 1, \ldots, R)$.*

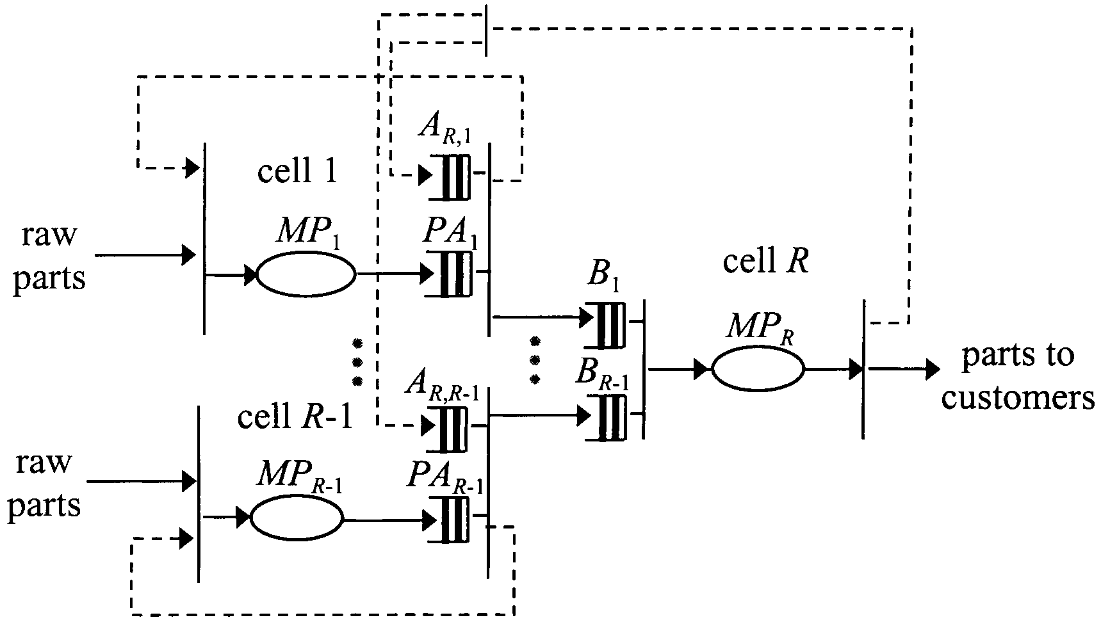

Fig. 7. Queueing network model for the saturated IEKCS.

Property 14. *The production capacity of the IEKCS, with parameters K_i and S_i $(i = 1, \ldots, R)$, is equal to the production capacity of the IKCS with the same parameters K_i $(i = 1, \ldots, R)$, as those in the IEKCS.*

6. A comparison between the SEKCS and the IEKCS

An intuitive comparison between the queueing network model of the SEKCS shown in Fig. 4 and the queueing network model of the IEKCS shown in Fig. 5 leads to the conjecture that the IEKCS responds faster to customer demands than does the SEKCS with the same parameters. This is due to the mutual independence of operation of the manufacturing cells upstream of the assembly cell and it is stated more precisely in Property 15 that follows.

To distinguish equivalent times in the two systems, let us introduce the following notation:

$I^S_{i,n}$ and $I^I_{i,n}$ $(i = 1, \ldots, R)$ = the time of the nth arrival in MP_i, in the SEKCS and the IEKCS, respectively;

$I^S_{R+1,n}$ and $I^I_{R+1,n}$ = the time of the nth delivery to a customer, in the SEKCS and the IEKCS, respectively;

$O^S_{i,n}$ and $O^I_{i,n}$ $(i = 1, \ldots, R)$ = the time of the nth departure from MP_i, in the SEKCS and the IEKCS, respectively;

$D^S_{i,n}$ and $D^I_{i,n}$ = the time of the nth demand arrival, in the SEKCS and the IEKCS, respectively.

Property 15. *Consider two systems, the SEKCS and the IEKCS, having the same parameters K_i and S_i, the same sequence of service times $\sigma_{i,n}$ and the same customer demand times D_n for all $i = 1, \ldots R$. Then*:

$$I^I_{i,n} \leq I^S_{i,n}, \quad i = 1, \ldots, R+1, \tag{53}$$

$$O^I_{i,n} \leq O^S_{i,n}, \quad i = 1, \ldots, R. \tag{54}$$

The proof is similar to that of Property 9 and can be found in Chaouiya *et al.* (1998)

Property 15 states that the time at which the nth part begins its processing (respectively finishes its processing) in MP_i in the IEKCS is smaller than the time when the nth part begins its processing (respectively finishes its processing) in MP_i in the SEKCS. Therefore, customer demands are satisfied earlier in the IEKCS than they are in the SEKCS. This does not necessarily mean that the IEKCS has an overall better performance than the SEKCS, since the inventory storage costs are not taken into account. In fact, the IEKCS is likely to incur higher inventory storage than does the SEKCS.

7. Conclusions

The EKCS is a combination of the Base Stock Control System (BSCS) and the Kanban Control System (KCS) and it includes both systems as special cases. As such, it performs no worse and is in any case more robust than either of the two systems. The EKCS depends on two parameters per cell, each parameter having a distinct role: the number of kanbans is used basically to limit the WIP in the cell, and the base stock is used basically to provide a buffer against stockouts, which are associated with bad customer service.

We extended the EKCS to assembly systems, defining two policies and proving some of their properties. The difference between the two policies rests on whether component parts and kanbans are released simultaneously or independently prior to assembly. For simplicity, we restricted our study to assembly systems having $(R - 1)$ manufacturing cells supplying a single assembly cell. The results in this paper, however, can be easily extended to more general assembly topologies. For instance, queues $P_{0,i}$ in Figs. 4 and 5 may be transformed into queues $PA_{0,i}$ containing finished parts and kanbans from previous cells. In this case, when a part in $PA_{0,i}$ is transferred downstream, a kanban is released to the previous cell, in the same way as it happens in cells $i = 1, \ldots, R - 1$, or in cell R.

We assumed that each assembled part is made by putting together one of each of the component parts. A generalization of this is straightforward, and can be performed in two ways. One way is to consider the case where the assembly cell requires n items from manufacturing cell i, so that the latter cell produces batches of n items, each batch being associated with one kanban. Another way is to consider that n parts of manufacturing cell i are released into the assembly cell, one at a time, and that n kanbans are transferred upstream to cell i.

An important question when implementing a pull control policy is how to define its parameters to achieve a good trade off between the customer service levels and the inventory costs. For this purpose, performance evaluation of the EKCS for assembly systems should be performed using simulation or analytical approximate techniques.

References

Baccelli, P., Cohen, G., Olsder, G.J. and Quadrat, J.-P. (1992) *Synchronization and Linearity: an Algebra for Discrete Event Systems*, John Wiley & Sons, New York

Baccelli, F. and Liu, Z. (1992) Comparison properties of stochastic decision free Petri nets. *IEEE Transactions on Automatic Control*, **37**(12), 1905–1920.

Buzacott, J.A. (1989) Queueing models of kanban and MRP controlled production systems. *Engineering Cost and Production Economics*, **17**, 3–20.

Buzacott, J.A. and Shanthikumar, J.G. (1993) *Stochastic Models of Manufacturing Systems*, Prentice-Hall, Englewood Cliffs, NJ.

Chaouiya, C., Liberopoulos, G. and Dallery, Y. (1998) The extended kanban system for production control of assembly systems. Rapport de Recherche LIP6 1998/024, Laboratoire d'Informatique de Paris 6.

Dallery, Y. and Liberopoulos, G. (2000) Extended kanban control system: combining kanban and base stock. *IIE Transactions on Design and Manufacturing*, **32**, 369–386.

Dallery, Y., Liu, Z. and Towsley, D. (1994) Equivalence, reversibility, symmetry and concavity in fork/join queueing networks with blocking. *Journal of the ACM*, **41**(5), 903–942.

Dallery, Y., Liu, Z. and Towsley, D. (1997) Properties of fork/join queueing networks with blocking under various operating mechanisms. *IEEE Transactions on Robotics and Automation*, **13**(4), 503–518.

Di Mascolo, M. and Dallery, Y. (1996) Performance evaluation of kanban controlled assembly systems, in *Proceedings of the Symposium on Discrete Events and Manufacturing Systems of the Multiconference on Computational Engineering in Systems Applications (CESA'96, IMACS)*, Lille, France.

Hopp, W.J. and Spearman, M.L. (1996) *Factory Physics*, Irwin/McGraw Hill, New York, NY.

Liberopoulos, G. and Dallery, Y. (2000) A unified framework for pull control mechanisms in multi-stage manufacturing systems. *Annals of Operations Research*, Vol 93: Performance evaluation and optimization of production lines, MacGregor Smith, J., Gershwin, S.B. and Papadopoulos, C.T. (eds), Baltzer Science, Bussum, The Netherlands, pp. 325–355.

Monden, Y. (1983) *Toyota Production System: Practical Approach to Production Management*, Industrial Engineering and Management Press, Narcross, GA.

Sbiti, N., Di Mascolo, M. and Amghar, M. (1999) Analysis of base-stock controlled assembly systems, in *Proceedings of the 2nd Aegean Conference on Analysis and Modeling of Manufacturing Systems*, Editions Ziti, Thessaloniki Greece, pp. 141–151.

Biographies

Claudine Chaouiya received her Ph.D. from the Université de Nice Sophia-Antipolis, in 1992. She is currently an Assistant Professor in Computer Science at the Ecole Superieure d'Ingenieurs de Luminy (ESIL). In 1993–1994, she was a Visiting Professor at the Universidade Federal do Ceara (Brasil), in 1994–1999 she was an Assistant Professor at the Université de Nouvelle Caledonie. Her research interests are in modelling and analysis of discrete event systems, with applications in manufacturing and communication networks.

George Liberopoulos received his B.S. and M.Eng. from Cornell University in 1985 and 1986, respectively, and his Ph.D. from Boston University in 1993. In 1994–96 he was a visiting scientist at Universite Paris IV. He is currently an Assistant Professor of Production Management at the University of Thessaly. His research interests are in operations management with an emphasis on the analysis and control of manufacturing systems. He serves at the Editorial Board of *IIE Transactions on Design and Manufacturing*.

Yves Dallery received his Ph.D. and the degree of Habilitation a Diriger des Recherches from the Institut National Polytechnique de Grenoble (INPG) in 1984 and 1989, respectively. He is currently Professor of Manufacturing and Logistics at Ecole Centrale de Paris. Before that, he was Directeur de Recherche at the Centre National de la Recherche Scientifique (CNRS). In 1984–1985, he was a post-doctoral fellow at Harvard University. In 1991–1992, he was a visiting scientist at M.I.T. and in 1992–1993 he was an Associate Professor of Manufacturing Engineering at Boston University. His research interests are in operations management, supply chain management, and stochastic models.