The Extended Kanban System for Production Control of Assembly Systems

C. Chaouiya[*], G. Liberopoulos[**], Y. Dallery[***]

Abstract : In many manufacturing systems, production of parts proceeds in stages. An important managerial concern is how to control the flow of parts through the stages. In many systems this is done by implementing a pull control policy, that is, a policy that decides when to produce parts 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 stages in series. Different pull control mechanisms have been proposed in the literature, among which the Base Stock Control System (BSCS) the Kanban Control System (KCS) and the Generalized Kanban Control System (GKCS) are of special interest. Recently a new control mechanism referred to as the Extended Kanban Control System (EKCS) was introduced in [7]. The purpose of this paper is to generalize the EKCS to assembly structures. It turns out that we need to define two different control policies depending on whether parts are transferred simultaneously or independently into the assembly stage. This leads to the definition of the Simultaneous Extended Kanban Control System (SEKCS) and the Independent Extended Kanban Control System (IEKCS). Properties and comparisons of these two control mechanisms are presented.

[*] Universite Française du Pacifique, CUNC
BP4477, 98847 Noumea - NEW CALEDONIA
chaouiya@ufp.nc

[**] University of Thessaly, Department of Mechanical and Industrial Engineering
Pedion Areos
GR-38334 Volos - GREECE
glib@uth.gr

[***] LIP6, Laboratoire d'Informatique de Paris 6
Universite Pierre et Marie Curie - CNRS
4, Place Jussieu - F-75252 Paris Cedex 05 - FRANCE
Yves.Dallery@lip6.fr

1. Introduction

In many manufacturing systems, production of parts proceeds in stages. Each stage 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 or a manufacturing cell). It contains parts which are currently being processed (referred to as the *Work In Process,* (WIP) of the stage). The output buffer is a storage area that contains parts that have completed processing in the stage (referred to as *finished parts* of the stage). The manufacturing system is fed by raw parts, and releases finished parts to customers.

An important managerial concern is how to control the flow of parts through the stages. In many systems this is done by implementing a pull control policy, that is, a policy that decides when to produce parts 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 stages 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 proved to be of high interest [4]. The BSCS is a classical mechanism borrowed from inventory theory, while the KCS was invented by Toyota and has since been widely used in industry. The main advantage of these control policies is that they are very simple to understand and implement. In particular, they depend on only one parameter per stage. Therefore, to define a policy of this type it suffices to fix this parameter for each stage of the system. However, it has also been found that these policies do not always achieve a good trade-off between inventory costs and customer service levels. As a result, several authors have proposed more general pull control policies with the goal of leading to better trade-offs. Among these, the Generalized Kanban Control System (GKCS) introduced in [3] and studied in detail in [4] and the Extended Kanban Control System (EKCS) proposed in [7] are of special interest. Both of these policies are characterized by two parameters per stage. A general discussion and comparison of these four control policies can be found in [7]. It appears that the EKCS has some important advantages over the GKCS.

For industrial applications, it is important to extend these control policies to systems having more general structures, in particular assembly systems. In this case, a stage may have more than one immediat upstream stage. Although this situation is highly encountered in industry, little work has been done in this area. In [8] the KCS is extended to assembly systems. One important additional feature arises (with respect to the serial system case), namely the definition of the way parts are released into the assembly stage. Two cases can be considered, namely a simultaneous transfer mechanism or an independent transfer mechanism leading to the definition 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 structures. As for the KCS we need to define two different control policies depending on whether parts are transferred simultaneously or independently into the assembly stage. This leads to the definition of the Simultaneous Extended Kanban Control System (SEKCS) and the Independent Extended Kanban Control System (IEKCS).

The rest of the paper is organized as follows. Section 2 briefly describes the EKCS defined in [7] in the case of serial systems. In section 3, we define the generalization of the EKCS to assembly systems, successively describing the SEKCS and the IEKCS. Properties of these two control mechanisms are given in section 4. These properties are pertaining to invariants, bounds, evolution equations, the influence of the variation of the parameters and production capacity. Finally, the two mechanisms are compared in section 5. For the sake of completeness, the generalizations of the BSCS and the KCS to assembly systems are discussed in Appendix A.

2. Extended Kanban Control System: stages in series

The EKCS has recently been introduced by Y.Dallery and G.Liberopoulos in [7]. This new kanban based control system appears to be an interesting alternative to other pull controlled systems. The EKCS has the following features:

- simplicity,
- limitation of the WIP in each stage (unlike in the Base Stock Control System),
- immediate transfer of demands to all stages of the system (unlike in the Kanban Control System),
- clearly separated role of the parameters, number of kanbans and base stock level (unlike in the Generalized Kanban Control System).

In this section, we briefly describe the Extended Kanban Control System for a manufacturing system having stages in series. For a detailed description and properties, refer to [7].
Figure 2-1 describes a manufacturing system having N stages in series. Each stage may be seen as a manufacturing process (single machine, production line, flexible manufacturing cell, job-shop, etc.) with an output buffer.

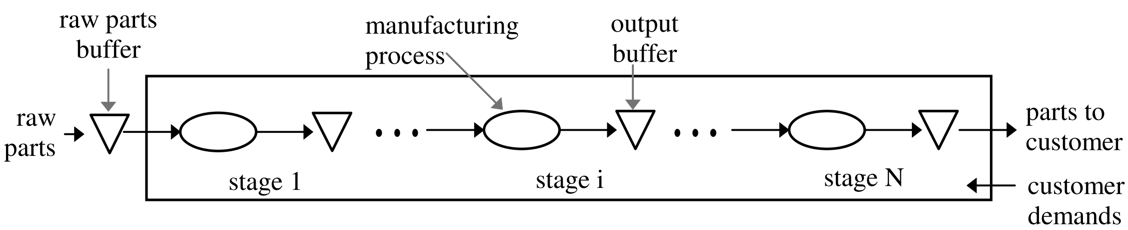

Figure 2-1: Manufacturing system with N stages in series

The EKCS is a pull control mechanism that can be viewed as a combination of the Base Stock Control System (BSCS) and the Kanban Control System (KCS) (see [4], [7] and Appendix A). Figure 2-2 shows the queueing network model of an EKCS having N stages in series.

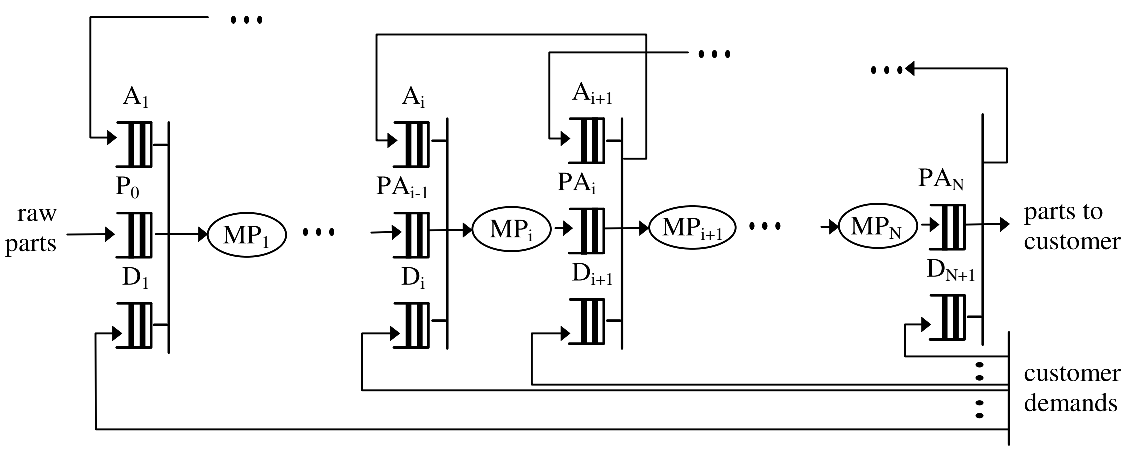

Figure 2-2: Queueing network model of the EKCS

We use the following notation introduced in [7]:

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

Table 2-1 describes the contents and initial values of the queues, or network of queues in the case of MP_i, in Figure 2-2:

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

Table 2-1: Contents and initial values of the queues in the EKCS

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 stage i has K_i kanbans a_i that authorize the production of stage-i finished parts. Initially, in stage 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 stages by adding 1 to the contents d_i of queue D_i ($i=1,...,N+1$).

The behavior of the system can be described as follows.

Release of parts into the manufacturing stages:
At the 1st manufacturing stage, queues P_0, A_1, and D_1 are joined in a synchronization station. Raw parts in P_0 do not have any kanbans attached to them. Therefore the stage 1 can begin processing a part as soon as there are 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,

– the kanban a_1 is attached onto p_0 which is relabelled q_1, and together they are transferred downstream to MP_1 as a pair (q_1,a_1), and
– the demand d_1 is satisfied and is therefore discarded.

The i^{th} manufacturing stage ($i=2,...,N$) can begin processing a part only when there are at least: a pair (p_{i-1},a_{i-1}) in PA_{i-1}, an authorization card a_i in A_i and a demand d_i in D_i. When these conditions are satisfied, then,

– the kanban a_{i-1} is detached from p_{i-1} and is transferred upstream to A_{i-1},
– the kanban a_i is attached onto p_{i-1} which is relabelled q_i, and together they are transferred downstream to MP_i as a pair (q_i,a_i), and
– the demand d_i is satisfied and is therefore discarded.

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

Delivery of finished parts to the customer:
There is no need for an authorization to release a finished part p_N to the customer. Therefore, the delivery of a finished part can occur as soon as there are a pair (p_N,a_N) in PA_N and a demand d_{R+1} in D_{R+1}. When these conditions are satisfied, then,
– the kanban a_N is detached from p_N and is transferred upstream to A_N,
– the part p_N is released to the customer, and
– the demand d_{N+1} is satisfied and is therefore discarded.

The EKCS has two parameters per stage, K_i and S_i. These parameters must be adjusted to achieve a good compromise between 1) keeping a low inventory of parts in the system and 2) attaining a high level of immediate customer demand satisfaction. The use of kanbans guarantees that the number of parts (WIP plus the finished parts) in each stage is bounded by the number of kanbans in that stage.

3. Extended Kanban Control Systems for Assembly

The EKCS described for the serial system configuration in Figure 2-2 can be generalized to manufacturing systems having assembly system configuration. Figure 3-1 illustrates the topology of a system having assembly stages (stages supplied by several raw parts buffers) and manufacturing stages (stages supplied by a single raw parts buffer). This topology is a tree structure.

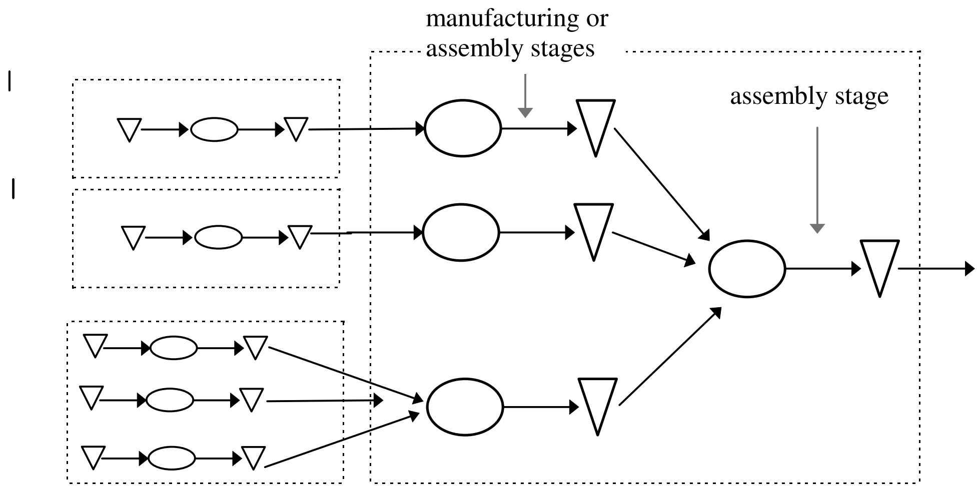

Figure 3-1: General topology for assembly manufacturing systems

For simplicity, we restrict our study to assembly systems having (R-1) manufacturing stages supplying a single assembly stage (Figure 3-2). However, the results in this paper can be easily extended to general topologies.
The assembly stucture we are considering is the classical one. That is, each item of the final product is produced by assembly of one item of each manufacturing stage. The extension of the presentation to situations where more than one item of each manufacturing stage 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.

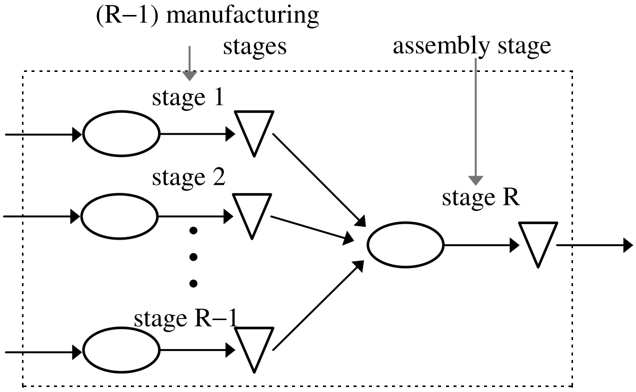

Figure 3-2: R-1 manufacturing stages supplying a single assembly stage

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 [8]. These mechanisms are the Simultaneous EKCS (SEKCS) and the Independent EKCS (IEKCS). Next, we describe them.

3.1. Simultaneous Extended Kanban Control System (SEKCS): Definition

Figure 3-3 shows the queueing network model for the SEKCS. We represent the case of (R-1) manufacturing stages supplying a single assembly stage.

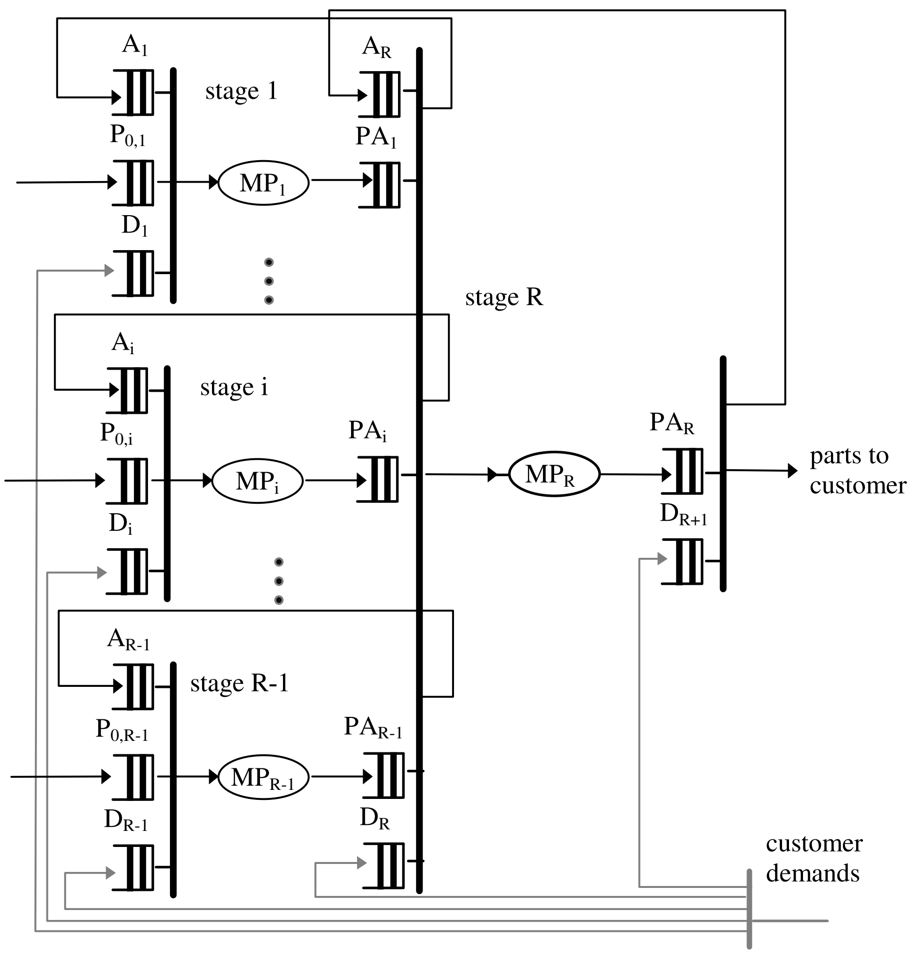

Figure 3-3: Queueing network model for the SEKCS

We use the following notation:

$p_{0,i}$	$i=1,...,R-1$	a stage-i raw part
p_i	$i=1,...,R-1$	a stage-i finished part
p_R		a stage-R assembled part
q_i	$i=1,...,R-1$	a part currently being processed in stage i
q_R		a part currently being assembled in stage R
d_i	$i=1,...,R-1$	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 a p_R
a_i	$i=1,...,R-1$	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

Table 3-2 shows the contents and initial values of the queues or network of queues in the case of MP_i ,

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

Table 3-2: Contents and initial values of the queues in the SEKCS

As was the case in the EKCS, queue $P_{0,i}$ ($i=1,...,R-1$) represents the raw parts buffer supplying manufacturing stage 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 stages by adding 1 to the contents d_i of queue D_i ($i=1,...,R+1$).

The behavior of the SEKCS can be described as follows.

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

– the kanban a_i is attached onto $p_{0,i}$ which is relabelled q_i, and together they are transferred downstream to MP_i as a pair (q_i,a_i), and
– the demand d_i is satisfied and is therefore discarded.

When the part q_i finishes its processing in MP_i, it is relabelled p_i, and, together with the 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 stage:
At the assembly stage R, queues PA_i (i=1,...,R-1), A_R, and D_R are joined in a synchronization station. This means that the assembly operation can begin only when there are at least: a pair (p_i,a_i) in PA_i for every i=1,...,R−1, an authorization a_R in A_R and a demand d_{R+1} in D_{R+1}. When these conditions are satisfied, then,

– the kanbans a_i are <u>simultaneously</u> detached from the p_i (i=1,...,R-1) and are transferred upstream to their corresponding previous stage,
– the kanban a_R is attached onto $(p_1,p_2, ..., p_{R-1})$ which is relabelled q_R, and together they are transferred downstream to MP_R as a pair (q_R,a_R),
– the demand d_R is satisfied and is therefore discarded.

When a part q_R finishes its assembly process in MP_R, it is relabelled 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 stage, queues PA_R and D_{R+1} are joined in a synchronization station. There is no need for authorization to release a finished part to the customer. Therefore, the delivery of a finished assembled part can occur as soon as there are 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,

– the kanban a_R is detached from p_R and is transferred upstream to A_R,
– part p_R is released to the customer,
– the demand d_{R+1} is satisfied and is therefore discarded.

3.2. Independent Extended Kanban Control System (IEKCS): Definition

Figure 3-4 shows the queueing network model for the IEKCS in the case of (R-1) manufacturing processes supplying a single assembly process.

We use the following notation:

$p_{0,i}$	i=1,...,R-1	a raw part for stage-i
p_i	i=1,...,R-1	a stage-i finished part
p_R		a stage-R assembled part
q_i	i=1,...,R-1	a part currently being processed in stage i
q_R		a part currently being assembled in stage R
d_i	i=1,...,R-1	a demand for the production of a new p_i
$d_{R,i}$	i=1,...,R-1	a demand for the assembly of a new p_R using a stage-i finished part p_i
d_{R+1}		a demand for a p_R
a_i	i=1,...,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
$a_{R,i}$	i=1,...,R-1	an authorization card (issued from some a_R) for the assembly of a new p_R using a stage-i finished part p_i

Table 3-3 describes the contents and initial values of the queues, or network of queues in the case of MP_i .

As was the case in the SEKCS, queues $P_{0,i}$ (i=1,...,R-1) represent the raw parts buffer supplying the manufacturing stage 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 stages by adding 1 to the contents d_i of queue D_i (i=1,...,R+1) and by adding 1 to the contents $d_{R,i}$ of queue $D_{R,i}$ (i=1,...,R-1).

Queue	Contents	Initial value
MP_i i=1,...,R-1	(q_i,a_i)	0
MP_R	(q_R,a_R)	0
PA_i i=1,...,R-1	(p_i,a_i)	S_i
PA_R	(p_R, a_R)	S_R
A_i i=1,...,R-1	a_i	K_i-S_i
$A_{R,i}$ i=1,...,R-1	$a_{R,i}$	K_R-S_R
B_i i=1,...,R-1	$(p_i,a_{R,i})$	0
D_i i=1,...,R-1	d_i	0
$D_{R,i}$ i=1,...,R–1	$d_{R,i}$	0
D_{R+1}	d_{R+1}	0

Table 3-3: Contents and initial values of the queues in the IEKCS

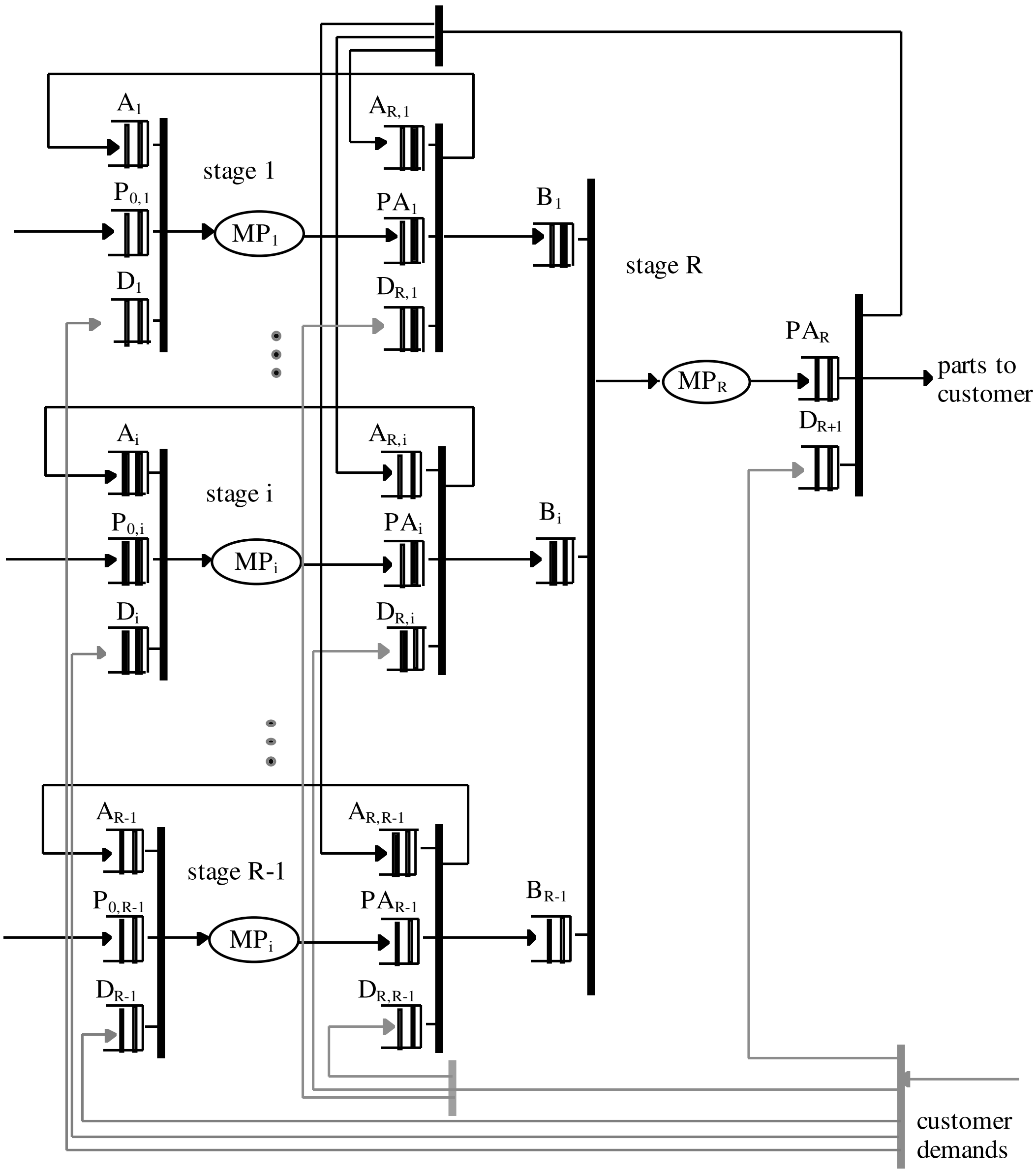

Figure 3-4: Queueing network model for the IEKCS

The behavior of the IEKCS can be described as follows.

Release of parts into the manufacturing stages: It is identical to the SEKCS.

Release of parts into the assembly stage:
The difference between the SEKCS and the IEKCS is in the way kanbans are transferred in stages 1,...,R-1. In the SEKCS all kanbans are transferred simultaneously, whereas in the IEKCS they can be transffered <u>independently</u> of each other. In the IEKCS, each demand for an assembly operation is split into R-1 demands $d_{R,i}$ (i =1,...,R-1). In the same way, each stage-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 stages 1,...,R-1 and the assembly stage R there are 2 layers of synchronization stations. The first layer consists of R-1 synchronization stations in parallel, one for each manufacturing stage, and the second layer consists of a single synchronization station fed by the synchronization stations of the first layer. More precisely:

1. For each manufacturing stage i (i=1,...,R-1), queues PA_i, $A_{R,i}$, and $D_{R,i}$ are joined in a synchronization station. The supply of stage-i finished parts for the assembly operation can occur only when there are at least: a pair (p_i,a_i) in PA_i, an authorization $a_{R,i}$ in $A_{R,i}$, a demand $d_{R,i}$ in $D_{R,i}$. When these conditions are met, then,
   – the kanban a_i is detached from p_i and is transferred upstream to A_i,
   – the kanban $a_{R,i}$ is attached onto p_i, and together they are transferred downstream to B_i as a pair $(p_i,a_{R,i})$, and
   – the demand $d_{R,i}$ is satisfied and is therefore discarded.
2. Queues B_i (i=1,...,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:
   – the pairs $(p_i,a_{R,i})$, are removed from queues B_i (i =1,...,R-1), the (R-1)-tuple $(p_1,p_2,...,p_{R-1})$ is relabelled q_R, the kanbans $(a_{R,1},a_{R,2},...,a_{R,R-1})$ are merged into a kanban a_R, and the 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 relabelled 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:
Identical to the SEKCS except that when the kanban a_R is transferred back to the input of the assembly stage, it is split into R-1 kanbans $a_{R,1},a_{R,2},...,a_{R,R-1}$, and kanban $a_{R,i}$ joins queue $A_{R,i}$, i=1,...,R-1.

4. Properties of the SEKCS and the IEKCS

In this section, we present some basic properties of the SEKCS and IEKCS. It is indeed important to get a good understanding, as well as some insights, of the behavior of the two mechanisms. The purpose is twofold: 1) understand the behavior of each control mechanism and in particular the influence of their parameters (the K_i's and the S_i's); and 2) compare the behavior of the SEKCS and the IEKCS to emphasize the specificity of each one and how the behavior of one relates to the behavior of the other. We will establish some relations on the population of the queues which are always valid, independent of the time (section 4.1). Such relations are called invariants of the system, they express useful relationships between queues. Moreover, bounds will be derived which in particular imply the limitation of the WIP and of the number of finished parts in each stage (section 4.2). We will also show special cases for which the Extended Kanban Control System reduces to the traditional Kanban Control System and to the Base Stock Control System.

Both the SEKCS and the IEKCS can be modeled as Fork-Join Queuing Networks with Blocking (FJQN/B) as defined in [5]. FJQN/B 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). Figure 3-3 and Figure 3-4 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 only operators '+' and 'max'.

These results are similar to those in [2], [5] and [6]. They are based on the equivalence of FJQN/Bs to Strongly Connected Marked Graphs (SCMGs).
We will denote by M(Q) the current population of any queue Q[1] in the system. We first state some basic relations for the SEKCS and the IEKCS.

By definition of the initial state of the SEKCS, queue A_i has $(K_i$-$S_i)$ free kanbans (i=1,...,R), where clearly $K_i - S_i \geq 0$. The two parameters of stage i (i=1,...,R) are therefore constrained by:

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

By definition of a synchronization station, at all times, at least one of the queues in a synchronization station must be 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:

Eq. 1 $\quad M(A_i).M(P_{0,i}).M(D_i) = 0 \qquad i=1,\ldots,R-1,$

Eq. 2 $\quad M(A_R).\left(\prod_{i=1}^{R-1} M(PA_i)\right).M(D_R) = 0,$

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

Similarly, by the definition of the initial state of the IEKCS, queue A_i has $(K_i$-$S_i)$ free kanbans (i=1,...,R-1), and queue $A_{R,i}$ has $(K_R$-$S_R)$ free kanbans (i=1,...,R-1). Therefore, we have:

$$K_i \geq S_i, \qquad i=1,\ldots,R,$$

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

Eq. 4 $\quad M(A_{R,i}).M(PA_i).(D_{R,i}) = 0 \qquad i=1,\ldots,R-1,$

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

[1] M(Q) varies with time and should also be a function of time. However, for simplicity, and insofar as we are interested in invariants or instantaneous relations, we will omit this dependence on time.

4.1. 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.

4.1.1. Invariants for the SEKCS

Property 1: In the SEKCS, the following holds:

Eq. 6 $M(A_i) + M(MP_i) + M(PA_i) = K_i$ $\quad i=1,...,R-1,$

Eq. 7 $M(A_i) - M(D_i) + M(D_R) = K_i-S_i$ $\quad i=1,...,R-1,$

Eq. 8 $M(PA_i) - M(D_R) + M(MP_i) + M(D_i) = S_i$ $\quad i=1,...,R-1,$

Eq. 9 $M(A_R) + M(MP_R) + M(PA_R) = K_R,$

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

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

Proof: When the SEKCS is in its initial state, there are (K_i-S_i) kanbans a_i in queue A_i, S_i pairs (p_i,a_i) in queue PA_i, and MP_i is empty (i=1,...,R-1). Therefore, Eq. 6 holds initially. By observing the events that can modify the contents of A_i, MP_i and PA_i , it is clear that, as the SEKCS evolves starting from its initial state, Eq. 6 remains true, since:

- when a kanban leaves A_i, it is attached onto a part $p_{0,i}$, and together they join MP_i as a pair (q_i, a_i),
- when a pair (q_i, a_i) leaves queue MP_i, it is transferred to PA_i as a pair (p_i, a_i),
- when a pair (p_i, a_i) leaves queue PA_i, the kanban a_i is transferred to A_i .

When the SEKCS is in its initial state, there are (K_i-S_i) kanbans a_i in queue A_i (i=1,...,R-1) and queue D_i is empty, i=1,...,R. Therefore, Eq. 7 holds initially. As the SEKCS evolves starting from its initial state, Eq. 7 remains true since:

- when a kanban is transferred to A_i, a demand leaves D_R,
- when a kanban leaves A_i, a demand also leaves D_i,
- when a demand joins D_i, a demand also joins D_R.

To prove Eq. 8, it suffices to substitute $M(A_i)$ from Eq. 6 into Eq. 7.

Eq. 9, Eq. 10 and Eq. 11 can be proved using similar arguments.

■

4.1.2. Invariants for the IEKCS

Property 2: In the IEKCS, the following holds:

Eq. 12 $M(A_i) + M(MP_i) + M(PA_i) = K_i$ $\quad i=1,...,R-1,$

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

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

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

Eq. 16 $M(A_{R,i}) + M(B_i) + M(MP_R) + M(PA_R) = K_R$ $i=1,...,R-1$,

Eq. 17 $M(A_{R,i}) + M(D_{R+1}) - M(D_{R,i}) = K_R-S_R$ $i=1,...,R-1$,

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

Proof: Eq. 12, Eq. 13, Eq. 14, are similar to Eq. 6, Eq. 7, Eq. 8 respectively, and can be proved in the same way.

When the IEKCS is in its initial state, queue $A_{R,i}$ contains (K_R-S_R) kanbans, and B_i is empty ($i=1,...,R-1$). Consequently, Eq. 15 holds at the initial state. By observing the events that can modify the state of queues $A_{R,i}$ and B_i ($i=1,...R-1$), it is clear that, as the IEKCS evolves starting from its initial state, Eq. 15 remains true since:

- when a kanban a_R is released from PA_R, it is split into R-1 kanbans which are simultaneously transferred to their respective queues $A_{R,i}$. In other words, arrivals in queues $A_{R,i}$ are simultaneous, and
- departures from queues B_i ($i=1,...,R-1$) are also simultaneous, and
- when a kanban $a_{R,i}$ leaves $A_{R,i}$, it joins B_i.

When the IEKCS is in its initial state, there are (K_R-S_R) kanbans $a_{R,i}$ in $A_{R,i}$ ($i=1,...,R-1$), S_R pair (p_R,a_R) in queue PA_R, and queues B_i ($i=1,...,R-1$) and MP_R are empty. Therefore Eq. 16 is true at the initial state. It remains true thereafter since, as the IEKCS evolves starting from the initial state, we have:

- when a kanban $a_{R,i}$ leaves $A_{R,i}$, it is attached onto a part p_i and transferred to B_i,
- when a kanban leaves queue B_i, one pair (q_R,a_R) is transferred to MP_R,
- when a pair (q_R,a_R) leaves MP_R, a pair (p_R,a_R) joins PA_R, and
- when a pair (p_R,a_R) leaves PA_R, a kanban a_R is split into R-1 kanbans which join separately queues $A_{R,i}$ ($i=1,...,R-1$) and one part is delivered to the customer.

At the initial state, there are (K_R-S_R) kanbans $a_{R,i}$ in each $A_{R,i}$ ($i=1,...,R-1$), and queues $D_{R,i}$ ($i=1,...,R-1$) and D_{R+1} are empty. Therefore Eq. 17 is true initially. It remains true since, as the IEKCS evolves starting from the initial state:

- when a kanban $a_{R,i}$ leaves $A_{R,i}$, a demand also leaves $D_{R,i}$,
- when a demand is transferred to $D_{R,i}$, a demand also arrives in D_{R+1}, and
- when a kanban arrives into $A_{R,i}$ a demand is discarded from D_{R+1}.

To prove Eq. 18, it is enough to substitute $M(A_{R,i})$ from Eq. 16 into Eq. 17.

■

4.2. Bounds

Next, we present bounds which express that the work in process (WIP) is limited in each stage.

Property 3: In the SEKCS, the following holds:

Eq. 19
$$\left| \begin{array}{ll} 0 \le M(A_i) \le K_i & i=1,...,R, \\ 0 \le M(PA_i) \le K_i & i=1,...,R, \\ 0 \le M(MP_i) \le K_i & i=1,...,R, \\ 0 \le M(MP_i) + M(PA_i) \le K_i & i=1,...,R. \end{array} \right.$$

Proof: Eq. 19 directly follows from Eq. 6 and Eq. 9 for the SEKCS since all the quantities in these equations are non-negative.

■

Property 4: In the IEKCS, the following holds for all manufacturing stages:

Eq. 20
$$\begin{cases} 0 \le M(A_i) \le K_i & i=1,...,R-1, \\ 0 \le M(PA_i) \le K_i & i=1,...,R-1, \\ 0 \le M(MP_i) \le K_i & i=1,...,R-1, \\ 0 \le M(MP_i) + M(PA_i) \le K_i & i=1,...,R-1. \end{cases}$$

and at the assembly stage,

Eq. 21
$$\begin{cases} 0 \le M(B_i) \le K_R & i=1,...,R\text{-}1, \\ 0 \le M(A_{R,i}) \le K_R & i=1,...,R\text{-}1, \\ 0 \le M(MP_R) \le K_R & \\ 0 \le M(PA_R) \le K_R & \\ 0 \le M(B_i) + M(MP_R) + M(PA_R) \le K_R & i=1,...,R\text{-}1. \end{cases}$$

Proof: Eq. 20 follows directly from Eq. 12, and Eq. 21 follows directly from Eq. 16, since all the quantities in these two equations are non-negative.

■

Property 5: In the SEKCS the following holds:

Eq. 22 $\quad M(PA_i) - M(D_R) \le S_i \qquad i=1,...,R\text{-}1,$

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

Proof: Eq. 22 follows from Eq. 8, and Eq. 23 follows from Eq. 11.

■

Property 6: In the IEKCS, the following holds:

Eq. 24 $\quad M(PA_i) - M(D_{R,i}) \le S_i \qquad i=1,...,R\text{-}1,$

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

Proof: Eq. 24 follows from Eq. 14 and Eq. 25 follows from Eq. 18.

■

4.3. 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 systems can be described by recursive evolution equations that utilize the operators "+" and "max" only. These evolution equations are of interest because they will allow to establish some useful properties on the behavior of the SEKCS and the IEKCS (section 4.4) and of their comparison (section 5). Moreover, this approach is very general since it is a sample-path approach that does not require any assumption on the distribution of the random variables (processing times, 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,...R-1. The results that follow, however, could be extended to incorporate external arrival processes of parts at the expense of more tedious derivations.

4.3.1. Evolution equations for the SEKCS

Let us introduce the following notation for the times of certain events:

$I_{i,n}$	the time of the n^{th} arrival in MP_i (Input),	i=1,...,R,
$I_{R+1,n}$	the time of the n^{th} departure from the system,	
$O_{i,n}$	the time of the n^{th} departure from MP_i (Output),	i=1,...,R,
D_n	the time of the n^{th} demand arrival (Demand).	

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

Clearly, the following holds[2]:

$$I_{i,n-m} \le I_{i,n} \quad i=1,...,R+1,\ n,m \in \{1,2,...\},$$
$$O_{i,n-m} \le O_{i,n} \quad i=1,...,R,\ n,m \in \{1,2,...\},$$
$$D_{n-m} \le D_n \quad n,m \in \{1,2,...\}.$$

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,

$$\text{Eq. 26} \quad I_{i,n} = \max\left(D_n , I_{R,n-(K_i-S_i)} \right) \qquad i=1,...,R-1,$$

$$\text{Eq. 27} \quad I_{R,n} = \max\left(D_n , \max_{i=1...R-1}\left(O_{i,n-S_i} \right) , I_{R+1,n-(K_R-S_R)} \right),$$

$$\text{Eq. 28} \quad I_{R+1,n} = \max\left(D_n , O_{R,n-S_R} \right),$$

$$\text{Eq. 29} \quad O_{i,n} = \sigma_{i,n} + \max\left(I_{i,n} , O_{i,n-1} \right) \qquad i=1,...,R .$$

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

– the n^{th} demand d_i has arrived in D_i,
– the $(n-(K_i-S_i))^{th}$ kanban has arrived in A_i, since there are initially (K_i-S_i) kanbans in A_i.

---

[2] Note that it is natural to assume that $I_{i,n}$, $O_{i,n}$ and D_n are zero for $n \le 0$.

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 Eq. 26.

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

- the n^{th} demand d_R has arrived in D_R,
- the $(n\text{-}S_i)^{th}$ stage-i finished part with its attached kanban has arrived in PA_i (i=1,...,R-1), since there are initially S_i pairs (p_i,a_i) in PA_i,
- the $(n\text{-}(K_R\text{-}S_R))^{th}$ kanban a_R has arrived in A_R, since there are initially (K_R-S_R) kanbans in A_R.

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

- the n^{th} demand has arrived in D_n,
- the $(n\text{-}S_R)^{th}$ finished part p_R with its attached kanban has been released in PA_R, since there are initially S_R pairs (p_R,a_R) in PA_R .

Eq. 29 gives an expression for $O_{i,n}$ which is, by definition, the time at which the n^{th} 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 n^{th} pair (q_i,a_i) begins its processing in MP_i plus its processing time $\sigma_{i,n}$. The n^{th} pair (q_i,a_i) begins its processing when the two conditions below are satisfied:

- the n^{th} pair (q_i,a_i) has been released in MP_i,
- the $(n\text{-}1)^{th}$ part has completed its processing at MP_i .

■

4.3.2. Evolution equations for the IEKCS

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

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

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

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

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

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

Proof: Eq. 30 differs from Eq. 26 only in the condition for the release of kanbans a_i. Otherwise, the argument is similar to that for the SEKCS: the n^{th} pair (q_i,a_i) enters MP_i (i=1,...,R-1) when the two conditions below are met:

- the n^{th} demand d_i has arrived in D_i,
- the $(n-(K_i-S_i))^{th}$ kanban has arrived in A_i, since there are initially (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 Eq. 30.

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

Eq. 32 is the same as Eq. 28.

In Eq. 33, we consider the time at which the stage-i n^{th} finished part is transferred to B_i . This event occurs as soon as the following three conditions are met:

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

Finally, Eq. 34 is the same as Eq. 29.

■

4.4. Variation of the parameters

This section studies the influence of varying parameters K_i and S_i (i=1,...,R) on the above mentioned event times and derives some monotonicity properties.
Proofs are similar to the proof of "Stochastic Monotonicity with Respect to the Initial Marking" in [1]. 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 $\mathcal{S}$ and the modified system denoted by $\tilde{\mathcal{S}}$. These two systems differ only through their parameters: the number of kanbans and the base stock level in each stage. 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,...,R). On the other hand, the two systems have the same sequence of customer demand times (denoted by D_n for $\mathcal{S}$, and $\tilde{D}_n$ for $\tilde{\mathcal{S}}$) and the same sequence of processing times (denoted by $\sigma_{i,n}$ and $\tilde{\sigma}_{i,n}$, i=1,...,R).

Part 1 of Property 7 and of Property 8 states that increasing the number of kanbans in some stage decreases the arrival and departure times at each stage of the system.
Parts 2 and 3 state that increasing the base stock in some stage q (q=1,...,R) decreases the arrival and departure times of all other stages. Moreover, the arrival time (respectively the departure time) of the n^{th} part from stage 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 stage q from S_q to $\tilde{S}_q$ has the same effect as of having ($\tilde{S}_q$ -S_q) extra parts enter the stage q and receive processing before the first demand arrives.
Property 7 and Property 8 give a way to improve the "speed" of a SEKCS or IEKCS by increasing the number of kanbans or the base stock level. We will see in section 4.6 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.

4.4.1. Variation of the parameter of the SEKCS

Property 7: 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,...,R}, $\tilde{K}_i = K_i$ for all i in {1,...,R}-{q}, and $\tilde{S}_i = S_i$ for all i in {1,...,R}, then for all n,

Eq. 35 $\quad \tilde{I}_{i,n} \le I_{i,n} \qquad i \in \{1,...,R+1\}$,

Eq. 36 $\quad \tilde{O}_{i,n} \le O_{i,n} \qquad i \in \{1,...,R\}$.

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

Eq. 37 $\quad \tilde{I}_{i,n} \le I_{i,n} \qquad i \in \{1,...,R+1\}-\{q\}$,

Eq. 38 $\quad \tilde{I}_{q,n} \le I_{q,n+(\tilde{S}_q-S_q)}$,

Eq. 39 $\quad \tilde{O}_{i,n} \le O_{i,n} \qquad i \in \{1,...,R\}-\{q\}$,

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

3. If $\tilde{K}_i = K_i$, $i \in \{1,...R\}$, $\tilde{S}_i = S_i$, $i \in \{1,...R-1\}$, and $\tilde{S}_R \ge S_R$, then for all n

Eq. 41 $\quad \tilde{I}_{i,n} \le I_{i,n+(\tilde{S}_R-S_R)} \qquad i \in \{1,...,R\}$,

Eq. 42 $\quad \tilde{I}_{R+1,n} \le I_{R+1,n}$,

Eq. 43 $\quad \tilde{O}_{i,n} \le O_{i,n+(\tilde{S}_R-S_R)} \qquad i \in \{1,...,R\}$.

The proof is given in Appendix B.

4.4.2. Variation of the parameters of 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 7.

Property 8: 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,...,R}, $\tilde{K}_i = K_i$, $i \in \{1,...,R\}-\{q\}$, and $\tilde{S}_i = S_i$, $i \in \{1,...,R\}$, then for all n,

Eq. 44 $\tilde{I}_{i,n} \le I_{i,n} \qquad i \in \{1,...,R+1\},$

Eq. 45 $\tilde{O}_{i,n} \le O_{i,n} \qquad i \in \{1,...,R\}.$

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

Eq. 46 $\tilde{I}_{i,n} \le I_{i,n} \qquad i \in \{1,...,R+1\}-\{q\},$

Eq. 47 $\tilde{I}_{q,n} \le I_{q,n+(\tilde{S}_q-S_q)},$

Eq. 48 $\tilde{O}_{i,n} \le O_{i,n} \qquad i \in \{1,...,R\}-\{q\},$

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

3. If $\tilde{K}_i = K_i$, $\tilde{S}_i = S_i$, $i \in \{1,...,R-1\}$, and $\tilde{S}_R \ge S_R$, then for all n:

Eq. 50 $\tilde{I}_{i,n} \le I_{i,n+(\tilde{S}_R-S_R)} \qquad i \in \{1,...,R\},$

Eq. 51 $\tilde{I}_{R+1,n} \le I_{R+1,n},$

Eq. 52 $\tilde{O}_{i,n} \le O_{i,n+(\tilde{S}_R-S_R)} \qquad i \in \{1,...,R\}.$

4.5. Special Cases

Property 9 and Property 10 that follow express two special cases where the SEKCS (respectively the IEKCS) is equivalent to the Simultaneous Kanban Control System (SKCS), and the Base Stock Control System (BCSC) (respectively to the Independent Kanban Control System (IKCS) and the BSCS), where the SKCS, the BSCS and the IKCS are briefly described in Appendix A.

Property 9: 1) The SEKCS with $K_i = \infty$, $S_i \ge 0$, (i=1,...,R) is equivalent to the BSCS having a base stock of S_i finished parts in stage i, i=1,...,R.

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

Proof: 1) Consider the SEKCS shown in Figure 3-3, with $K_i = \infty$, $S_i \ge 0$, i=1,...,R. Queues A_i have an infinite number of kanbans and therefore 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 queues A_i (i=1,...,R) are eliminated from the network in Figure 3-3, the resulting network is the same as the queueing network model of the BSCS in Figure 6-1 and has the same initial conditions.

2) Consider now the SEKCS shown in Figure 3-3, with $K_i = S_i$, i=1,...,R. In the initial state of the system, since K_i-S_i = 0, there are no available kanbans a_i in queues A_i and all kanbans a_i are attached to parts in PA_i. As in the case of the SKCS, a kanban a_i becomes available in A_i only when finished parts p_i are transferred to MP_R (i=1,...,R-1), and a kanban a_R becomes available only when a finished part p_R is delivered to the customer. Also, since K_i-S_i = 0, Eq. 7 and Eq. 10 imply that $M(D_i) \ge M(A_i)$, i = 1,...,R. Queues D_i, therefore, play no role in the synchronization station they belong to; hence they can be eliminated. Once queues D_i (i=1,...,R), are eliminated from the network in Figure 3-3, the resulting network is the same as

the queueing network model of the SKCS in Figure 6-2, where queues A_i in Figure 3-3 play the same role as queues DA_i in Figure 6-2 (i=1,...,R).

■

Property 10: 1) The IEKCS with $K_i = \infty$, $S_i \geq 0$, (i=1,...,R) is equivalent to the BSCS having a base stock of S_i finished parts in stage i, i=1,...,R.

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

Proof: 1) Consider the IEKCS shown in Figure 3-4, with $K_i = \infty$, $S_i \geq 0$, i=1,...,R. Queues A_i and $A_{R,i}$ have an infinite number of kanbans and therefore 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 queues A_i, $A_{R,i}$ (i=1,...,R-1) are eliminated from the network in Figure 3-4, the resulting network is shown in Figure 4-1. Arrivals in queues $D_{R,i}$ are simultaneous, therefore assembly in MP_R occurs when there is one demand for this operation (which is split into R-1 demands) and one finished part in queue PA_i, i=1,...,R-1. Therefore the queueing network in Figure 4-1 is equivalent to the queueing network model of the BSCS shown in Figure 6-1 and has the same initial conditions.

2) Consider now the IEKCS shown in Figure 3-4**,** with $K_i = S_i$, i=1,...,R. In the initial state of the system, since K_i-S_i = 0, there are no available kanbans a_i in queues A_i and all kanbans a_i are attached to parts in PA_i. As in the case of the IKCS, a kanban a_i becomes available in A_i when a finished part p_i is transferred to B_i (i=1,...,R-1), and a kanban a_R becomes available only when a finished part p_R is delivered to the customer. Also, since K_i-S_i = 0, Eq. 13 implies that $M(D_i) \geq M(A_i)$ (i=1,...,R-1), and Eq. 17 implies $M(D_{R,i}) \geq M(A_{R,i})$ (i=1,...,R-1). Therefore queues D_i and $D_{R,i}$, i=1,...,R-1, play no role in the synchronization station they belong to; hence they can be eliminated. Once queues D_i and $D_{R,i}$ are eliminated from the network in Figure 3-4, the resulting network is the same as the queueing network model of the SKCS in Figure 6-3, where queues A_i, i=1,...,R, in Figure 3-4 play the same role as queues DA_i in Figure 6-3.

■

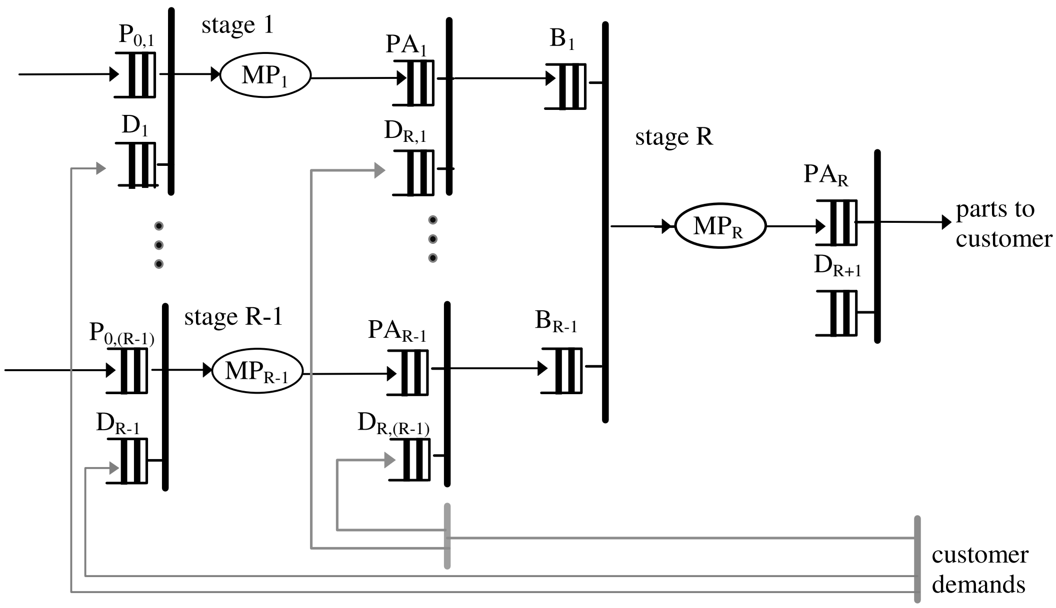

Figure 4-1: Queueing Network Model for the IEKCS with $K_i = \infty$, i =1,...,R

4.6. 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 are 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 10 and Property 12 are important since they state that the production capacity depends only on one parameter, namely K_i. The role of parameters S_i (base stock level) and K_i (number of kanbans) is thus clearly separated. The parameters S_i are related to the satisfaction of demands whereas K_i are related to the production of new parts. Thus, parameters K_i should be designed first to obtain a desirable production capacity and then parameters S_i should be designed to obtain a desirable customer satisfaction level [7].

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

4.6.1. Production Capacity of the SEKCS

Figure 4-1 shows the queueing network model for the saturated SEKCS having R stages (R−1 manufacturing stages and a single assembly stage). Figure 4-1 is obtained from Figure 3-3 as follows:

- By definition of the saturated SEKCS, queues $P_{0,i}$, (i=1,..., R-1) have an infinite number of raw parts, and queues D_i, (i=1,..., 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 the only queue in the synchronization station at the entry of stage i (i=1,...,R-1). Similarly, once D_{R+1} has been removed, PA_R remains the only queue in the synchronization station at the output of stage R. Clearly if there is only one queue feeding a synchronization station, 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,...,R-1) and PA_R may therefore be eliminated.

The queueing network model that results after these eliminations is shown in Figure 4-1.

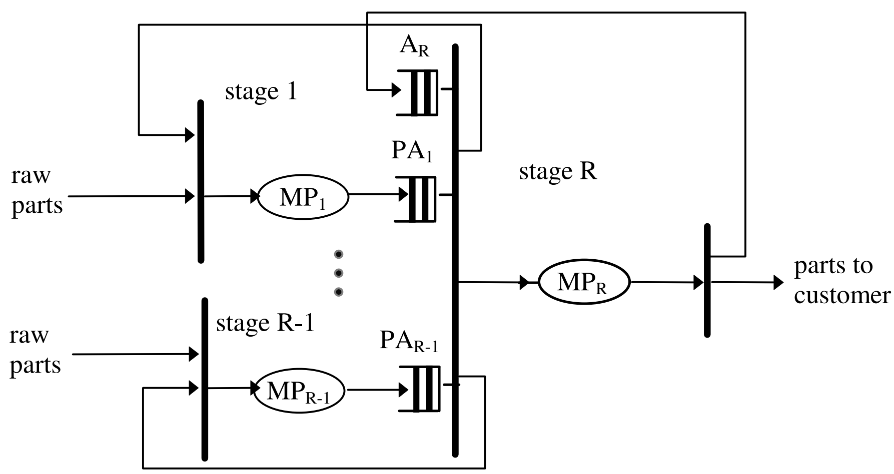

Figure 4-1:Queueing network model for the saturated SEKCS

We now have the following two properties.

Property 11: The production capacity of the SEKCS depends only on parameters K_i (i=1,...,R), and is independent of S_i (i=1,...,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 ([5], [6]). 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,...,R), is equal to the production capacity of the SKCS with the same parameters K_i (i=1,...,R), as those in the SEKCS.

The queueing network model for the saturated SKCS is obtained from Figure 6-2 by performing similar eliminations as described previously for the SEKCS. By comparing Figure 4-1 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.

4.6.2. Production Capacity of the IEKCS

Figure 4-2 shows the queueing network model for the saturated IEKCS having R stages (R-1 manufacturing stages and a single assembly stage). Figure 4-2 can be obtained from the queueing network model in Figure 3-4 after eliminating:

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

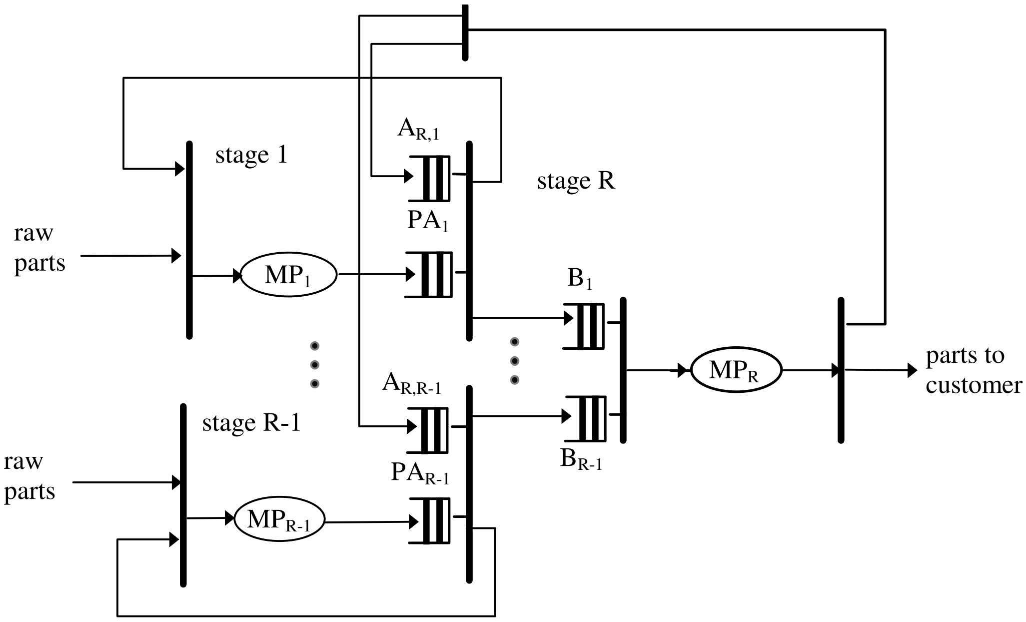

Figure 4-2: Queueing network model for the saturated IEKCS

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 ($i=1,...,R$), and is independent of S_i ($i=1,...,R$).

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

5. Comparison between the SEKCS and the IEKCS

A visual comparison between the queueing network model of the SEKCS shown in Figure 3-3 and the queueing network model of the IEKCS shown in Figure 3-4 may lead to the conjecture that the IEKCS responds faster to customer demands than does the SEKCS with the same parameters. This is due to the relative independence of the manufacturing stages upstream the assembly stage and it is stated more precisely by Property 15 that follows.

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

- $I^S_{i,n}$ and $I^I_{i,n}$, ($i=1,...,R$): the time of the n^{th} 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 n^{th} departure from the system, in the SEKCS and the IEKCS, respectively,
- $O^S_{i,n}$ and $O^I_{i,n}$, ($i=1,...,R$): the time of the n^{th} departure from MP_i, in the SEKCS and the IEKCS, respectively,

- D^S_n and D^I_n: the time of the n^{th} demand, 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 in {1,...R}. Then,

Eq. 53 $\quad I^I_{i,n} \le I^S_{i,n}, \quad i \in \{1,...,R+1\},$

Eq. 54 $\quad O^I_{i,n} \le O^S_{i,n}, \ i \in \{1,...,R\}.$

Proof: The proof is similar to the proof of Property 7. Let us assume that Eq. 53 and Eq. 54 hold up to n-1, that is:

$$I^I_{i,n-m} \le I^S_{i,n-m} \qquad i \in \{1,...,R+1\},\ m=1,2,...$$

$$O^I_{i,n-m} \le O^S_{i,n-m} \qquad i \in \{1,...,R\},\ m=1,2,...$$

- Let us first consider the case i=1,...,R-1. We have,

$$I^S_{i,n} = \max\left(D^S_n, I^S_{R,n-(K_i-S_i)}\right) \qquad \text{by Eq. 26,}$$

$$= \max\left(D^S_n, \max\left(D^S_{n-(K_i-S_i)}, \max_{j=1...R-1}\left(O^S_{j,n-(K_i-S_i)-S_j}\right), I^S_{R+1,n-(K_i-S_i)-(K_R-S_R)}\right)\right) \qquad \text{by Eq. 27,}$$

$$= \max\left(D^S_n, \max_{j=1...R-1}\left(O^S_{j,n-(K_i-S_i)-S_j}\right), O^S_{R,n-(K_i-S_i)-K_R}\right) \qquad \text{by Eq. 28,}$$

and,

$$I^I_{i,n} = \max\left(D^I_n, L^I_{i,n-(K_i-S_i)}\right) \qquad \text{by Eq. 30,}$$

$$= \max\left(D^I_n, \max\left(D^I_{n-(K_i-S_i)}, O^I_{i,n-(K_i-S_i)-S_i}, I^I_{R+1,n-(K_R-S_R)-(K_i-S_i)}\right)\right) \qquad \text{by Eq. 33,}$$

$$= \max\left(D^I_n, O^I_{i,n-K_i}, \max\left(D^I_{n-(K_R-S_R)-(K_i-S_i)}, O^I_{R,n-(K_R-S_R)-(K_i-S_i)-S_R}\right)\right) \qquad \text{by Eq. 32,}$$

$$= \max\left(D^I_n, O^I_{i,n-K_i}, O^I_{R,n-(K_i-S_i)-K_R}\right).$$

To prove Eq. 53 for i=1,...,R-1, it suffices to show that the terms inside the parenthesis in the definition of $I^S_{i,n}$ above are greater than or equal to their equivalent term in the definition of $I^I_{i,n}$. Clearly, we have $\max_{j=1...R-1}\left(O^S_{j,n-(K_i-S_i)-S_j}\right) \ge O^S_{i,n-(K_i-S_i)-S_i} = O^S_{i,n-K_i} \ge O^I_{i,n-K_i}$, where the first inequality holds by definition and the second inequality holds by the induction hypothesis. Also, $D^S_n = D^I_n$ by definition. Finally $O^S_{R,n-(K_i-S_i)-K_R} \geq O^I_{R,n-(K_i-S_i)-K_R}$ by the induction hypothesis. Therefore $I^I_{i,n} \leq I^S_{i,n}$, i=1,...,R-1.

- For i=R, we have,

$$I^S_{R,n} = \max\left(D^S_n, \max_{i=1...R-1}\left(O^S_{i,n-S_i}\right), I^S_{R+1,n-(K_R-S_R)}\right) \qquad \text{by Eq. 27,}$$

$$I^I_{R,n} = \max_{i=1...R-1}(L_{i,n}) \qquad \text{by Eq. 31,}$$

$$= \max\left(D^I_n, \max_{i=1...R-1}\left(O^I_{i,n-S_i}\right), I^I_{R+1,n-(K_R-S_R)}\right) \qquad \text{by Eq. 33.}$$

By definition, $D^S_n = D^I_n$. Also, by the induction hypothesis, $\max_{i=1...R-1}\left(O^S_{i,n-S_i}\right) \geq \max_{i=1...R-1}\left(O^I_{i,n-S_i}\right)$ and $I^S_{R+1,n-(K_R-S_R)} \geq I^I_{R+1,n-(K_R-S_R)}$. Therefore, $I^S_{R,n} \geq I^I_{R,n}$.

- For i=R+1, we have,

$$I^S_{R+1,n} = \max\left(D^S_n, O^S_{R,n-S_R}\right) \qquad \text{by Eq. 28,}$$

$$I^I_{R+1,n} = \max\left(D^I_n, O^I_{R,n-S_R}\right) \qquad \text{by Eq. 32.}$$

Clearly, $D^S_n = D^I_n$ and $O^S_{R,n-S_R} \geq O^I_{R,n-S_R}$ by the induction hypothesis. Therefore $I^S_{R+1,n} \geq I^I_{R+1,n}$.

- Finally, we have,

$$O^S_{i,n} = \sigma^S_{i,n} + \max(I^S_{i,n}, O^S_{i,n-1}) \qquad \text{by Eq. 29,}$$

$$O^I_{i,n} = \sigma^I_{i,n} + \max(I^I_{i,n}, O^I_{i,n-1}) \qquad \text{by Eq. 34.}$$

To prove Eq. 54 it suffices to show that the terms inside the parenthesis in the definition of $O^S_{i,n}$ above are greater than or equal to their equivalent terms in the definition of $O^I_{i,n}$, since $\sigma^S_{i,n} = \sigma^I_{i,n}$. Indeed, $I^S_{i,n} \geq I^I_{i,n}$ by Eq. 53 and $O^S_{i,n-1} \geq O^I_{i,n-1}$ by the induction hypothesis.

■

Property 15 states that the time at which the n^{th} part begins its processing (respectively finishes its processing) in MP_i in the IEKCS is smaller than the time when the n^{th} 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 the SEKCS does.

6. Appendix A : Base Stock and Kanban Control Systems for Assembly

6.1. Base Stock Control System for Assembly

The Base Stock Control System (BSCS) is a simple pull control mechanism for coordinating multi-stage manufacturing systems [4]. Figure 6-1 shows the queueing network model of a BSCS for an assembly systems. For the sake of simplicity, we represent the case of R-1 manufacturing stages supplying one assembly stage.

We use the same notations as in section 3. The contents and initial values of the queues, or network of queues in the case of MP_i, are described in Table 6-1.

Queue	Contents	Initial value
MP_i i=1,...,R	q_i	0
P_i i=1,...,R	p_i	S_i
D_i i=1, ...,R+1	d_i	0

Table 6-1: Contents and initial values of the queues in the BSCS

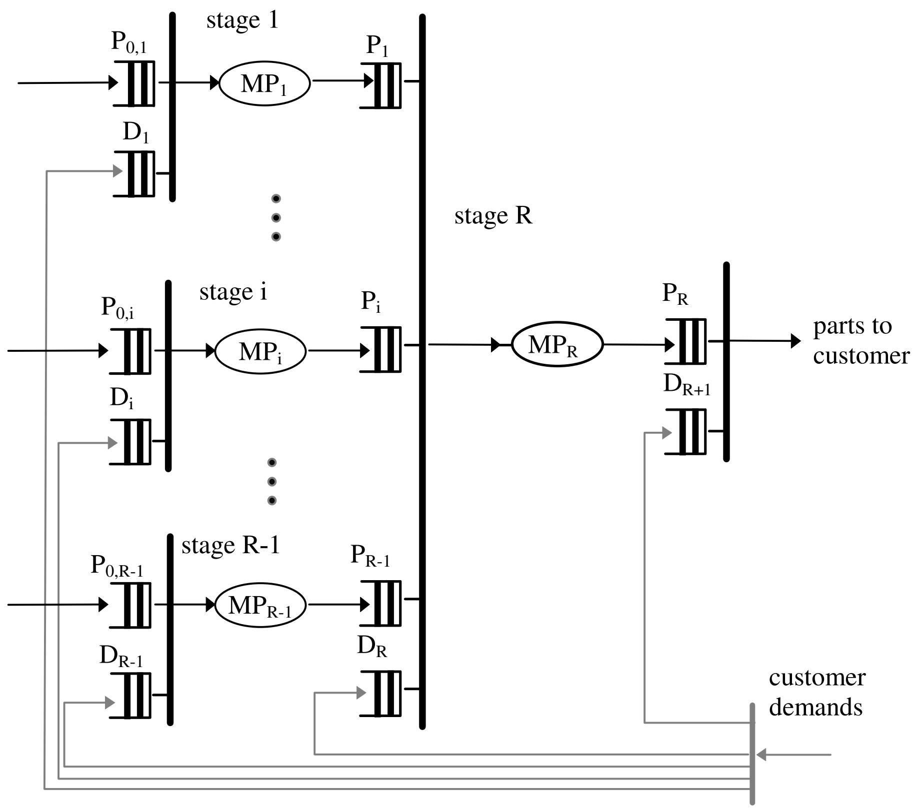

Figure 6-1: Queueing Network model for the BSCS

When a customer demand arrives to the system, it is immediately transmitted to all stages (queues D_i, i=1,...,R+1).

The behavior of the BSCS can be described as follows.

Release of parts into the manufacturing stage:
At the i^{th} manufacturing stage, queues $P_{0,i}$ and D_i are joined in a synchronization station (i=1,...R-1). This means that stage-i can begin the production of a part only when there are at least: a raw part $p_{0,i}$ in $P_{0,i}$ and a demand d_i in D_i. When these conditions are met, then,

- the raw part $p_{0,i}$ is relabelled q_i and is transferred to MP_i,
- the demand d_i is satisfied and is therefore discarded.

When the part q_i finishes its processing in MP_i, it is relabelled p_i and it joins P_i.

Release of parts into the assembly stage:
At the assembly stage R, queues P_i (i=1,...,R-1) and D_R are joined in a synchronization station. This means that the assembly operation can begin only when there are at least: a part p_i in P_i (for all i=1,...,R-1) and a demand d_R in D_R. When these conditions are satisfied, then,

- $(p_1,p_2,...,p_{R-1})$ is relabelled q_R and is transferred downstream to MP_R,
- the demand d_R is satisfied and is therefore discarded.

When a part q_R finishes its assembly process in MP_R, it is relabelled p_R and it joins P_R.

Delivery of finished parts to the customer:
Queues P_R and D_{R+1} are joined in a synchronization station. The delivery of a finished part occurs when there are at least: a part p_R in P_R and a demand d_{R+1} in D_{R+1}. Then,

- p_R is released to the customer,
- the demand d_{R+1} is satisfied and is therefore discarded.

Note that the BSCS depends only on one parameter per stage, namely S_i.

6.2. Kanban Control System for Assembly (KCS)

We briefly describe the KCS for assembly systems (see [8] for details). Two systems are defined, depending on the mechanism for the release of kanbans: the Simultaneous Kanban Control System (SKCS) and the Independent Kanban Control System (IKCS).

6.2.1. Simultaneous Kanban Control System (SKCS)

Figure 6-2 shows the queueing network model of a Simultaneous Kanban Control System (SKCS) for an assembly system. Once again, for the sake of simplicity, we represent the case of R-1 manufacturing stages supplying one assembly stage.

In addition to the notation of the section 3, we use the following bold notation:

$\mathbf{d}_i$ i=1,...,R+1 an i-element vector $(d_1,d_2,...,d_i)$; note that $\mathbf{d}_1 = (d_1)$.

Table 6-2 describes the contents and initial values of the queues, or, in the case of MP_i, network of queues:

Queue	Contents	Initial value
MP_i i=1, ...,R	(q_i,a_i)	0
PA_i i=1, ...,R	(p_i,a_i)	K_i
DA_i i=1, ...,R-1	(d_i,a_i)	0
$\mathbf{D}A_R$	$(\mathbf{d}_R,a_R)$	0
$\mathbf{D}_{R+1}$	$\mathbf{d}_{R+1}$	0

Table 6-2: Contents and initial values of the queues in the SKCS

When a customer demand arrives it joins queue $\mathbf{D}_{R+1}$ as a vector $\mathbf{d}_{R+1} = (d_1,d_2,...,d_{R+1})$.
Each manufacturing stage i has K_i kanbans a_i that authorize the production of new stage-i finished parts. Initially, all these kanbans are attached onto an equal number of parts p_i and are stored as pairs (p_i,a_i) in queue PA_i. As in the case of the extended kanban control system, queue $P_{0,i}$ (i=1,...,R−1) represents the raw parts buffers supplying the manufacturing stage i. The initial number of raw parts in $P_{0,i}$ and the arrival of new parts into $P_{0,i}$ fall outside the scope of the control mechanism and are considered as given.

The behavior of the BSCS can be described as follows.

Release of parts into the manufacturing stage:
Queues $P_{0,i}$ and DA_i (i=1,...,R-1) are joined in a synchronization station. If there are a pair (d_i,a_i) in DA_i and a part $p_{0,i}$ in $P_{0,i}$ then,
- the kanban a_i is attached onto $p_{0,i}$ which is relabelled q_i and together they are transferred downstream into MP_i,
- the demand d_i is satisfied and is therefore discarded.

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

Release of parts into the assembly stage:
At the assembly stage R, queues PA_i (i=1,...,R-1) and $\mathbf{D}A_R$ are joined in a synchronization station. This means that the assembly operation can begin only when there are at least: a pair (p_i,a_i) in PA_i (for all i=1,...,R-1) and a pair $(\mathbf{d}_R,a_R)$ in $\mathbf{D}A_R$. When these conditions are satisfied, then,
- the kanban a_i is detached from p_i, i=1,...,R-1,
- $(p_1,p_2,...,p_{R-1})$ is relabelled q_R and, together with the kanban a_R they are transferred downstream to MP_R,
- the kanban a_i is attached onto the demand d_i, and together they are transferred upstream into DA_i as a pair (d_i,a_i), i=1,...,R-1,
- the demand d_R is satisfied and is therefore discarded.

When a part q_R finishes its assembly process in MP_R, it is relabelled p_R and, together with the kanban a_R that is attached onto it, they join queue PA_R.

Delivery of finished parts to the customer:
Queues PA_R and $\mathbf{D}_{R+1}$ are joined in a synchronization station. The delivery of a finished assembled part occurs when there are at least: a pair (p_R,a_R) in PA_R, a demand vector $\mathbf{d}_{R+1}$ in $\mathbf{D}_{R+1}$. Then,
- the kanban a_R is detached from p_R,
- p_R is released to the customer,
- the demand d_{R+1} is satisfied and is therefore discarded,
- the kanban a_R is attached onto the demand vector $\mathbf{d}_R$, and together they are transferred upstream into $\mathbf{D}A_R$.

Note that the SKCS depends only on one parameter per stage, namely K_i.

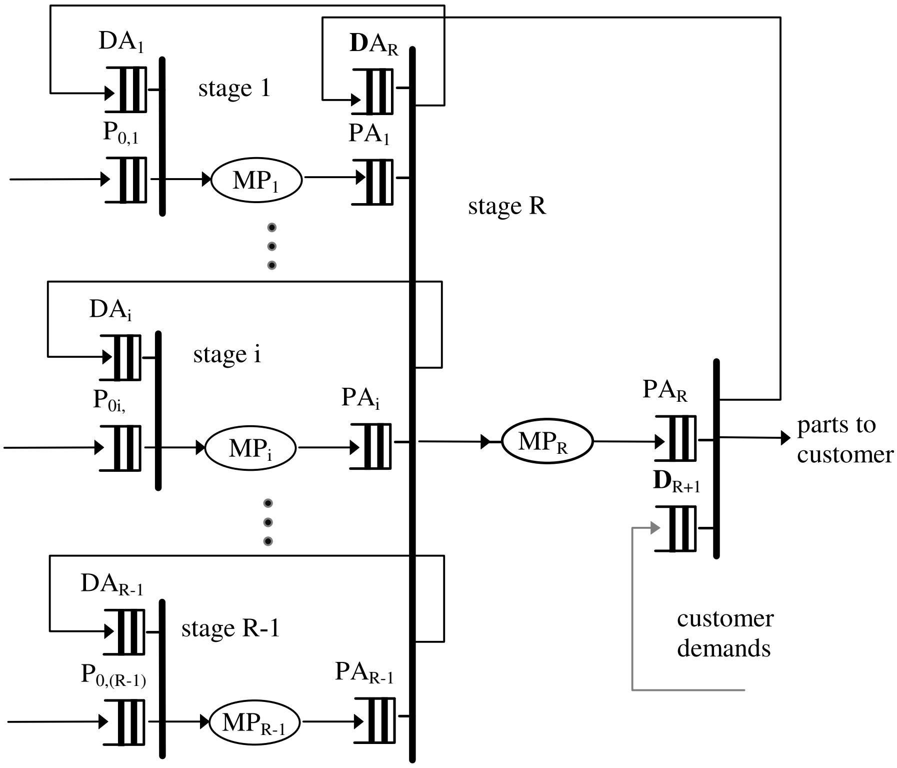

Figure 6-2: Queueing Network model for the SKCS

6.2.2. Independent Kanban Control System (IKCS)

Figure 6-3 shows the queueing network model of the Independent Kanban Control System (IKCS) for an assembly systems. We use the same notations as in section 3. Concerning the customer demands, let us introduce the following notations,

d_{R+1}		a demand for a assembled part,
d_R		a demand for an assembly operation,
d_i	$i=1,...,R-1$	a demand for the production of a new p_i,
$d_{R,i}$	$i=1,...,R-1$	a demand for the assembly of a new p_R using a stage-i finished part p_i,
$\mathbf{d}_{R+1}$		a vector demand $(d_1,d_2,...,d_{R+1})$.

Table 6-2 describes the contents and initial values of the queues, or network of queues in the case of MP_i:

Queue	Contents	Initial value
MP_i $i=1,...,R$	(q_i,a_i)	0
PA_i $i=1,...,R$	(p_i,a_i)	K_i
DA_i $i=1,...,R-1$	(d_i,a_i)	0
$\mathbf{DA}_{R,i}$	$((d_i,d_{R,i}),a_{R,i})$	0
$\mathbf{D}_{R+1}$	$\mathbf{d}_{R+1}$	0

Table 6-3: Contents and initial values of the queues in the IKCS

When a customer demand arrives it joins queue $\mathbf{D}_{R+1}$ as a vector $\mathbf{d}_{R+1} = (d_1,d_2,...,d_{R+1})$. Each demand d_R for an assembly operation is split into R-1 demands $d_{R,i}$ (i=1,...,R-1). In the same way, each kanban is split into R-1 kanbans $a_{R,i}$.

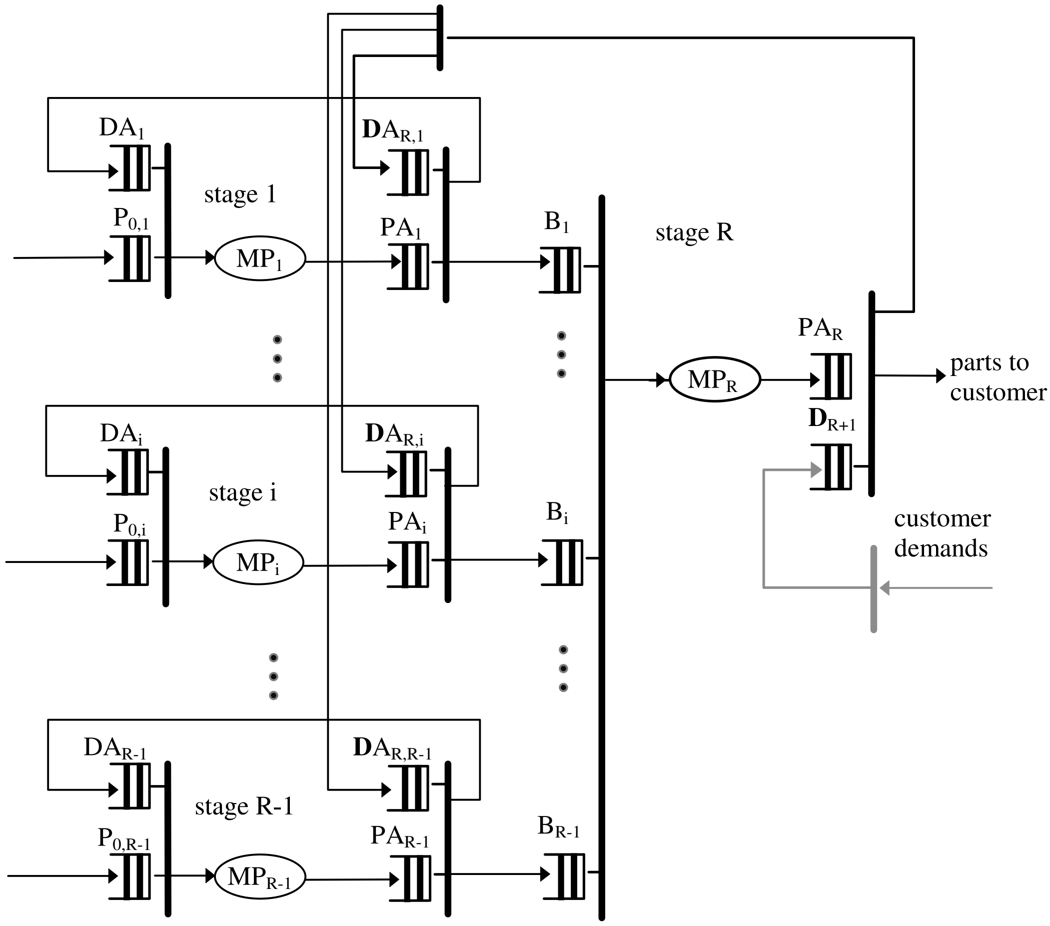

Figure 6-3: Queueing Network model for the IKCS

The behavior of the IKCS can be described as follows.

Release of parts into the manufacturing stage: It is identical to the SKCS.

Release of parts into the assembly stage:
The difference between the SKCS and the IKCS is in the way kanbans are transferred in stages 1,...,R-1. In the SKCS kanbans are transferred in stages 1,...,R-1 simultaneously, whereas in the IKCS they are transferred <u>independently</u> of each other.
The entry of the assembly stage R is composed by R synchronization stations:
– for each stage i (i=1,...,R-1), queues PA_i and $\mathbf{D}A_{R,i}$ are joined in a synchronization station,
– queues B_i are joined in a synchronization station.
The supply of stage-i finished parts for the assembly operation can occur only when there are at least: a pair (p_i,a_i) in PA_i, a pair $((di,d_{R,i}),a_{R,i})$ in $\mathbf{D}A_{R,i}$. When these conditions are satisfied, then,

- the kanban a_i is detached from p_i,
- the kanban $a_{R,i}$ is attached onto p_i and together they are transferred downstream to B_i,
- the kanban a_i is attached onto the demand d_i, and together they are transferred upstream into DA_i as a pair (d_i,a_i), i=1,...,R-1,
- the demand $d_{R,i}$ is satisfied and is therefore discarded.

When there is at least one pair $(p_i,a_{R,i})$ in each B_i the assembly process can begin, then,

– the pairs $(p_i,a_{R,i})$, are removed from queues B_i $(i=1,...,R-1)$, the $(R-1)$-tuple $(p_1,p_2,...,p_{R-1})$ is relabelled q_R, the kanbans $(a_{R,1},a_{R,2},...,a_{R,R-1})$ are merged into a kanban a_R, and the 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 relabelled 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:

Queues PA_R and $\mathbf{D}_{R+1}$ 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 part can occur as soon as there are a pair (p_R,a_R) in PA_R and a demand vector $\mathbf{d}_{R+1}$ in $\mathbf{D}_{R+1}$. When these conditions are satisfied, then,

– the kanban a_R is detached from p_R and is split into R-1 kanbans $a_{R,i}$,

– the demand d_R is split into R-1 demands $d_{R,i}$ which are transferred to queues $\mathbf{D}A_{R,i}$ together with the demand d_i and the kanban $a_{R,i}$ as a pair $((d_i,d_{R,i}),a_{R,i})$, $i=1,...,R-1$,

– p_R is released to the customer,

– the demand d_{R+1} is satisfied and is therefore discarded.

Note that the SKCS depends only on one parameter per stage, namely K_i.

7. Appendix B : Proof of Property 7

The proofs are based on expressing the times on either side of the inequality sign in Eq. 35-Eq. 43 in terms of the max-plus expressions Eq. 26-Eq. 29, and then showing that each term inside the "max" parenthesis on the left hand side of the inequality sign is smaller than or equal to the equivalent term on the right hand side of the inequality sign.

Let us recall some facts which are essential along these proofs :

– there exists a total ordering on the event times:

$$I_{i,n-m} \le I_{i,n} \text{ and } \tilde{I}_{i,n-m} \le \tilde{I}_{i,n} \qquad i=1,...,R+1, \quad n,m \in \{1,2,...\},$$

$$O_{i,n-m} \le O_{i,n} \text{ and } \tilde{O}_{i,n-m} \le \tilde{O}_{i,n} \qquad i=1,...,R, \quad n,m \in \{1,2,...\},$$

$$D_{n-m} \le D_n \text{ and } \tilde{D}_{n-m} \le \tilde{D}_n \qquad n,m \in \{1,2,...\}.$$

– initial values are fixed: $I_{i,n}$, $\tilde{I}_{i,n}$, $O_{i,n}$, $\tilde{O}_{i,n}$, D_n and $\tilde{D}_n$ are zero for $n \le 0$ (first step for the induction we use bellow),

– systems $\mathcal{S}$ and $\tilde{\mathcal{S}}$ are submitted to the same sequence of customer demands ($D_n = \tilde{D}_n$, $\forall$ n),

– sequences of processing times ($\sigma_{i,n}$ and $\tilde{\sigma}_{i,n}$) are the same in systems $\mathcal{S}$ and $\tilde{\mathcal{S}}$.

1. We assume that $\tilde{K}_q > K_q$ for some q in $\{1,...,R\}$, $\tilde{K}_i = K_i$ for all i in $\{1,...,R\}$-$\{q\}$, $\tilde{S}_i = S_i$ for all i in $\{1,...,R\}$. In addition, we assume that Eq. 35 and Eq. 36 hold up to n−1 (induction hypothesis), that is:

$$\tilde{I}_{i,n-m} \le I_{i,n-m} \qquad i=1,...,R+1, \quad m=1,2,...,$$

$$\tilde{O}_{i,n-m} \le O_{i,n-m} \qquad i=1,...,R, \quad m=1,2,....$$

- For i=1,...,R−1, we prove Eq. 35 for n, by Eq. 26, the induction hypothesis and the fact that $\tilde{K}_i - \tilde{S}_i \geq K_i - S_i$:

$$\tilde{I}_{i,n} = \max\left(\tilde{D}_n , \tilde{I}_{R,n-(\tilde{K}_i-\tilde{S}_i)}\right)$$
$$\leq \max\left(D_n , I_{R,n-(K_i-S_i)}\right) = I_{i,n} .$$

Eq. 35 is therefore proved for i=1,...,R-1.

- For i=R, by Eq. 27, the induction hypothesis, and the fact that $S_i = \tilde{S}_i$ and $\tilde{K}_R - \tilde{S}_R \geq K_R - S_R$:

$$\tilde{I}_{R,n} = \max\left(\tilde{D}_n , \max_{i=1...R-1}\left(\tilde{O}_{i,n-\tilde{S}_i}\right), \tilde{I}_{R+1,n-(\tilde{K}_R-\tilde{S}_R)}\right)$$
$$\leq \max\left(D_n , \max_{i=1...R-1}\left(O_{i,n-S_i}\right), I_{R+1,n-(K_R-S_R)}\right) = I_{R,n} .$$

Eq. 35 is therefore proved for i=R.

- For i=R+1, by Eq. 28, the induction hypothesis and the fact that $S_R = \tilde{S}_R$:

$$\tilde{I}_{R+1,n} = \max\left(\tilde{D}_n , \tilde{O}_{R,n-\tilde{S}_R}\right) \leq \max\left(D_n , O_{R,n-S_R}\right) = I_{R+1,n} .$$

Eq. 35 is therefore proved for i=R+1.

- For i=1,...,R, by Eq. 29, Eq. 35, the induction hypothesis, and the fact that $\sigma_{i,n} = \tilde{\sigma}_{i,n}$:

$$\tilde{O}_{i,n} = \tilde{\sigma}_{i,n} + \max(\tilde{I}_{i,n} , \tilde{O}_{i,n-1}) \leq \sigma_{i,n} + \max(I_{i,n} , O_{i,n-1}) = O_{i,n}.$$

Eq. 36 is therefore proved for i=1,...,R.

2. Let us now assume that $\tilde{K}_i = K_i$ for all i in {1,...,R}, $\tilde{S}_q > S_q$ for some q in {1,...,R-1}and $\tilde{S}_i = S_i$ for all i in {1,...,R}-{q}. In addition, we will assume that Eq. 37, Eq. 38, Eq. 39, and Eq. 40 hold up to n-1 (induction hypothesis).

- For i ∈ {1,...,R-1}-{q}, by Eq. 26, the induction hypothesis, and the fact that $\tilde{K}_i - \tilde{S}_i = K_i - S_i$:

$$\tilde{I}_{i,n} = \max\left(\tilde{D}_n , \tilde{I}_{R,n-(\tilde{K}_i-\tilde{S}_i)}\right)$$
$$\leq \max\left(D_n , I_{R,n-(\tilde{K}_i-\tilde{S}_i)}\right) \leq \max\left(D_i , I_{R,n-(K_i-S_i)}\right) = I_{i,n}.$$

Eq. 37 is therefore proved for i ∈ {1,...,R-1}-{q}.

- For i=R, by Eq. 27, the induction hypothesis and the hypotheses $\tilde{K}_R - \tilde{S}_R = K_R - S_R$ and $\tilde{S}_i \geq S_i$ for i=1,...,R-1:

$$\tilde{I}_{R,n} = \max\left(\tilde{D}_n , \max_{i=1...R-1}\left(\tilde{O}_{i,n-\tilde{S}_i}\right), \tilde{I}_{R+1,n-(\tilde{K}_R-\tilde{S}_R)}\right)$$

$$\leq \max\left(D_n\,,\ \max_{i=1\ldots R-1}\left(O_{i,n-S_i}\right),\ I_{R+1,n-(K_R-S_R)}\right) = I_{R,n}\ .$$

Eq. 37 is therefore proved for i=R.

- For i=R+1, by Eq. 55 and the same arguments as previously:

$$\tilde{I}_{R+1,n} = \max\left(\tilde{D}_n\,,\ \tilde{O}_{R,n-\tilde{S}_R}\right) \leq \max\left(D_n\,,\ O_{R,n-S_R}\right) = I_{R+1,n}\ .$$

Eq. 37 is therefore proved for i=R+1.

- For i=q, by the Eq. 26, Eq. 37, the assumption on the parameters $\tilde{S}_q$-S_q > 0 and the induction hypothesis:

$$\begin{aligned}\tilde{I}_{q,n} \quad &\leq \tilde{I}_{q,n+(\tilde{S}_q-S_q)} = \max\left(\tilde{D}_n\,,\ \tilde{I}_{R,n+(\tilde{S}_q-S_q)-(\tilde{K}_q-\tilde{S}_q)}\right)\\ &\leq \max\left(\tilde{D}_n\,,\ \tilde{I}_{R,n+(\tilde{S}_q-S_q)}\right)\\ &\leq \max\left(D_n\,,\ I_{R,n+(\tilde{S}_q-S_q)}\right) = I_{q,n+(\tilde{S}_q-S_q)}\ .\end{aligned}$$

Eq. 38 is therefore proved.

To prove Eq. 39 and Eq. 40, we first observe that the processing time in all MP_i ($i \neq q$) remains the same because the base stock is the same in the two systems. For MP_q for which the base stock is increased, however, we will assume that the $\tilde{S}_q$-S_q supplementary parts have been previously produced by the stage. In other words:

$$\tilde{\sigma}_{i,n} = \sigma_{i,n} \quad \text{for } i \neq q,$$

$$\tilde{\sigma}_{q,n} = \sigma_{q,n+(\tilde{S}_q-S_q)}\ .$$

- For $i \in \{1,...R\}-\{q\}$, by Eq. 29, Eq. 37 and the induction hypothesis:

$$\tilde{O}_{i,n} = \tilde{\sigma}_{i,n} + \max\left(\tilde{I}_{i,n}\,,\ \tilde{O}_{i,n-1}\right) \leq \sigma_{i,n} + \max\left(I_{i,n}\,,\ O_{i,n-1}\right) = O_{i,n}\ .$$

Eq. 39 is therefore proved for $i \in \{1,...R\}-\{q\}$.

- For i=q, by Eq. 29, Eq. 38, the assumption on the parameters $\tilde{S}_q$-S_q > 0 , $\tilde{\sigma}_{i,n} = \sigma_{i,n+(\tilde{S}_q-S_q)}$ and the induction hypothesis:

$$\begin{aligned}\tilde{O}_{q,n} \quad &= \tilde{\sigma}_{q,n} + \max\left(\tilde{I}_{q,n}\,,\ \tilde{O}_{q,n-1}\right)\\ &\leq \sigma_{q,n+(\tilde{S}_q-S_q)} + \max\left(I_{q,n+(\tilde{S}_q-S_q)}\,,\ O_{q,n-1+(\tilde{S}_q-S_q)}\right) = O_{q,n+(\tilde{S}_q-S_q)}\ .\end{aligned}$$

Eq. 40 is therefore proved.

3. Let us assume now that $\tilde{K}_i = K_i$ (i=1,...,R) and $\tilde{S}_i = S_i$ (i=1,...,R−1) and $\tilde{S}_R \geq S_R$. In addition, we assume that Eq. 41, Eq. 42 and Eq. 43 hold up to n-1 (induction hypothesis).

- For i=1,...,R-1, by Eq. 26, the hypothesis on the parameters and the induction hypothesis:

$$\begin{aligned}\tilde{I}_{i,n} &= \max\left(\tilde{D}_n\,,\,\tilde{I}_{R,n-(\tilde{K}_i-\tilde{S}_i)}\right)\\ &\leq \max\left(D_{n+(\tilde{S}_R-S_R)}\,,\,I_{R,n-(K_i-S_i)+(\tilde{S}_R-S_R)}\right) = I_{i,n+(\tilde{S}_R-S_R)}.\end{aligned}$$

  Eq. 41 is therefore proved for i=1,...,R-1.

- For i=R, by Eq. 27, $D_n = \tilde{D}_n$, the assumption on the parameters, and the induction hypothesis:

$$\begin{aligned}\tilde{I}_{R,n} &= \max\left(\tilde{D}_n\,,\,\max_{i=1\ldots R-1}\left(\tilde{O}_{i,n-\tilde{S}_i}\right),\,\tilde{I}_{R+1,n-(\tilde{K}_R-\tilde{S}_R)}\right)\\ &\leq \max\left(D_{n+(\tilde{S}_R-S_R)}\,,\,\max_{i=1\ldots R-1}\left(O_{i,n-S_i+(\tilde{S}_R-S_R)}\right),\,I_{R+1,n-(K_R-S_R)+(\tilde{S}_R-S_R)}\right)\\ &= I_{R,n+(\tilde{S}_R-S_R)}\,.\end{aligned}$$

  Eq. 41 is therefore proved for i=R.

- For i=R+1, by Eq. 28, and induction hypothesis:

$$\begin{aligned}\tilde{I}_{R+1,n} &= \max\left(\tilde{D}_n\,,\,\tilde{O}_{i,n-\tilde{S}_R}\right)\\ &\leq \max\left(D_n\,,\,O_{i,n-\tilde{S}_R+(\tilde{S}_R-S_R)}\right) = \max\left(D_n\,,\,O_{i,n-S_R}\right) = I_{R+1,n}.\end{aligned}$$

  Eq. 42 is therefore proved.

- To prove Eq. 43, we first observe that as the base stock in PA_R has been increased, we can consider that the $\tilde{S}_R$-S_R supplementary parts in the second system have been previously produced. In other words:

$$\tilde{\sigma}_{i,n} = \sigma_{i,n+(\tilde{S}_R-S_R)}\,,\text{ for i=1,...,R.}$$

  Now, by Eq. 29, Eq. 41 and the induction hypothesis,

$$\begin{aligned}\tilde{O}_{i,n} &= \tilde{\sigma}_{i,n} + \max\left(\tilde{I}_{i,n}\,,\,\tilde{O}_{i,n-1}\right)\\ &\leq \sigma_{i,n+(\tilde{S}_R-S_R)} + \max\left(I_{i,n+(\tilde{S}_R-S_R)}\,,\,O_{i,n-1+(\tilde{S}_R-S_R)}\right) = O_{i,n+(\tilde{S}_R-S_R)}.\end{aligned}$$

  Eq. 43 is therefore proved for i=1,...,R.

■

8. References

[1] F. Baccelli, Z. Liu, *Comparison Properties of Stochastic Decision Free Petri Nets*, IEEE Transactions on Automatic Control, Vol.37, No.12, Dec.1992.

[2] F. Baccelli, G. Cohen, GJ. Olsder, J-P. Quadrat, *Synchronization and Linearity: An Algebra for Discrete Event Systems,* John Wiley & Sons, 1992.

[3] J.A. Buzacott, *Queueing Models of Kanban and MRP Controlled Production Systems*, Engineering Cost and Production Economics, 1989.

[4] J.A. Buzacott and J.G. Shanthikumar, *Stochastic Models of Manufacturing Systems,* Prentice-Hall, 1993.

[5] Y. Dallery, Z. Liu, D. Towsley, *Properties of Fork/Join Queueing Networks with Blocking under Various Operating Mechanisms*, IEEE Transactions on Robotics and Automation, Vol.13, N°4, pp 503-518, 1997.

[6] Y. Dallery, Z. Liu, D. Towsley, *Equivalence, Reversibility, Symmetry and Concavity in Fork/Join Queueing Networks with Blocking,* Journal of the ACM, Vol.41, N°5, Sept.1994, pp.903-942.

[7] Y. Dallery and G. Liberopoulos, *Extended Kanban Control System: A New Kanban-type Pull Control Mechanism for Multistage Manufacturing Systems,* Technical Report, MASI, January 1996.

[8] M. Di Mascolo and Y. Dallery, *Performance Evaluation of Kanban Controlled Assembly Systems,* Symposium on Discrete Events and Manufacturing Systems of the Multiconference IMACS-IEEE/SMC CCESA'96, Lille, France, July 1996.