UNIVERSITY OF THESSALY
DEPARTMENT OF MECHANICAL & INDUSTRIAL ENGINEERING

Postgraduate (Master's) Dissertation

"COMPARATIVE MODELING OF MULTI-STAGE PRODUCTION-INVENTORY CONTROL POLICIES WITH LOT SIZING AND ADVANCE DEMAND INFORMATION"

by

TSIKIS I. ISIDOROS

A thesis submitted in partial fulfillment of the
requirements for the degree of
"Postgraduate Specialization Diploma"

2003

Acknowledgments

First of all, I am indebted to my advisor, Assistant professor George Liberopoulos, who has influenced the direction of my research and my thinking about this subject. Besides the valuable support and guidance that he freely offered me i would like to thank him for his great patience.

I thank my fellow graduate students and office-mates, Naris Stergios and Stelios Koukoumialos for their help and friendship. The Department Of Mechanical & Industrial Engineering at University of Thessaly provided research support and an appropriate environment for completing this thesis.

I thank my dear friend E. Katsarou, for her support during the course of the work that led to this thesis. Most of all, i am grateful to my parents, Ioannis and Maria Tsiki without whose encouragement and help i would not have been here. I dedicate this thesis to my father and my mother, as a token of my appreciation for their love and support.

Tsikis Isidoros

COMPARATIVE MODELING OF MULTI-STAGE PRODUCTION-INVENTORY CONTROL POLICIES WITH LOT SIZING AND ADVANCE DEMAND INFORMATION

TSIKIS ISIDOROS

University of Thessaly, 2003

Major professor: George Liberopoulos, Assistant Professor of University of Thessaly

Abstract

We present a unified framework, which is based on a queuing network modeling representation, for *describing, comparing and contrasting* simple and hybrid multi-stage production-inventory control policies with lot-sizing and advance demand information (ADI). The simple policies that we consider are reorder point and kanban policies. The hybrid policies are combinations of the simple policies, which can be materialized in a synchronized or an independent way, leading to synchronized and independent hybrid policies, respectively.

We then, attempt to describe their basic operations and functions, emphasizing on their advantages and disadvantages. Wherever it is possible, we develop evolution equations, so as to describe in detail the dynamics of each system. Using the above analysis we attempt a comparison of the above systems, emphasizing in equivalencies and superiorities.

We consider two cases where in the first, ADI is available and in the second isn't. The above analysis is carried out for both cases.

Contents

Chapter 1

Introduction and literature review

1.1. Introduction

Every operations manager should be familiar with the terms reorder point policies (RPPs), material requirements planning (MRP) and its successor manufacturing resources planning (MRP II), and just in time (JIT). These terms have been used to describe three widely practiced approaches for coordinating the flow of material in multi-stage production-inventory systems. The literature advocating one or the other approach is voluminous. Each approach has its merits and its drawbacks; however, which approach is overall better remains a point of controversy among practitioners and researchers. In a growing literature that brings to light this controversy it is often pointed out that 'which approach is better?' may not be the correct question to ask since most real systems include all three approaches anyway.

The main difficulty in comparing RPP, MRP, and JIT is that they have emerged at different points in time, within different scientific cultures, and under different general assumptions. Thus, RPPs were developed for make-to-stock inventory (i.e., uncapacitated) systems, and MRP was developed as a computerized stage coordination tool in a deterministic, discrete-time setting with *advance demand information* (ADI) in the form of a finite, planning horizon. Finally, the kanban system, the single technique most closely associated with JIT practices, was developed as a manual production control mechanism for production lines.

The purpose of our work is not to study the controversy of RPP vs. MRP vs. JIT, although some of the important issues that are related to this controversy are highlighted in Chapter 2 & 3. Instead, the goal our work is to propose a unified modeling framework, based on a queuing network representation, to help ***describe, compare and contrast*** classical multi-stage production-inventory control policies in a clear and precise manner and introduce new control approaches as hybrids of simpler policies. By exposing these policies we hope to provide a connection between RPPs, MRP and JIT and show that all three approaches can live under the same roof. The proposed framework is built by extending the unified modeling framework for pull control mechanisms in multi-stage serial systems developed in Liberopoulos and Dallery (2000) to include policies that deal with order lot sizing and Advance Demand Information.

We extend classical & hybrid multi-stage production-inventory control policies so as to include ADI mechanisms. To be more specific, in most models of production/inventory systems, demand is a random variable (or process), for which statistical information exists, and suppliers determine their inventory policies in order to deal with the uncertainty in the demand. We consider a different situation where the medium/long term demand is still assumed to be random but there is more than statistical information about the timing of short-term demand. The situation that we have in mind is that of downstream customers who place orders with a future due-date. Because our objective is to gain basic insights on the value of advance information, we consider a simple model of advance information. Namely, all customers order exactly τ periods in advance of their required delivery date. This restrictive assumption is justified when the downstream customer plans his/her production according to an MRP-type system.

In order to profit by the existence of such information, we must modify our control policy in a way that makes use of the advance demand information that is available. In that direction we integrate in the classic control policies, delay or else production order release mechanisms. In the simple case (where A.D.I is not available), a production order is released at each demand arrival. In other words, the release mechanism is triggered by an actual demand. In the modified systems, a production order is released L units before the due-date of an order. In this case, production orders are triggered by information signals rather than by actual demand arrivals. It is important to note that the release lead time L is a parameter of the policy which has to be optimized. Finally, it has to be stressed that the release lead time is not unrestricted as it is constrained by the demand lead time. Namely, L has to be less than or equal to τ. Policies that take into account advance demand information and hence MRP systems are included in Chapter 3.

1.2. General Background & Literature Review

A relatively recent review of the controversy of RPP vs. MRP vs. JIT is presented in Benton and Shin (1998). A key element to the controversy of RPP vs. MRP vs. JIT is a set of related issues regarding multi-stage production-inventory control. We will give highlights of some of these issues next.

An issue that surfaces over and over again is the issue of push vs. pull control. The most common definition of push vs. pull regards the timing of production initiation in response to demand. According to this definition, a pull system initiates production in reaction to current demand, i.e., after the realization of demand, hence the word 'pull'. A push system, on the other hand, initiates production in anticipation to future demand, i.e., before the realization of demand, hence the word 'push' (Karmarkar, 1989). In the latter case, it is assumed that advance demand information is available, i.e., that future demand is either announced in advance in the form of actual orders or commitments, or is guessed in advance in the form of a forecast, or is a combination of the two. In terms of this definition, MRP is a push system, whereas RPPs and JIT are pull systems.

According to the above definition of push vs. pull control, in both push and pull systems production responds to demand; in push systems it responds proactively, whereas in pull systems it responds reactively. One of the difficulties with this definition is that it leaves out systems where production 'ignores' demand, for example in cases where mean demand exceeds mean capacity, at least for some time. One way to circumvent this difficulty is to introduce a term to characterize control systems where production is initiated in response to demand (present or future), for example 'closed-loop' or 'feedback', and a term to characterize systems where production is initiated without regard to demand, for example 'open-loop'. Another way is to redefine push vs. pull to explicitly mean open-loop vs. closed-loop. This latter view seems to be adopted by a number of researchers. For example, in their description of JIT, Bedworth and Bailey (1987) state that JIT uses the demand pull concept, requesting inventory replenishment from main stores when work-in-process inventories reach their minimum levels, rather than pushing material through the manufacturing process and ignoring the rate at which the materials are consumed. Also, Vollman et al. (1997) find that the distinction between push and pull that is useful pertains to whether individual work centers are allowed to utilize capacity without being driven by a specific end item schedule. If we adopt the definition that push is open-loop control and pull is closed-loop control, then MRP is a pull-through approach where the master production schedule is converted into material requirements and the requirements are time phased to provide JIT production (Rice and Yoshikawa, 1982).

Closely related to the issue of open-loop vs. closed-loop is yet another definition of push vs. pull, which regards the performance measure that is being controlled. According to this definition, a push system controls throughput and measures WIP, whereas a pull system controls WIP and measures throughput (Elsayed and Boucher, 1994 and Spearman and Zazanis, 1992). In terms of this definition, an open queuing network of workstations with infinite queuing capacity is a push system, whereas a closed queuing network is a pull system. This means that MRP and RPPs are push systems since both can be modeled as open queuing networks, whereas JIT systems, such as kanban and CONWIP (Spearman et al., 1990), are pull systems since they can be modeled as closed queuing networks.

Thus far, one thing is clear: That the definition of push vs. pull is unclear. For a relatively recent and comprehensive review on the push vs. pull issue see Pyke and Cohen (1990). Two other important issues related to the controversy of RPP vs. MRP vs. JIT are the issues of local information vs. global-information control and centralized vs.decentralized control.

Local-information vs. global-information control is sometimes also referred to as decentralized-information vs. centralized-information control (Chen et al., 2000 and Simchi-Levi et al., 2000). Local information implies that each stage sees demand only in the form of orders that arrive from the stages it directly supplies and has visibility of only its own inventory status, costs, and so on. Global information implies that each stage has visibility of the demand and inventory status of all the downstream stages in the system (Silver et al., 1998). According to this definition, installation stock RPP, MRP, and kanban are local-information systems, whereas echelon stock RPP is a global-information system. One of the major advantages of global information over local information is that using global information can help significantly reduce the so-called 'bullwhip effect' (Lee et al., 1997 and Chen et al., 2000), which can result in important inventory cost savings.

Centralized vs. decentralized control refers to the number of decision makers. Centralized control implies that all relevant information in the system flows to a central point where all decisions are made in an attempt to globally optimize the entire system. These decisions are then communicated to all stages to be implemented. Clearly, this is the case if the entire system is owned by a single organization, but it can also be true in a system that includes many owning organizations. In this case the savings or profits that result from the optimization must be allocated across the organizations using some contractual mechanism (Simchi-Levi et al., 2000). Decentralized control implies that decisions are made independently by separate stages, which leads to local optimization of the system (Silver et al., 1998 and Zipkin, 2000). With the above definition in mind, the issue of centralized vs. decentralized control is also closely related to the issue of cooperation vs. competition (Cachon and Zipkin, 1999). Notice that a centralized control system is at least as effective as a decentralized control system because the former system can make all the decisions that the latter system would make. Also notice that in a local-information system, i.e., a system where each stage can access only its own information, centralized control is impossible. Centralized control is often identified with push systems because a central decision maker pushes stock to the stages that need it most, whereas decentralized control is often identified with pull systems because independent decision makers pull stock from their suppliers (Federgruen, 1993 and Pyke and Cohen, 1990). The issue of centralized vs. decentralized control is sometimes erroneously confused with the issue of local-information vs. global-information control. Namely, centralized control is often identified with global-information control and decentralized control is often identified with local-information control.

A final note regards the issue of make-to-stock vs. make-to-order control. In a make-to-stock system every stage 'blindly' produces inventory up to a certain target level ahead of time, i.e., before any demands have arrived to the system, so that when a demand arrives there is a good chance that it may be filled from inventory. The target level is determined either dynamically based on the distribution of forecasted demand and forecast error, or statically based on the stationary distribution of demand. In a make-to-order system no inventory is produced ahead of time. Instead, production is initiated whenever an order arrives to the system. RPPs are make-to-stock, unless they do not allow for positive inventory levels. JIT systems are always make-to-stock since in JIT systems replenishment orders are triggered after parts are consumed.

The issue of make-to-stock vs. make-to-order is related to the availability of advance demand information and therefore to the first definition of push vs. pull. More specifically, if advance demand information is available in the form of a demand lead time (e.g., confirmed customer orders), allowing production to be initiated before demand, there is less need to 'blindly' produce inventory ahead of time. Indeed, in such a case it may be advantageous to exploit as much of the demand lead time as possible by initiating production as early as it takes in order to reduce the inventory target level as much as possible. In other words, there is a tradeoff between the production lead time and the inventory target level: the sooner production begins the less inventory needs to be kept. With this in mind, MRP is a deterministic make-to-order system that copes with uncertainty usually by inflating lead times rather than by introducing an inventory target level in the form of safety stock. For a discussion on the issue of safety stock vs. safety time in MRP systems see Buzacott and Shanthikumar (1994), Karaesmen, Buzacott and Dallery (2002) and Karaesmen, Liberopoulos and Dallery (2002).

One of the difficulties that arise when comparing different control policies stems from the ambiguity surrounding the issues raised above. The main difficulty, however, is that most control policies have emerged at different points in time, within different scientific cultures and under different general assumptions. Thus, RPPs were developed for pure inventory (i.e., uncapacitated) systems usually with no advance demand information. MRP systems were developed as computerized stage coordination tools in a deterministic, discrete-time setting with advance demand information in the form of a finite, planning horizon, usually using forecasts. Finally, the kanban system, the single technique most closely associated with JIT practices, was developed as a manual production control mechanism at the machine level.

In the rest of this thesis we use a unified modeling framework, based on a queuing network representation, to describe, compare and contrast classical multi-stage production-inventory control policies in a clear and precise manner and introduce new control approaches as hybrids of simpler policies. In the following section we present some of the modeling assumptions that are common to all these policies.

1.3. Modeling Assumptions

We consider an N-stage serial production-inventory system. Every stage consists of a *work-in-process* (WIP) facility where parts are processed, followed by a *finished goods* (FG) output store where processed parts are stored. We assume that:

1. ***there is no advance demand information***. More specifically, we assume that customers arrive randomly in time and that each customer places an immediate request for a non-fixed number of *end items*, i.e. stage-N FG. Demands that are not satisfied from FG inventory immediately are backordered and are referred to as *backordered demands* (BD). The arrival of a customer demand for end items *eventually* triggers a replenishment order for FG inventory at

every stage. The exact time at which these orders are placed depends on the control policy in place. We assume that there is an infinite supply of raw parts feeding the first stage. FG inventory levels at all stages are followed continuously, and replenishments of FG inventory may be ordered at any time. There is a setup cost associated with placing and processing an order, therefore orders are placed and released for processing in *batches* or *lots*. Demands that are waiting for the arrival of other demands to complete a lot are referred to as *single demands* (*SD*).

Our goal is to define in a clear and precise manner production-inventory control policies that decide when to place and release replenishment orders at each stage. Since there is no advance demand information, we focus on make-to-stock policies. According to such policies, FG inventory at the last stage is "blindly" produced up to a certain target level ahead of time so that when a demand arrives there is a good chance that it may be filled from FG inventory. Upon the arrival of a demand, FG inventory is consumed and replenishment orders are eventually placed and released for processing at every upstream stage to raise the FG inventory back to the target level. To speed up the replenishment process, FG inventory at other stages may also be "blindly" produced up to a certain target level ahead of time. The target levels are either dynamic, if they are based on the distribution of forecasted demand and forecast errors, or static, if they are based on the stationary distribution of demand. In this thesis we assume that they are static. Typical policies for implementing make-to-stock control are reorder point and kanban policies.

Reorder point policies were initially developed in the context of uncapacitated inventory systems with stochastic demand, whereas kanban policies were developed in the context of capacitated production systems. In reorder point policies, the triggering of replenishment orders at every stage is based on the *inventory position* of the stage. In kanban policies, the triggering of replenishment orders at every stage is based on the *actual inventory* of the stage. The definition of inventory position in reorder point policies, and of actual inventory in kanban policies, follows the *installation* or the *echelon* concept, leading to *local* or *global* information policies, respectively. In the sections that follow, we will present classical reorder point and kanban policies as well as combinations of these policies.

2. ***the system has access to perfect ADI over a finite time horizon***. More specifically, we assume that customers arrive randomly in time and that each customer places an order for a non-fixed number of *end items*, i.e., stage-N FG, to be delivered to him exactly T time units

after the time of his arrival. The order can be neither cancelled nor modified. T is referred to as the *demand lead time*.

The arrival of every customer demand triggers the consumption of an end-item from FG inventory and the placement, activation, and release of a replenishment production order to the facility of each stage in the system. The consumption of an end-item from FG inventory is activated T time units after the arrival time of the demand. If no end-items are available at that time, the demand is backordered. The placement, activation, and release of replenishment production orders to the facilities of each stage depend on the control policy in place. To speed up the replenishment process, FG inventory at some or all the stages may have been built up to a certain target level ahead of time, i.e., before any demands have arrived to the system.

We have used the terms *placement*, *activation*, and *release* to indicate the three different phases in the life of a replenishment order. These phases are defined as follows. When an order is placed at a stage, the stage receives the order information. When an order is activated, parts corresponding to the order are *requested to be released* into the WIP facility of the stage for processing. Finally, when an order is released, parts corresponding to the order are *actually released* into the WIP facility of the stage for processing. The placement, activation, and release of a replenishment order are indicated graphically in Figure 11.

In the presence of ADI, it may be cost effective to introduce a deliberate time delay between placing and activating an order, particularly if the demand lead time T is long. An order that has been placed but has not yet been activated is referred to as an *outstanding demand* (OD). An order that has been activated may not be immediately released due to the temporary lack of parts or production authorizations, in those control policies that require production authorizations (e.g., kanban-type policies). An order that has been activated but has not yet been released is referred to as a *backordered demand* (BD).

The deliberate delay between placing and activating a replenishment order depends on the so-called installation and echelon planned lead times associated with each stage. These lead times are defined as follows. The *installation planned lead time* of stage n is denoted by l_n and is a specified fixed control parameter that is usually related to the flow time of a typical part through the facility of the stage. The *echelon planned lead time* of a stage is denoted by L_n and is the sum of the installation planned lead times of the stage and all its downstream stages, i.e.,

$$L_n = \sum_{k=n}^{N} l_k, \; n = 1, 2, \ldots, N. \qquad (1)$$

With these definitions in mind, the time of activating a replenishment order at stage n is determined using an MRP-system logic by offsetting the due date of the demand that triggered the order by the stage echelon planned lead time, L_n. This means that the order is activated with no delay, if $L_n \geq T$, or with a delay equal to $T - L_n$ with respect to the demand arrival time, if $L_n < T$. In other words, the delay in activating an order, denoted T_n, is given by

$$T_n = \max[0, T - L_n], \; n = 1, 2, \ldots, N. \qquad (2)$$

We assume that there is an infinite supply of raw parts feeding the first stage. FG inventory levels at all stages are followed continuously, and replenishments of FG inventory may be ordered at any time. There is a setup cost associated with placing and processing an order, therefore orders are placed and released for processing in *batches* or *lots*. Demands that are waiting for the arrival of other demands to form a complete lot are referred to as *single demands* (SD).

Chapter 2

Multi-Stage Production-Inventory Control Policies with Lot Sizing (no Advance Demand Information)

2.1. Introduction

In a stochastic demand environment, typical inventory control policies are RPP, (s,S) & "order up to" policies and Kanban policies. In this section we will present:

- Installation and Echelon (Q, r) Policies
- Installation and Echelon Kanban Policies
- Hybrid Installation and Kanban/Reorder Point (Q, r) Policies

Most of the results on IS and ES (Q,r) policies presented here are interpreted from Axsater and Rosling (1993) and Axsater (2000).

2.2. Installation and echelon stock (Q, r) policies

When a multi-stage production-inventory system is controlled by an IS or an ES (Q, r) policy, every stage is controlled by a (Q, r) rule based on its inventory position. This means that as soon as the inventory position of stage n falls at or below a reorder point r_n, an order is placed for the least integer number of lot sizes Q_n that raises the inventory position above r_n.

The difference between IS and ES policies lies in the definition of inventory position. In an IS policy, the inventory position at stage n is defined as the *installation stock* at stage n, i.e. stock on hand (stage-n FG) plus outstanding orders (stage-n WIP + BD) minus backorders (stage-(n+1) BD). In an ES policy, the inventory position at stage n is defined as the *echelon stock* at stage n, i.e. the sum of the installation stocks at stage n and all its downstream stages. In other words, the installation and echelon stock at stage n, which are denoted by i_n and I_n, respectively, are defined as

$$i_n = \mathrm{BD}_n + \mathrm{WIP}_n + \mathrm{FG}_n - \mathrm{BD}_{n+1},\ n = 1, 2, \ldots, N, \tag{3}$$

$$I_n = \mathrm{BD}_n + \sum_{k=n}^{N}\left(\mathrm{WIP}_k + \mathrm{FG}_k\right) - \mathrm{BD}_{N+1},\ n = 1, 2, \ldots, N, \tag{4}$$

and are related as follows:

$$I_n = \sum_{k=n}^{N} i_k,\ n = 1, 2, \ldots, N, \tag{5}$$

$$i_n = I_n - I_{n+1},\ n = 1, 2, \ldots, N\text{-}1, \text{ and } i_N = I_N. \tag{6}$$

With the above definitions in mind, the decision to place an order at each stage is based on local information, in an IS policy, and on global information, in an ES policy. The parameters Q_n and r_n are in general different for each stage. We make the common assumption that the order lot sizes satisfy

$$Q_n = j_n \cdot Q_{n+1},\ n = 1, 2, \ldots, N, \text{ and } Q_{N+1} = 1, \tag{7}$$

for some positive integers j_n. Assumption (7) is necessary if the rationing policy is to satisfy all or nothing of a production order, because then the installation stock at every stage should always consist of an integer number of downstream lot sizes (except for the last stage where the rationing policy allows the partial satisfaction of a customer order as long as stock is available). Besides simplifying material handling, the integer ratio constraint (7) also simplifies analysis significantly. The cost increase due to constraint (7) is likely to be insignificant due to the insensitivity of inventory costs to the choice of order quantities (Chen, 1998).

A queueing network model representation of a two-stage production-inventory system operating under an IS (Q, r) policy is shown in Figure 1.

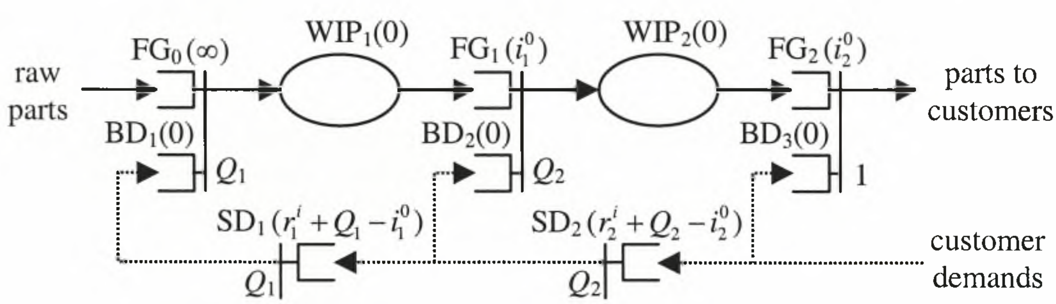

Figure 1: Queueing network model representation of a two-stage production-inventory system operating under an IS (Q, r) policy.

The symbolism used in Figure 1 and all other similar figures that follow in the rest of the thesis is as follows. The ovals represent WIP facilities, and the queues followed by vertical bars represent synchronization stations linking the queues. Queues are labeled according to their content, and their initial value is indicated inside parentheses. For example, the queue representing the FG output store of stage 1 is labeled FG_1 and its initial value is i_1^0. The marking at the bottom corner of every synchronization station indicates the lot size needed to activate the synchronization station, i.e. the minimum number of customers that must be present in each queue to activate the synchronization station. For example, queues FG_1 and BD_2 are linked in a synchronization station marked with "Q_2." This means than as soon as there are at least Q_2 parts in FG_1 and Q_2 demands in BD_2, then exactly Q_2 of the parts depart from FG_1 and enter into WIP_2, and exactly Q_2 of the backordered demands depart from BD_2 and are discarded since they are satisfied. Another example is the synchronization station consisting of a single queue, SD_1, which is marked with "Q_1." This marking means that as soon as there are at least Q_1 demands in SD_1, then exactly Q_1 of the demands depart from SD_1 and enter into queue BD_1.

The local information nature of IS policies and the global information nature of ES policies is reflected in the way customer demand information is communicated to all stages, as can be seen in Figure 1 and Figure 2, respectively. In an IS policy, customer demand information is communicated from a stage to its previous upstream stage only when an order is placed at the former stage. In an ES policy, on the other hand, a customer demand is communicated to all stages immediately upon its arrival to the system.

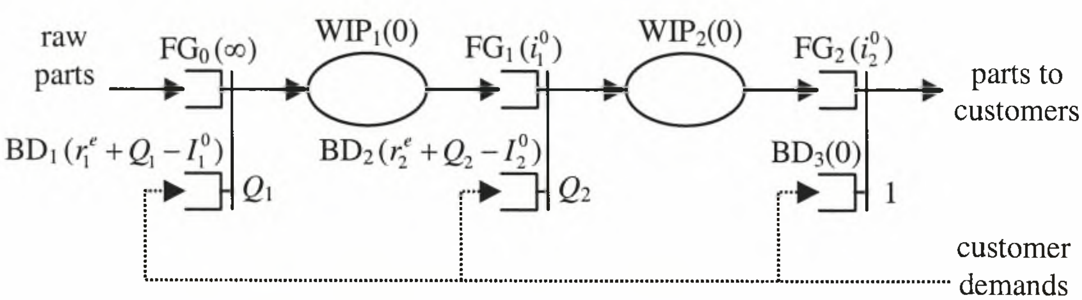

Figure 2: Simplified queueing network model representation of a two-stage production-inventory system operating under an ES (Q, r) policy.

2.2.1. Behavior and properties of Installation stock (Q, r) policies

Let us first examine IS (Q, r) policies. We assume that the initial installation stock FG inventory positions in an IS (Q, r) policy satisfy $r_n^i < i_n^0 \le r_n^i + Q_n$ for all n, where r_n^i are the installation stock reorder points. These conditions will anyway be satisfied as soon as an order has been placed by every stage. An IS (Q, r) policy is always nested in the sense that when an order is placed at stage *n*, then orders must simultaneously be placed at all downstream stages as well. This is evident by looking at Figure 1. Further more an IS (Q, r) policy does not depend on the initial installation stock positions but only on the reorder points r_n^i and the reorder quantities Q_n.

With the above observations in mind an IS (Q, r) policy (with initial installation stock positions equal to their maximum level) can always be replaced by an equivalent ES policy with initial echelon stock positions

$$I_n^0 = \sum_{k=n}^{N} i_k^0 = r_n^e + Q_n, \qquad (8)$$

where r_n^e are the echelon stock reorder points in the equivalent ES policy and are given by (see Axsater and Rosling, 1993)

$$r_n^e = r_n^i + \sum_{k=n+1}^{N} (r_k^i + Q_k), \text{ for } n = 1, 2, \ldots, N\text{-}1, \text{ and } r_N^e = r_N^i. \qquad (9)$$

2.2.2. Behavior and properties of Echelon stock (Q, r) policies

A similar analysis can be performed for ES (Q, r) policies. Here again we assume that the initial echelon stock FG inventory positions in an ES (Q, r) policy satisfy $r_n^e < I_n^0 \leq r_n^e + Q_n$, for all *n*, where r_n^e are the echelon stock reorder points. These conditions will anyway be satisfied as soon as an order has been placed at every stage. Unlike IS (Q, r) policies, ES (Q, r) policies generally depend on the initial echelon stock positions I_n^0 as well as on the echelon stock reorder points r_n^e and the reorder quantities Q_n. Also, unlike IS (Q, r) policies, ES (Q, r) policies are not always nested. If an ES (Q, r) policy is nested, however, then it can be replaced by an equivalent IS (Q, r) policy; otherwise, it can not. Axsater and Rosling (1993) show that a necessary and sufficient condition for an ES policy to be nested is that the initial installation stock inventory positions satisfy

$$i_n^0 = I_n^0 - I_{n+1}^0 = r_n^e - r_{n+1}^e + (k_n - 1) \cdot Q_{n+1}, \text{ for } n = 1, 2, \ldots, N\text{-}1, \qquad (10)$$

for some positive integers k_n such that $1 \leq k_n \leq j_n$. If condition (10) holds, the resulting nested ES policy can be replaced by an equivalent IS policy with initial installation stock positions

$$i_n^0 = I_n^0 - I_{n+1}^0 = r_n^i + k_n \cdot Q_{n+1}, \text{ for } n = 1, 2, \ldots, N\text{-}1, \text{ and } i_N^0 = I_N^0, \tag{11}$$

where r_n^i are the reorder points in the equivalent IS policy and are given by:

$$r_n^i = r_n^e - r_{n+1}^e - Q_{n+1}, \text{ for } n = 1, 2, \ldots, N\text{-}1, \text{ and } r_N^i = r_N^e. \tag{12}$$

Moreover, in this case, the resulting nested ES policy does not depend on the initial installation stock positions and therefore on the initial echelon stock positions.

2.2.3. Comparison of Installation & Echelon stock (Q, r) policies

To summarize, *IS (Q, r) policies are nested ES (Q, r) policies and are therefore special cases of ES policies.* Axsater and Rosling (1993) give an example where a non-nested ES policy is preferable to a nested ES policy. An important implication of the preceding analysis is that the behavior of an IS policy does not depend on the initial FG inventory positions. The behavior of an ES policy, on the other hand generally depends on the initial FG inventory positions, except when condition (10) holds, in which case the resulting ES policy is nested and can be replaced by an equivalent IS policy. In other words, IS policies depend on two parameters per stage, r_n and Q_n, i.e. they have two degrees of freedom on the choice of parameters per stage, whereas ES policies generally depend on three parameters per stage, r_n, Q_n, and I_n^0, i.e. they have three degrees of freedom per stage. Chen (1998) gives an alternative definition on the difference in the degrees of freedom on the choice of parameters between the two policies. More specifically, he states that without loss of generality, the installation stock reorder points in an IS policy, r_n^i, must be integer multiples of Q_{n+1}, for $n = 1, 2, \ldots, N\text{-}1$, whereas in an ES policy no such restrictions are placed on the echelon stock reorder points r_n^e, for $n = 1, 2, \ldots, N$.

2.3. Installation and echelon kanban policies

Motivated by the preceding discussion regarding IS and ES policies, we examine the notions of *installation* and *echelon kanbans*, which lead to the definitions of IK and EK policies, respectively.

When a multi-stage production-inventory system is controlled by an *installation kanban* (IK) or an *echelon kanban* (EK) *policy*, every stage n has associated with it a finite number of authorization cards or kanbans, which is equal to an integer multiple of the stage lot size Q_n. A kanban is either free or attached onto a part. A free stage-n kanban is used to communicate a customer demand for one part at stage n. More specifically, when a lot of Q_n free stage-n kanbans has been completed, an order of equal size is placed at stage n. If a lot of Q_n parts in stage-(n-1) FG inventory is available, the lot of free kanbans is attached onto the lot of parts and the combined lot is released into the WIP facility of stage n. The kanbans remain attached to the lot of parts until the combined lot reaches a certain *final* FG output store. From there, the lot of parts is being depleted as FG inventory is being consumed by the next downstream stage or by customers (if the final FG output store is the output store of the last stage). When a part is consumed, the kanban that was attached to it is detached and becomes a free kanban again. This free kanban is used once again to communicate a customer demand for one part at stage n so that when a lot of Q_n free kanbans has been formed, an order of equal size is placed at stage n.

The difference between IK and EK policies lies in the definition of the *final* FG output store, i.e. the point after which kanbans are detached from parts. In an IK policy, the final FG output store at stage n is the FG output store of stage n. In an EK policy, it is the FG output store of the last stage, i.e. stage N. This means that in an IK policy, a stage-n kanban follows a part through the WIP facility and the FG output store of stage n and is detached from the part after the part leaves the FG output store of stage n. In an EK policy, on the other hand, a stage-n kanban follows a part through the WIP facilities and FG output stores of stages n through N and is detached from the part after the part leaves the FG output store of stage N. This implies that in an IK policy, the decision to place an order at each stage is based on local information, whereas in an EK policy it is based on global information from all downstream stages. The kanbans used in IK and EK policies are referred to as *installation* and *echelon kanbans*, respectively. Note that in an IK policy, every part in the WIP facility or FG output buffer at stage n has attached onto it a stage-n installation kanban. In an EK policy, on the other hand, every part in the WIP facility or FG output buffer at stage n has attached onto it one echelon kanban from each of stages 1 through n. This means that in an EK policy, when an end item is consumed by a customer, N echelon kanbans are detached from the part and turn free.

The simplified queueing network model representations of a two-stage production-inventory system operating under an IK and an EK policy are shown in Figure 3 and Figure 4, respectively.

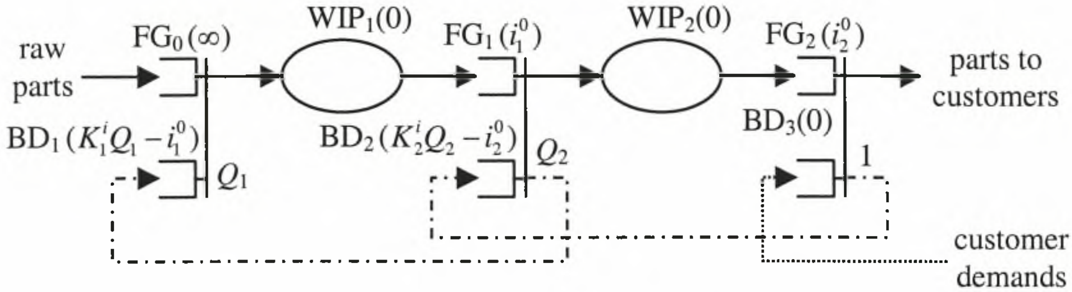

Figure 3: Simplified queueing network model representation of a two-stage production-inventory system operating under an IK policy.

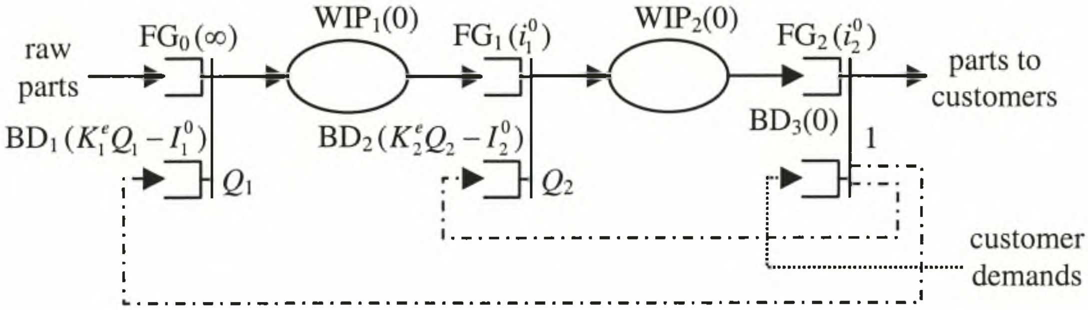

Figure 4: Simplified queueing network model representation of a two-stage production-inventory system operating under an EK policy.

2.3.1. Behavior and properties of installation kanban policies

Axsater and Rosling (1993) and Axsater (2000) compare IK policies to IS policies. More specifically, they view IK policies as being inherently IS (Q, r) policies, where a) backorders are not subtracted from the definition of the installation stock inventory position, and the reorder point at stage n is defined as $r_n^i = (K_n^i - 1) \cdot Q_n$, where K_n^i is an integer such that $K_n^i \geq 1$ and $K_n^i \cdot Q_n$ is the number of installation kanbans at stage n, or b) the inventory position is defined exactly as in IS policies, and the reorder point is occasionally decreased (when there are backorders).

It is important to note that in an IK policy the reorder point r_n^i is an integer multiple of the stage lot size Q_n, whereas in an IS (Q, r) policy it need not be. Instead, in an IS (Q, r) policy, it is natural to require only that r_n^i be an integer multiple of the downstream stage lot size Q_{n+1}, for $n = 1, 2, \ldots, N-1$, if the rationing policy is to satisfy all or nothing of a production order.

As in the case of *Installation (Q,r) policies,* IK policys does not depend on the initial installation stock positions but only on the reorder quantities Q_n and the integers K_n^i, which together with the Q_n define the reorder points r_n^i.

Axsater and Rosling (1993) conclude that IK policies are inherently IS (Q, r) policies with some limitations and are therefore inferior to IS (Q, r) policies, although in a more recent paper they recognize that this conjecture is not always correct (Axsater and Rosling, 1999). In our view there is a fundamental difference between installation stock and IK policies. In an IK policy, demand is communicated at a stage only when FG inventory is consumed by the next downstream stage or by a customer. In an IS (Q, r) policy, on the other hand, demand is communicated at a stage irrespectively of whether FG inventory is consumed or not. This difference is quite evident when one compares Figure 3 to Figure 1.

A consequence of this difference is that an IK policy is never nested in the sense that an IS (Q, r) policy is, i.e. in the sense that when an order is placed at stage *n*, then orders must simultaneously be placed at all downstream stages as well. Instead, the only thing that can be said about IK policies is that when an order is placed at stage *n*, then an order must be released in the next downstream stage or to a customer (if stage *n* is the last stage).

A more important consequence of the difference between IK and IS (Q, r) policies is that in an IK policy, the WIP + FG inventory at every stage is always bounded by the number of installation kanbans. In an IS (Q, r) policy, on the other hand, although the FG inventory at every stage is bounded by the initial FG inventory position, the WIP inventory is unbounded. The lack of an upper bound on the WIP inventory may not be a problem in uncapacitated systems such as pure inventory systems. Suggestively, Axsater and Rosling (1993) cite Veinott (1965) who demonstrated the optimality of reorder point (Q, r) policies for single-stage systems for classical cost structures and fixed lot-size *Q*. In capacitated production-inventory systems, however, (Q, r) policies may lead to significant congestion and are certainly not optimal. Suggestively, Veach and Wein (1994) and Liberopoulos and Dallery (2001) demonstrate that base stock policies are not optimal for single-stage production-inventory systems under classical cost structure assumptions. Numerical evidence in Duri *et al.* (2000), Karaesmen and Dallery (2000) and Liberopoulos and Koukoumialos (2001) also support the argument that base stock policies are not optimal in production-inventory systems with more than one stages.

2.3.2. Behavior and properties of echelon kanban policies

Similar arguments hold when one compares EK to ES policies. Namely, one can view EK policies as being inherently ES (Q, r) policies, where a) backorders are not subtracted from the definition of the echelon stock inventory position, and the reorder point at stage n is defined as $r_n^e = (K_n^e - 1) \cdot Q_n$, where K_n^e is an integer such that $K_n^e \geq 1$ and $K_n^e \cdot Q_n$ is the number of echelon kanbans at stage n, b) the inventory position is defined exactly as in ES policies, and the reorder point is occasionally decreased (when there are backorders).

Unlike IK policies, EK policies generally depend on the initial echelon stock positions I_n^0 as well as on the reorder quantities Q_n and the integers K_n^e, which together with the Q_n, define the reorder points r_n^e. An EK policy may never be nested in the sense that an ES policy may be nested, i.e. in the sense that when an order is placed at stage n, then orders must simultaneously be placed at all downstream stages as well. Nonetheless, an EK policy may be *partially* nested in the sense that when an order is placed at stage n, then orders must simultaneously be placed at *all but the last* downstream stages as well. A necessary and sufficient condition for an EK policy to be partially nested is that the initial installation stock inventory positions satisfy

$$i_n^0 = I_n^0 - I_{n+1}^0 = (K_n^e - 1) \cdot Q_n - (K_{n+1}^e - 1) \cdot Q_{n+1} + (k_n - 1) \cdot Q_{n+1}, \text{ for } n = 1, 2, \ldots, N\text{-}1, \qquad (13)$$

for some positive integers k_n such that $1 \leq k_n \leq j_n$. If condition (13) holds, the resulting EK policy does not depend on the initial installation stock position of any stage except the last stage. Unlike a nested ES (Q, r) policy, which can always be replaced by an equivalent IS (Q, r) policy, a partially nested EK policy can never be replaced by an equivalent IK policy, because, as was already mentioned above, IK policies are never nested (either partially or fully).

Here also we find that there is a fundamental difference between EK and ES (Q, r) policies. In an EK policy, demand is communicated at a stage only when an end item from FG inventory is consumed by a customer. In an ES (Q, r) policy, on the other hand, a demand is communicated to all stages immediately upon its arrival to the system, irrespectively of whether an end item from FG inventory is consumed by a customer or not. This difference becomes evident when one compares Figure 4 to Figure 2. In fact, a close look at Figure 2 and Figure 4 reveals that the model of an EK policy is identical to the model of an ES (Q, r) policy, for all but the last stage. The implications of the difference between EK and ES (Q, r) policies are similar to the implications of the corresponding difference between IK and IS (Q, r) policies that was discussed.

An EK policy with $K_{n+1}^{e} \cdot Q_{n+1} \geq K_{n}^{e} \cdot Q_{n}$, or equivalently $K_{n+1}^{e} \geq j_{n} \cdot K_{n}^{e}$ due to (7), for some stage *n*, has the property that the release or an order into the WIP facility of stage *n* + 1 will never be blocked due to the lack of an available stage-(*n*+1) kanban in queue BD_{n+1}. Therefore, queue BD_{n+1} and the entire loop traced by stage-(*n*+1) kanbans can be eliminated, and stages *n* and *n* + 1 can be merged into a single stage. In this case, the resulting system is equivalent to a production-inventory system with *N*-1 stages operating under an EK policy. A limiting case of this is when $K_{n+1}^{e} \geq j_{n} \cdot K_{n}^{e}$, for all stages *n* = 1, 2, …, *N*-1. In this case the entire system forms a single stage with $K_{1}^{e} \cdot Q_{1}$ kanbans, and the resulting policy is equivalent to a make-to-stock CONWIP policy with lot sizing (CONWIP policies were introduced in Spearman *et al.* (1990)). Therefore, without loss of generality we assume that for a system with *N* stages, $K_{n+1}^{e} < j_{n} \cdot K_{n}^{e}$, for *n* = 1, 2, …, *N*-1.

In Section 2.2 we saw that an IS (Q, r) policy is a special case of a ES (Q, r) policy. For this reason we concluded that ES (Q, r) policies are superior to IS (Q, r) policies. Such an argument can not be carried over to kanban policies because IK and EK policies are never equivalent to each other, except in the trivial case where there is a single stage. For this reason, it is not simple to determine whether EK policies are superior to IS policies or vice versa. It should be noted, however, that an advantage of EK policies over IK policies is that the former polices use global information, whereas the latter policies use only local information.

2.3.3. Disadvantages of installation and echelon kanban policies

We already mentioned that an important advantage of kanban policies over their reorder point counterpart policies is that the former policies impose an upper bound on the WIP + FG inventory. This advantage implies inventory holding cost savings. One of the disadvantages of kanban policies, however, is that they do not communicate customer demand information to all upstream stages as quickly as their corresponding reorder point policies. This is because in kanban policies customer demand information is communicated only when a lot of kanbans is detached, and kanbans are detached only when FG parts are consumed. This disadvantage has a direct impact on customer service since it implies longer customer response times, particularly if customer demand is highly variable. It also implies that the capacity of the system depends on the number of kanbans.

One way to overcome these disadvantages and increase customer service and system capacity would be to increase the number of kanbans at every stage. Unfortunately, however, this would also increase inventory costs at every stage, since the number of kanbans determines the initial FG inventory and the reorder point. Another approach would be to uncouple a) the actions of detaching a kanban and communicating demand information and b) the initial FG inventory and reorder point from the number of kanbans at every stage. This approach can be implemented by combining an IK or an EK policy with an IS or an ES (Q, r) policy to form a more sophisticated hybrid policy. Such hybrid policies are discussed next.

2.4. Hybrid installation kanban/reorder point policies

A hybrid *installation kanban/reorder point* (IK/RP) (Q, r) policy is a combination of an IK policy with an IS or an ES (Q, r) policy. In a hybrid IK/RP (Q, r) policy, installation kanbans trace a loop within each stage and are detached from the FG output store of the stage as in an IK policy. However, when an installation kanban is detached from a part in FG inventory, it does not carry with it customer demand information, as in an IK policy. Instead, demand is communicated according to the RP policy in place.

We distinguish two types of IK/RP (Q, r) policies: *synchronized* and *independent.* We differentiate between these two types because, synchronized IK/RP (Q, r) policies are related to the *production authorization card* (PAC) system or *generalized kanban control system* (GKCS) introduced by Buzacott and Shanthikumar (1993) and Zipkin (1989), whereas independent IK/RP (Q, r) policies are related to the *extended kanban control system* (EKCS) introduced by Dallery and Liberopoulos (2000).

In both synchronized and independent IK/RP (Q, r) policies, the actions of detaching a kanban and communicating demand are uncoupled. Moreover, in both cases, the initial FG inventory and the reorder point are not determined by the number of kanbans, as is the case in IK policies. Finally, in both cases, customer demand is communicated according to the RP policy in place. The difference between the two cases is that in a synchronized IK/RP (Q, r) policy, when a stage-*n* installation kanban is detached from a part in stage-*n* FG inventory, it is used to *authorize the placement* of a replenishment order for one part at stage *n*. In an independent IK/RP (Q, r) policy, on the other hand, when a stage-*n* installation kanban is detached from a part in stage-*n* FG inventory, it is used to *authorize the release* of a replenishment order for one part at stage *n*. In other words, in a synchronized IK/RP (Q, r) policy, the placement of orders is *synchronized* with the trajectory of installation kanbans, whereas in an independent IK/RP (Q, r) policy the placement of orders is *independent* of the trajectory of installation kanbans. In both synchronized and independent IK/RP (Q, r) policies, the decision to authorize the placement or release of an order at each stage is based on local information, since it depends on the availability of installation kanbans. The decision to place an order at each stage, on the other hand, is based on local information, if the reorder point policy in place is an IS (Q, r) policy, and on global information, if the reorder point policy is an ES (Q, r) policy. With the above definitions in mind, there are four hybrid IK/RP (Q, r) policies to consider: synchronized IK/IS (Q, r) policies, synchronized IK/ES (Q, r) policies, independent IK/IS (Q, r) policies and independent IK/ES (Q, r) policies.

Queueing network model representations of a two-stage production-inventory system operating under a synchronized IK/IS (Q, r) policy, an independent IK/ES (Q, r) policy, an independent IK/IS (Q, r) policy and a synchronized IK/ES (Q, r) policy are shown in Figure 5, Figure 6.

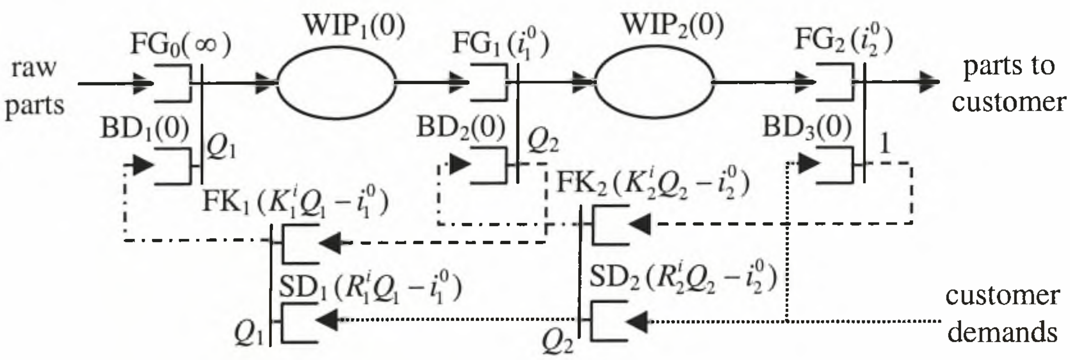

Figure 5: Queueing network model representation of a two-stage production-inventory system operating under a synchronized IK/IS (Q, r) policy.

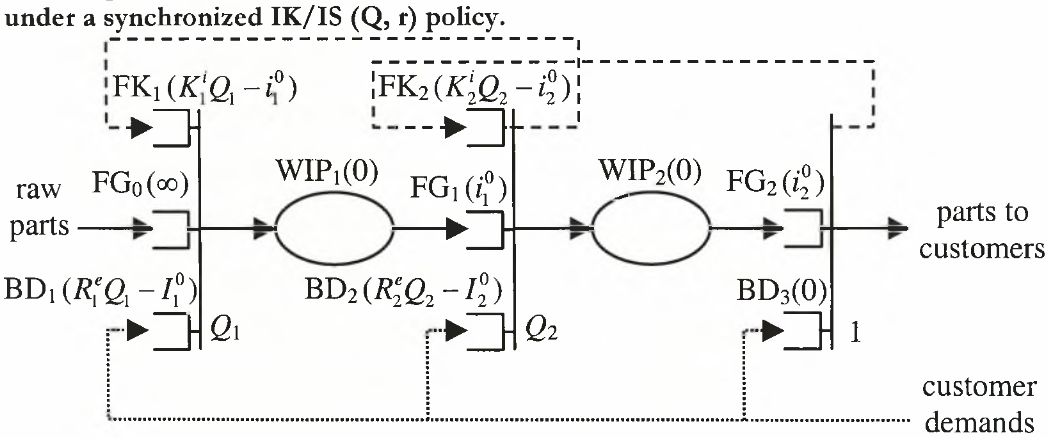

Figure 6: Queueing network model representation of a two-stage production-inventory system operating under an independent IK/ES (Q, r) policy.

A new element in all four figures (Figure 5, Figure 6), with respect to all previous figures, is the set of queues FK_n, which contain free stage-n kanbans. In all four hybrid IK/RP (Q, r) policies, the total number of installation kanbans at stage *n* is $K_n^i \cdot Q_n$, where K_n^i is an integer such that $K_n^i \geq 1$, as in IK policies. Initially, a number of these kanbans is attached onto an equal number of parts in the FG output buffer of stage *n*, defining the initial installation stock FG inventory position, i_n^0, and consequently the initial echelon stock FG inventory position, I_n^0, at stage *n*, for all *n*. The remaining installation kanbans, i.e. $K_n^i \cdot Q_n - i_n^0$ kanbans, are stored in queue FK_n as free installation kanbans ready to authorize the placement or release of an equal number of orders at stage *n*.

We omit the independent IK/IS (Q, r) policy and the synchronized IK/ES (Q, r) policy because 1) an independent IK/IS (Q, r) policy is equivalent to a nested independent IK/ES (Q, r) policy (Figure 6) and therefore a special case of the latter policy (just like an IS (Q, r) policy is equivalent to a nested ES (Q, r) policy and is therefore a special case of the latter policy), and 2) a synchronized IK/ES (Q, r) policy is identical to an independent IK/ES (Q, r) policy (Figure 6). To see this notice that the model of a synchronized IK/ES (Q, r) policy is equivalent to the model of an independent IK/ES (Q, r) policy (Figure 6), once the synchronization stations linking FK_n and SD_n are merged into the synchronization station linking queues FG_{n-1} and BD_n, $n = 1,2$.

With the above observations in mind, the only distinct hybrid IK/RP policies are synchronized IK/IS (Q, r) policies (Figure 5) and independent IK/ES (Q, r) policies (Figure 6). We will therefore focus on these policies only. Before proceeding to analyze them, however, let us make a few more observations on the hybrid IK/RP policies. The first observation is that the models of synchronized policies appear to be much more complicated than the models of independent policies, although a synchronized IK/ES (Q, r) policy is identical to an independent IK/ES (Q, r) policy, as was mentioned above. The second observation is that in a synchronized IK/IS (Q, r) policy (Figure 5), although the actions of detaching a kanban and communicating demand are not directly coupled, they are indirectly coupled. This is because the communication of demand is coupled with the placement of orders, and the placement of orders is synchronized with the trajectory of installation kanbans. This means that the communication of demands from a stage *n* to the previous upstream stage *n*-1 can be blocked due to the lack of free stage-*n* kanbans in queue FK_n. Based on these two observations, we conclude that *independent IK/RP policies appear to offer a simpler, more natural and probably more cost effective way of combining IK policies with RP policies than synchronized IK/RP policies.*

2.4.1. Behavior and properties of synchronized IK/IS (Q, r) policies

Let us first consider synchronized IK/IS (Q, r) policies. We assume that the initial installation stock FG inventory positions in a synchronized IK/IS (Q, r) policy satisfy $(R_n^i - 1)\cdot Q_n < i_n^0 \le R_n^i \cdot Q_n$, for all *n*, as was the case in IS (Q, r) policies, where R_n^i are integers such that $1 \le R_n^i \le K_n^i$. Moreover, we assume that $i_n^0 \ge Q_{n+1}$, otherwise, the system will come to a deadlock. Without loss of generality we also assume that $i_n^0 - (R_n^i - 1)\cdot Q_n = k_n \cdot Q_{n+1}$, where k_n is an integer such that $1 \le k_n \le j_n$. This assumption guarantees that the inventory of stage *n* is at the reorder point exactly when ordering. It also guarantees that $i_n^0 \ge Q_{n+1}$, so that the system will never come to a deadlock. Similarly to IS (Q, r) policies, a synchronized IK/IS (Q, r) policy does not depend on the initial installations stock positions but only on parameters Q_n, K_n^i and R_n^i.

A synchronized IK/IS (Q, r) policy is never nested in the sense that an IS (Q, r) policy is, i.e. in the sense that when an order is placed at stage *n*, then orders must simultaneously be placed at all downstream stages as well (except when $K_n^i = \infty$, for all *n*, as we will see below).

A synchronized IK/IS (Q, r) policy includes IK and IS (Q, r) policies as special cases. Namely, a synchronized IK/IS (Q, r) policy with $K_n^i = R_n^i$, for all *n* is equivalent to an IK policy. A synchronized IK/IS (Q, r) policy with $K_n^i = \infty$, for all *n* is equivalent to an IS (Q, r) policy with installation stock reorder points equal to $r_n^i = (R_n^i - 1)\cdot Q_n$, and is therefore nested. Any other synchronized IK/IS (Q, r) policy with $R_n^i \le K_n^i < \infty$ is never nested.

In order to determine the impact of the choice of system parameters on the departure times of parts from various points in the system, we will describe in detail the dynamics of kanban and material flow in the synchronized IK/IS (*Q*, *r*) policy. The dynamics of the synchronized IK/IS (*Q*, *r*) policy, can be described with recursive evolution equations that utilize operators " + " and " max " only. These equations relate the timing of a particular event in the IK/IS (*Q*, *r*) policy to the timings of events that must precede it. To elaborate, let: $D^G_{(i-1,i),n}$, $i = 1,\ldots,N+1,$: be the departure time of the *n*th part from the SS_{i-1} synchronization station to MP_i, and simultaneously the departure time of the of the corresponding kanban transfer to FK_{i-1}.

$D^{G}_{i,n}$, $i=1,...,N+1$,: be the departure time of the *n*th part and kanban from the MP_{i-1} to FG_i, $i=1,...,N$

$D^{G^*}_{(i-1,i),n}$, $i=1,...,N+1$,: be the departure time of the *n*th kanban of stage i from the SS^*_i synchronization station to BD_i, and the departure time of the corresponding information to upstream stages from the SS^*_i synchronization station to SD_{i-1}.

$D_{d,n}$: be the arrival time of the *n*th customer demand to the system. These times are shown in Figure 7:

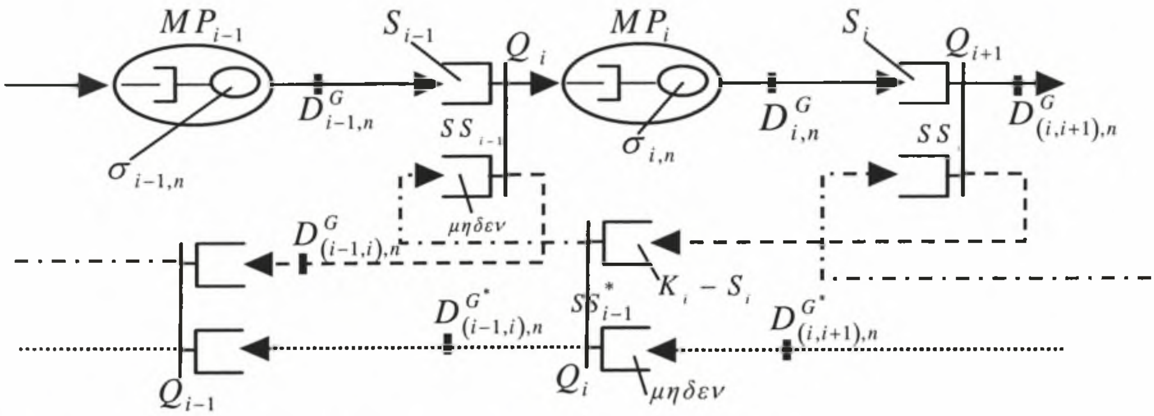

Figure 7: Timing of events in a synchronized IK/IS (Q, r) policy

Proposition 1. *In a synchronized IK/IS (Q, r) policy in which MP_i consists of a single machine, the timings of events are related by the following evolution equations*

$$D^{G}_{i,n}=\sigma_{i,n}+ max\left(D^{G}_{i,n-1},D^{G}_{(i-1,i),n}\right),i=1,...,N \tag{14}$$

$$D^{G}_{(i-1,i),n}= max\left(D^{G}_{i-1,\left\lceil \frac{n}{Q_i}\right\rceil Q_i-S_{i-1}},D^{G^*}_{(i-1,i),n}\right),i=1,...,N+1 \tag{15}$$

$$D^{G^*}_{(i-1,i),n}= max\left(D^{G}_{(i,i+1),\left\lceil \frac{n}{Q_i}\right\rceil Q_i-(K_i-S_i)},D^{G^*}_{(i,i+1),\left\lceil \frac{n}{Q_i}\right\rceil Q_i}\right),i=2,...,N+1 \tag{16}$$

where, *by convention, the maximum over an empty set is $-\infty$ and* $D^{G^*}_{(N+1,N+2),n}=D_{d,n}$

Proof. *Equation (14)* can be explained in a similar way as Equation (17) in the EKCS in Dallery's and Liberopoulos (2000). *Equations (15)* and *(16)* can be explained in a similar way as equation (18) in the EKCS in Dallery's and Liberopoulos(2000).

The symbolism $\left\lceil \frac{n}{Q_i} \right\rceil Q_i$ has to do with the batching of parts. *Equation (3)* is recursive in that it expresses $D^{G^*}_{(i-1,i),n}$ in terms of $D^{G^*}_{(i,i+1),\left\lceil \frac{n}{Q_i} \right\rceil Q_i}$. Expanding this recursion, starting from *i*=*N*+*1*, yields:

$$D^{G^*}_{(i-1,i),n} = max\left(D_{d,\left\lceil \frac{n}{Q_i} \right\rceil Q_i}, \max_{j=i}^{N} D^{G}_{j,\left\lceil \frac{n}{Q_i} \right\rceil Q_i - K_j} \right), i = 2,...,N+1 \tag{17}$$

Substituting $D^{G^*}_{(i-1,i),n}$ from *Equation (17)* into *Equation (15),* yields

$$D^{G}_{(i-1,i),n} = max\left\{ D_{d,\left\lceil \frac{n}{Q_i} \right\rceil Q_i}, D^{G}_{i-1,\left\lceil \frac{n}{Q_i} \right\rceil Q_i - S_{i-1}}, \max_{j=i}^{N}\left(D^{G}_{j,\left\lceil \frac{n}{Q_i} \right\rceil Q_i - K_j} \right) \right\}, i = 2,...,N+1, \tag{18}$$

so we have the following final evolution equations for a synchronized IK/IS (Q, r) policy:

$$\begin{cases} D^{G}_{i,n} = \sigma_{i,n} + max\left(D^{G}_{i,n-1}, D^{G}_{(i-1,i),n} \right), i = 1,...,N \\ D^{G}_{(i-1,i),n} = max\left\{ D_{d,\left\lceil \frac{n}{Q_i} \right\rceil Q_i}, D^{G}_{i-1,\left\lceil \frac{n}{Q_i} \right\rceil Q_i - S_{i-1}}, \max_{j=i}^{N}\left(D^{G}_{j,\left\lceil \frac{n}{Q_i} \right\rceil Q_i - K_j} \right) \right\}, i = 2,...,N+1 \end{cases} \tag{19}$$

A very interesting observation is that the *evolution equations (19)* of a synchronized IK/IS (Q, r) policy has the same structure with the evolution equations of an Generalized Kanban Control System (GKCS), with batch size equal to one, that are presented by Dallery and Liberopoulos (2000). The only one difference is the term *n* that appears as *n* in the evolution equations of the GKCS, while it appears as $\left\lceil \frac{n}{Q_i} \right\rceil Q_i$ in the synchronized IK/IS (Q, r). The explanation is that in the synchronized IK/IS (Q, r) policy parts must be grouped before proceeding, while in the GKCS, they don't have to.

Consequently all the properties that stand for the GKCS, stands for the synchronized IK/IS (Q, r) policy as well, with the only difference that the term *n* changes to $\left\lceil \frac{n}{Q_i} \right\rceil Q_i$.

2.4.2. Behavior and properties of independent IK/ES (Q, r) policies

A similar analysis can be carried out on independent IK/ES (Q, r) policies. We assume that in an independent IK/ES (Q, r) policy the initial echelon stock FG inventory positions satisfy $(R_n^e - 1)\cdot Q_n < I_n^0 \le R_n^e \cdot Q_n$, for all *n*, as is the case in an ES (Q, r) policy, where R_n^e are integers such that $R_n^e \cdot Q_n - R_{n+1}^e \cdot Q_{n+1} \le K_n^i \cdot Q_n$, or $R_n^e \cdot j_n - R_{n+1}^e \le K_n^i \cdot j_n$ by (7), for *n* = 1, 2, ..., N-1, and $R_N^e \le K_N^i$.

Unlike synchronized IK/IS (Q, r) policies, independent IK/ES (Q, r) policies generally depend on the initial echelon stock positions I_n^e as well as on the parameters Q_n, K_n^e and R_n^e. An independent IK/ES (Q, r) policy may be nested in the sense that an ES (Q, r) policy may be nested, i.e. in the sense that when an order is placed at stage *n*, then orders must simultaneously be placed at all downstream stages as well. The condition for this to happen is exactly the same as in an ES (Q, r) policies and is given by expression (10), where $r_n^e = (R_n^e - 1)\cdot Q_n$. If this condition holds, the nested independent IK/ES (Q, r) policy can be replaced by an equivalent independent IK/IS (Q, r) policy, just as an ES (Q, r) policy can be replaced by an equivalent IS (Q, r) policy. The resulting nested independent IK/IS (Q, r) policy does not depend on the initial installation stock positions and therefore on the initial echelon stock positions.

A nested synchronized (or equivalently independent) IK/ES (Q, r) policy, on the other hand, can not be replaced by an equivalent synchronized IK/IS (Q, r) policy, because as was already mentioned above, a synchronized IK/IS (Q, r) policy is never nested.

An independent IK/ES (Q, r) policy includes IK and ES (Q, r) policies as special cases. Namely, an independent IK/ES (Q, r) policy with $K_n^i \cdot Q_n = R_n^e \cdot Q_n - R_{n+1}^e \cdot Q_{n+1}$, or $K_n^i \cdot j_n = R_n^e \cdot j_n - R_{n+1}^e$ by (7), for *n* = 1, 2, ..., N-1, and $K_N^i = R_N^e$, is equivalent to an IK policy. An independent IK/ES (Q, r) policy with $K_n^i = \infty$ at every stage *n*, is equivalent to an ES (Q, r) policy with echelon stock reorder points equal to $r_n^e = (R_n^e - 1)\cdot Q_n$.

The dynamics of the independent IK/ES (Q, r) policy, can be described, as in the case of the synchronized IK/IS (*Q*, *r*) policy, by recursive evolution equations. These equations relate the timing of a particular event in the IK/ES (*Q*, *r*) policy to the timings of events that must precede it. To elaborate, let:

$D_{(i-1,i),n}$, $i=1,...,N+1,$: be the departure time of the *n*th part from the SS_{i-1} synchronization station to MP_i, and simultaneously the departure time of the of the corresponding kanban transfer to FK_{i-1}.

$D_{i,n}$, $i=1,...,N+1,$: be the departure time of the *n*th part and kanban from the MP_{i-1} into FG_i, $i=1,...,N$

$D_{d,n}$: be the arrival time of the *n*th customer demand to the system. These times are shown in Figure 8:

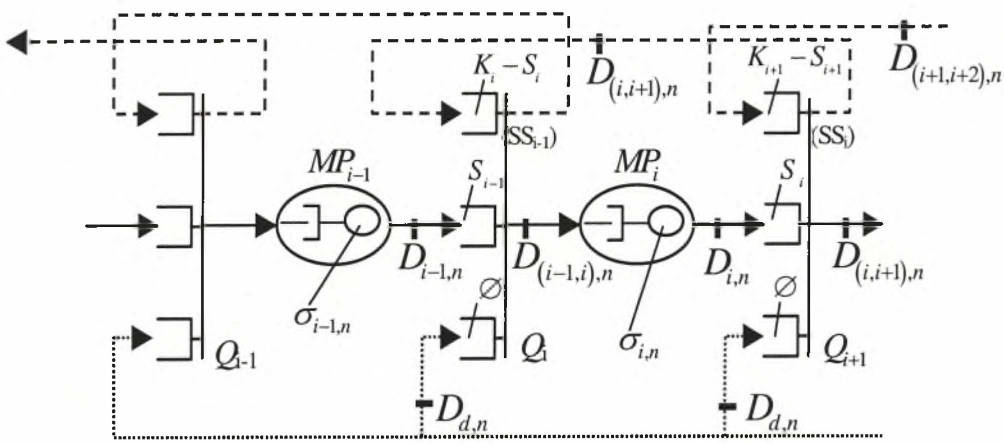

Figure 8: Timing of events in an independent IK/ES (Q, r) policy

Proposition 2. *In a independent IK/ES (Q, r) policy in which MP_i consists of a single machine, the timings of events are related by the following evolution equations*

$$D_{i,n}=\sigma_{i,n}+ max\left(D_{i,n-1}, D_{(i-1,i),n}\right), i=1,...,N \tag{20}$$

$$D_{(i-1,i),n}= max\left(D_{d,\left\lceil \frac{n}{Q_i}\right\rceil Q_i}, D_{(i,i+1),\left\lceil \frac{n}{Q_i}\right\rceil Q_i-(K_i-S_i)}, D_{i-1,\left\lceil \frac{n}{Q_i}\right\rceil Q_i-S_{i-1}}\right), i=2,...,N+1 \tag{21}$$

Proof. *Equation (20)* can be explained in a similar way as Equation (17) in the EKCS in Dallery's and Liberopoulos (2000)]. *Equation (21)* can be explained in a similar way as equation (18) in the EKCS in Dallery's and Liberopoulos (2000).

The symbolism $\left\lceil \frac{n}{Q_i}\right\rceil Q_i$ has to do with the batching of parts. *Equation (21)* is recursive in that it expresses $D_{(i-1,i),*}$ in terms of $D_{(i,i+1),*}$. Expanding this recursion, starting from $i=N+1$, yields:

$$D_{(i-1,i),n} = max\left\{D_{d,\left\lceil \frac{n}{Q_i}\right\rceil Q_i}, D_{i-1,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - S_{i-1}}, \max_{j=i}^{N}\left(D_{j,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - S_j - \sum_{m=i}^{j}(K_m - S_m)}\right)\right\}, i = 2,\ldots,N+1, \quad (22)$$

so we have the following final evolution equations for an independent IK/ES (Q, r) policy:

$$\left\{\begin{array}{l} D_{i,n} = \sigma_{i,n} + max\left(D_{i,n-1}, D_{(i-1,i),n}\right), i = 1,\ldots,N \\ D_{(i-1,i),n} = max\left\{D_{d,\left\lceil \frac{n}{Q_i}\right\rceil Q_i}, D_{i-1,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - S_{i-1}}, \max_{j=i}^{N}\left(D_{j,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - S_j - \sum_{m=i}^{j}(K_m - S_m)}\right)\right\}, i = 2,\ldots,N+1 \end{array}\right\} \quad (23)$$

As in the case of the synchronized IK/IS (Q, r) policy, the independent IK/ES (Q, r) policy has the same structure with the evolution equations of an Extended Kanban Control System (EKCS), with batch size equal to one, that are presented by Dallery and Liberopoulos (2000). The only one difference is the term n that appears as n in the evolution equations of the EKCS, while it appears as $\left\lceil \frac{n}{Q_i}\right\rceil Q_i$ in the independent IK/ES (Q, r) policy. The explanation is that in the independent IK/ES (Q, r) policy parts must be grouped before proceeding, while in the EKCS, they don't have to.

The above evolution equations imply property 1:

Property 1: *Consider the independent IK/ES (Q, r) policy, with parameters* $S_i^{'}, K_i^{'}$ *in place of* S_i, K_i, *i=1,2…N and let* $D_{i,n}^{'}, i = 1,2\ldots N$, and $D_{(i-1,i),n}^{'}, i = 1,2\ldots N+1, n = 1,2\ldots$, *denote the corresponding event times. Then the following hold.*

(i)If

$K_l^{'} > K_l$, for some $l \in \{1,\ldots,N\}$, $K_i^{'} = K_i$, for all i $\in \{1,\ldots,N\} - \{l\}$, $S_i^{'} = S_i$, for all i $\in \{1,\ldots,N\} - \{l\}$,

then

$D_{(i-1,i),n}^{'} \le D_{(i-1,i),n}, \quad i = 1,\ldots,N+1, n = 1,2,\ldots$

$D_{i,n}^{'} \le D_{i,n}, \quad i = 1,\ldots,N, n = 1,2,\ldots$

(ii)If

$S_l^{'} > S_l$, for some $l \in \{1,\ldots,N\}$, $S_i^{'} = S_i$, for all i $\in \{1,\ldots,N\} - \{l\}$, $K_i^{'} = K_i$, for all i $\in \{1,\ldots,N\} - \{l\}$, *then*

$D^{'}_{(i-1,i),n} \leq D_{(i-1,i),n}, \quad i=1,...,N+1, n=1,2,...$

$D^{'}_{i,n} \leq D_{i,n}, \quad i=1,...,N, n=1,2,...$

$D^{'}_{(i-1,i),n} \leq D_{(i-1,i),n+(S^{'}_l-S_l)}, i=1,...,l, n=1,2...$

$D^{'}_{i,n} \leq D_{i,n+(S^{'}_l-S_l)}, i=1,...,l, n=1,2...$

Part (i) of property 1 states that if the number of kanbans in stage l is increased from K_l to $K^{'}_l$, the departure time of the nth part from all the synchronization stations and manufacturing processes in the independent IK/ES (Q, r) policy will not increase but may decrease. Part (ii) of property 1 states that if the base stock in stage l is increased from S_l to $S^{'}_l$, the departure time of the nth part from those synchronization stations and manufacturing processes that are downstream of stage l will not increase but may decrease. Moreover, the departure time of the nth part from those synchronization stations and manufacturing processes that are upstream of stage l (including stage l) will not increase but may decrease with respect to the departure time of the $n+(S^{'}_l-S_l)th$ part from the same synchronization stations and manufacturing processes in the original system. To see why the latter is true, note that an increase in the base stock of stage l from S_l to $S^{'}_l$,has the same effect as that of having $(S^{'}_l-S_l)$ extra parts enter the system and receive processing all the way up to and including stage l, before the first demand arrives to the system. Therefore, the departure of the nth part in the independent IK/ES (Q, r) policy with $S^{'}_l$ from any point upstream of $J_{l-1,l}$ corresponds to the departure of the $n+(S^{'}_l-S_l)th$ p part in the independent IK/ES (Q, r) policy with S_l from the same point.

The evolution equations in Proposition 2 also clearly imply the following property:

Property 2:*The departure times* $D_{(i-1,i),n}, i=1,...,N+1, n=1,2,...$ *and* $D_{i,n}, i=1,...,N, n=1,2,...$ *are non-decreasing in* $\sigma_{l,m}, D_{l,m}$ and $D_{0,m}$ for any $m \geq 1$ and $l \in \{1,...N\}$.

The above property states that if a processing times $\sigma_{l,m}$, or a demand vector arrival time $D_{d,m}$,or a raw part arrival time $D_{0,m}$, is increased, the departure time of the nth part from all the synchronization stations and manufacturing processes in the independent IK/ES (Q, r) policy will not decrease but may increase.

2.4.3. Comparison between synchronized IK/IS and independent IK/ES (Q, r) policies

Based on our discussions above, there are two limiting cases where a synchronized IK/IS (Q, r) policy is equivalent to an IK and to an IS (Q, r) policy, respectively. There are also two other limiting cases where an independent IK/ES (Q, r) policy is equivalent to an IK and to an ES (Q, r) policy, respectively. In fact, these are the only cases where a synchronized IK/IS (Q, r) policy and an independent IK/ES (Q, r) policy are equivalent to each other. In any other case, a synchronized IK/IS (Q, r) policy and an independent IK/ES (Q, r) policy are never equivalent to each other. This means that if we take an IS (Q, r) policy and an equivalent nested ES (Q, r) policy, superimpose on each policy the same IK policy and synchronize the trajectory of installation kanbans with the placement of orders, the resulting synchronized IK/IS (Q, r) policy and synchronized IK/ES (Q, r) policy (which is equivalent to an independent IK/ES (Q, r) policy) will not be equivalent to each other. For this reason, we can not say with certainty whether an independent IK/ES (Q, r) policy is superior to a synchronized IK/IS (Q, r) policy or vice versa. It would not be surprising, however, if in many cases an independent IK/ES (Q, r) policy turned out to perform better than a synchronized IK/IS (Q, r) policy because the former policy uses global information, whereas the latter policy uses only local information. Moreover, as we observed above, synchronized IK/IS (Q, r) policies have the drawback that they appear to be more complicated than independent IKES (Q, r) policies and most importantly that they cause an indirect coupling between the actions of detaching a kanban and communicating demand. This coupling may cause delays in communicating demand information.

The notion of a synchronized IK/IS (Q, r) policy is not new. Buzacott (1989) introduced a system for coordinating multi-stage production-inventory systems called *production authorization card* (PAC) system or *generalized kanban control system* (GKCS) (see also Buzacott and Shanthikumar, 1993). A similar system was independently developed by Zipkin (1989) (see also Zipkin, 2000). The PAC system depends on three parameters per stage: The initial installation stock position, the number of installation kanbans, and the order lot size. Buzacott and Shanthikumar (1993) mention a fourth parameter, which is a time delay when placing an order. In this Chapter we will assume that the delay parameter is zero, but we deal with the case of advance demand information in the next chapter.

Buzacott and Shanthikumar (1993) demonstrate how through the appropriate choice of parameters the PAC system can be specialized into a wide variety of classical coordination approaches, such as kanban, base stock, etc. Buzacott (1989) divided the PAC system into two cases. In the first case, the number of installation kanbans at each stage is greater than or equal to the initial installation stock position. In the second case, the number of installation kanbans at each stage is smaller than the initial installation stock position. He referred to the first system as *backorderd kanban* system and to the second case as *reserve stock kanban* system. Liberopoulos and Dallery (2000) argued that the backordered kanban system is indeed a new stage coordination policy, whereas the reserve stock kanban system is a classical IK policy, i.e. a policy that limits the WIP + FG inventory at every stage, with an additional constraint on WIP inventory alone. For this reason, they identified the PAC system with the backordered kanban system only. We will follow the same approach here so that henceforth when we refer to the PAC system we will mean only the backordered kanban system. With this in mind, a PAC system (i.e. a backordered kanban system) is equivalent to a synchronized IK/IS (Q, r) policy, where the queues termed "store," "requisition tags," "process tags" and "order tags" in Buzacott and Shanthikumar (1993) are related to queues FG_n, BD_n, FK_n, and SD_n in Figure 5.

The notion of an independent IK/EK (Q, r) policy is not new, either. The idea of combining a local information kanban system and a global information reorder point policy was introduced by Dallery and Liberopoulos (2000). They defined a control system that combines a base stock policy and a kanban policy in the case of unit customer demand and unit lot sizes and called it *extended kanban control system* (EKCS). An EKCS is a special case of a independent IK/ES (Q, r) policy with unit customer demand and unit lot sizes.

A very close visual comparison between the queueing network model of the independent IK/ES (Q, r) policy shown in Figure 6 and the queueing network model of the synchronized IK/IS (Q, r) policy shown in Figure 5 may lead to the conjecture that the independent IK/ES (Q, r) policy responds faster to customer demands than does the synchronized IK/IS (Q, r) policy, given that the two systems have the same parameters. This is because in the independent IK/ES (Q, r) policy a demand d_i is transferred upstream to D_i immediately upon its arrival to the system, whereas in the synchronized IK/IS (Q, r) policy a demand d_i is transferred from D_N to DA_i in $N-i+1$ steps after the transfer of a kanban a_{N-1} from A_{N-1} into DA_{N-1} ,..., a kanban a_i from A_i into DA_i.

The comparison between the two systems is of course valid only for parameter values that satisfy inequality: $K_i \geq S_i$ For parameter values that do not satisfy this inequality the independent IK/ES (Q, r) policy is not defined. It should be noted however that a synchronized IK/IS (Q, r) policy with Si > Ki, i=1,...,N, becomes a very restrictive system known as the ``Local Control" system, in which a finished part p_i is released into MP_{i+1} depending only on the availability of space in MP_{i+1} and P_{i+1}.

What is a little puzzling, when comparing the independent IK/ES (Q, r) policy and the synchronized IK/IS (Q, r) policy, is that in the two special cases, when $K_i=\infty$ and when $K_i = S_i$; i=1,...,N, both the independent IK/ES (Q, r) policy and the synchronized IK/IS (Q, r) policy are equivalent to the BSCS and the KCS, respectively, and are therefore equivalent to each other. This raises the question: what is the difference between the independent IK/ES (Q, r) policy and synchronized IK/IS (Q, r) policy. To answer this question we compare the evolution equations relating equivalent event times in the two systems.

To distinguish equivalent event times in the two systems, we denote by $D^E_{(i-1,i),n}$ the departure time of the nth pair (p_{i-1}, a_{i-1}) from the synchronization station in $J_{i-1,i}$, $i=1,...,N+1$, and by $D^E_{i,n}$ the departure time of the nth pair (q_i, a_i) from MP_i, I=1,...,N in the independent IK/ES (Q, r) policy. The evolution equations relating $D^E_{(i-1,i),n}$ and $D^E_{i,n}$ are therefore given by Proposition 1, where $D_{(i-1,i),n}$ and $D_{i,n}$ are replaced by $D^E_{(i-1,i),n}$ and $D^E_{i,n}$ respectively. Similarly, we denote by $D^G_{(i-1,i),n}$ and $D^G_{i,n}$ the respective event times in the synchronized IK/IS (Q, r) policy, i.e., $D^G_{(i-1,i),n}$, i=1,...,N+1, is the departure time of the nth part p_i from the top synchronization station in $J_{i-1,i}$ *and* $D^G_{i,n}$ is the departure time of the nth pair (q_i, a_i) from MP_i; i=1,...,N+1. In addition, in the synchronized IK/IS (Q, r) policy we de note by $D^{G'}_{(i-1,i),n}, i=2,...,N+1$, the departure time of the nth kanban a_{i-1} from the bottom synchronization station in $J_{i-1,i}$ i.e., the time of the nth arrival of a pair (d_{i-1}, a_{i-1}) in DA_{i-1} and a vector $\mathbf{d}_{i-2}$ in $\mathbf{D}_{i-2}$ (or no vector, if i=2). These times are shown in Figure 7.

Comparing the evolution equations of the independent IK/ES (Q, r) policy and the synchronized IK/IS (Q, r) policy, given be Propositions 1 and 2, respectively, leads to the following property.

Property 3. *Consider two systems, the independent IK/ES (Q, r) policy and the synchronized IK/IS (Q, r) policy, having the same parameters Ki and Si i=1,...,N, the same sequence of service times,*

$\sigma_{i,n}, i=1,...,N, n=1,2,...,$ and the same sequence of customer demand and raw part arrival times, $D_{d,n}$ and $D_{0,n}, n=1,2,...$ respectively. Then, the following holds:

$$D^E_{(i-1,i),n} \le D^G_{(i-1,i),n},\ i=1,...,N+1,\ n=1,2,...,$$
$$D^E_{i,n} \le D^G_{i,n},\ i=1,...,N,\ n=1,2,...,$$

The above property states that the departure time of the nth part p_{i-1} from the synchronization station in $J_{i-1,i}$ in the independent IK/ES (Q, r) policy is smaller than the departure time of the nth part p_{i-1} from the top synchronization station in $J_{i-1,i}$ in the synchronized IK/IS (Q, r) policy, given the same parameter values for the two systems. This means that demands are satisfied earlier in the independent IK/ES (Q, r) policy than in the synchronized IK/IS (Q, r) policy, but it does not necessarily also mean that the independent IK/ES (Q, r) policy has an overall better performance than the synchronized IK/IS (Q, r) policy, since inventory storage costs are not taken into account. In fact, the independent IK/ES (Q, r) policy is likely to incur higher inventory storage costs than does the synchronized IK/IS (Q, r) policy. This is because in the independent IK/ES (Q, r) policy the bound on the number of pairs (p_i, a_i) in PA_i is higher than the bound on the number of finished parts p_i in P_i in the synchronized IK/IS (Q, r) policy. Namely, in the independent IK/ES (Q, r) policy the number of pairs (p_i, a_i) in PA_i is bounded by K_i (except in the last stage), whereas in the synchronized IK/IS (q, r) policy, the number of finished parts p_i in P_i is bounded by S_i, where $S_i \le K_i$. The WIP in MP_i as well as the WIP + number of finished parts in stage i, however, is bounded by K_i in both systems.

From Property 1, which states that parts move faster through the independent IK/ES (Q, r) policy than through the synchronized IK/IS (q, r) policy when production is driven by demands, it can be shown that parts move faster through the independent IK/ES (Q, r) policy than through the synchronized IK/IS (q, r) policy in the presence of infinite demands too. This then leads to the following property.

Property 4. *The production capacity of the independent IK/ES (Q, r) policy with parameters Ki and Si i=1,2,…,N, is higher than the production capacity of the synchronized IK/IS (q, r) policy with the same parameters Ki and Si.*

Further comparison between the evolution equations of the two policies reveals the following property.

Property 5. *The independent IK/ES (Q, r) policy with $K_i=S_i$ or $K_i=\infty$, $i=1,2,\ldots,N-1$ is equivalent to the synchronized IK/IS (q, r) policy with the same parameters Ki and Si.*

Proof. When $K_i=S_i$ or $K_i=\infty$, $i=1,2,\ldots,N-1$, the evolution equations of the independent IK/ES (Q, r) policy and of the synchronized IK/IS (q, r) policy are the same. The two systems are, therefore, equivalent.

Property 5 states that in order for the synchronized IK/IS (q, r) policy and the independent IK/ES (Q, r) policy to be equivalent to each other it suffices that $K_i=S_i$ or $K_i=\infty$,, for all but the last stage, i.e., for $i=1,2,\ldots,N-1$. Of course, the independent IK/ES (Q, r) policy and the synchronized IK/IS (q, r) policy with $K_i=S_i$ or $K_i=\infty$, $i=1,2,\ldots,N-1$, are both equivalent to the KCS and BSCS, respectively. In the case where there is only one stage, this stage is the last stage. In this case the two systems are equivalent no matter what parameters K_1 and S_1 are. This is stated as the following property.

Property 6. *The single-stage independent IK/ES (Q, r) policy with parameters K_1 and S_1 is equivalent to the single-stage synchronized IK/IS (q, r) policy with the same parameters.*

2.4.4. Hybrid echelon kanban/reorder point (Q, r) policies

A hybrid *echelon kanban/reorder point* (IK/RP) (Q, r) policy is a combination of an EK policy and an IS or an ES (Q, r) policy. In a hybrid EK/RP (Q, r) policy, echelon kanbans trace a loop within each stage and are detached from the FG output store of the last stage as in an EK policy. When an end item is consumed by a customer, *N* echelon kanbans (one for every stage) are detached from the item and become free. When an echelon kanban is detached from an end item in FG inventory it does not carry with it customer demand information, as in EK policies. Instead, demand is communicated according to the RP policy in place.

As in the case of hybrid IK/IS (Q, r) policies, we introduce two types of EK/RP (Q, r) policies: *synchronized* and *independent*. Their definitions are similar to the definitions of synchronized and independent IK/RP (Q, r) policies. Here again there are four hybrid EK/RP (Q, r) policies to consider: synchronized EK/IS (Q, r) and EK/ES (Q, r) policies and independent EK/IS (Q, r) and EK/ES (Q, r) policies. However only two of them are distinct, as is the case with IK/RP (Q, r) policies. The two distinct cases are synchronized EK/IS (Q, r) policies and independent EK/ES (Q, r) policies.

Queueing network model representations of a two-stage production-inventory system operating under a synchronized EK/IS (Q, r) policy and an independent EK/ES (Q, r) policy are shown in Figure 9 and Figure 10, respectively.

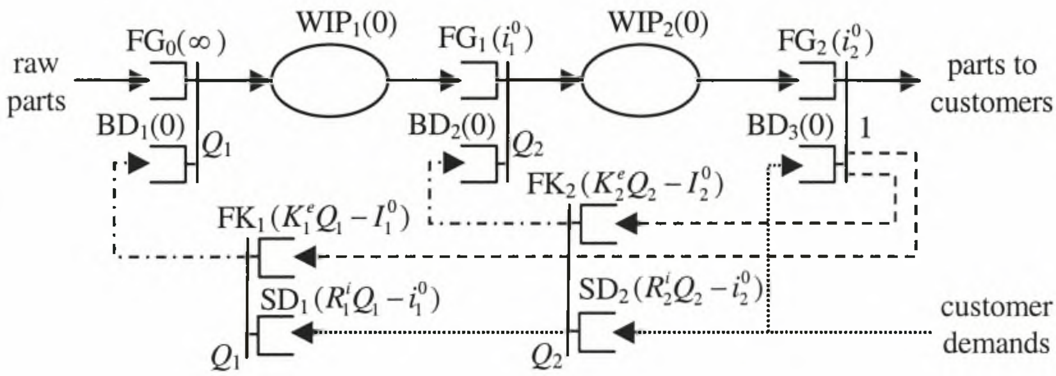

Figure 9: Queueing network model representation of a two-stage production-inventory system operating under a synchronized EK/ IS (Q, r) policy.

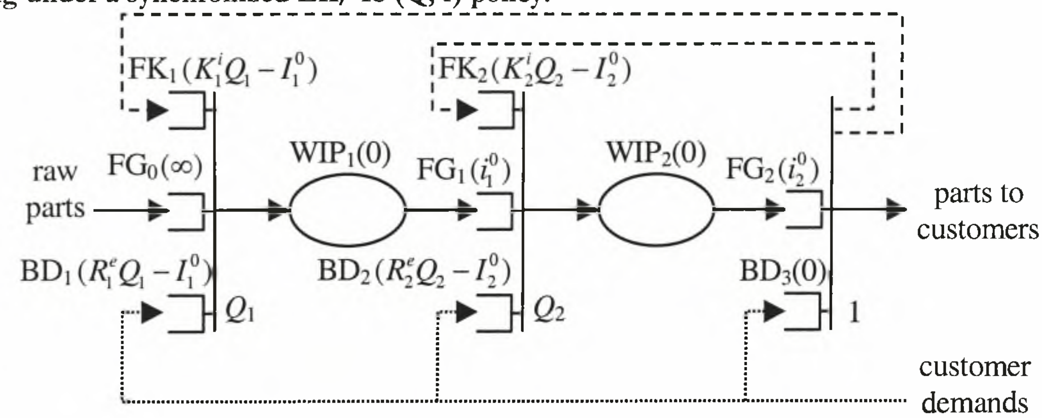

Figure 10: Queueing network model representation of a two-stage production-inventory system operating under an independent EK/ES (Q, r) policy.

Most of our discussion on hybrid IK/RP (Q, r) policies can be extended to hybrid EK/RP (Q, r) policies, but we will present it in future work. Instead, we will only point out that as is the case with IK/RP (Q, r) policies, we can not say with certainty whether an independent EK/ES (Q, r) policy is superior to a synchronized EK/IS (Q, r) policy or vice versa, but that it would not be surprising if in many cases an independent EK/ES (Q, r) policy turned out to perform better than a synchronized EchelonKanban/InstallationStock (Q,r) policy.

Chapter 3

Multi-Stage Production-Inventory Control Policies with Lot Sizing and Advance Demand Information (A.D.I)

3.1. Introduction

In order to profit by the existence of Advance Demand Information, we must modify our control policy in a way that makes use of the extra information that is available. In that direction we integrate the classic control policies that we have examined in previous sections, with delay or else production order release mechanisms. In the simple case (where A.D.I is not available), a production order is released at each demand arrival. In other words, the release mechanism is triggered by an actual demand. In the modified systems, a production order is released l units before the due-date of an order. In this case, production orders are triggered by information signals rather than by actual demand arrivals. It is important to note that the release lead time l is a parameter of the policy which has to be optimized. Finally, it has to be stressed that the release lead time is not unrestricted as it is constrained by the demand lead time. Namely, l has to be less than or equal to τ .

The deliberate delay between placing and activating a replenishment order depends on the so-called installation and echelon planned lead times associated with each stage. These lead times are defined as follows. The *installation planned lead time* of stage n is denoted by l_n and is a specified fixed control parameter that is usually related to the flow time of a typical part through the facility of the stage. The *echelon planned lead time* of a stage is denoted by L_n and is the sum of the installation planned lead times of the stage and all its downstream stages, i.e.,

$$L_n = \sum_{k=n}^{N} l_k , n = 1, 2, \ldots, N. \qquad (24)$$

With these definitions in mind, the time of activating a replenishment order at stage n is determined using an MRP-system logic by offsetting the due date of the demand that triggered the order by the stage echelon planned lead time, L_n. This means that the order is activated with no delay, if $L_n \geq T$, or with a delay equal to $T - L_n$ with respect to the demand arrival time, if $L_n < T$. In other words, the delay in activating an order, denoted T_n, is given by

$$T_n = \max[0, T - L_n],\ n = 1, 2, \ldots, N. \tag{25}$$

3.2. Installation and echelon stock (*Q*, *r*) policies with ADI

Two of the most common RPPs are *installation stock* (IS) and *echelon stock* (ES) (Q, r) policies. In this section we extend the definitions of IS and ES (Q, r) to include ADI.

3.2.1. Definition of installation and echelon stock (Q, r) policies with ADI

When a multi-stage production-inventory system is controlled by an IS or an ES (Q, r) policy with ADI, every stage is controlled by a (Q, r) rule based on its *inventory position*. This means that as soon as the inventory position of stage *n* falls at or below a reorder point r_n, a replenishment order is placed for the least integer number of lot sizes Q_n that raises the inventory position above r_n. Once a replenishment order has been placed at stage *n* it becomes an outstanding demand that will be activated after a time delay T_n, which is given by (25).

The difference between IS and ES (*Q*, *r*) policies with ADI lies in the definition of the inventory position. In an IS (Q, r) policy with ADI, the inventory position at stage *n* is defined as the *installation stock* at stage *n*, i.e., stock on hand (stage-*n* FG) plus outstanding orders (stage-*n* WIP + BD + OD) minus backorders (stage-(*n*+1) BD + OD). In an ES (Q, r) policy with ADI, the inventory position at stage *n* is defined as the *echelon stock* at stage *n*, i.e., the sum of the installation stocks at stage *n* and all its downstream stages. In other words, the installation and echelon stock at stage *n*, which are denoted by i_n and I_n, respectively, are defined as

$$i_n = \mathrm{OD}_n + \mathrm{BD}_n + \mathrm{WIP}_n + \mathrm{FG}_n - (\mathrm{OD}_{n+1} + \mathrm{BD}_{n+1}),\ n = 1, 2, \ldots, N, \tag{26}$$

$$I_n = \mathrm{OD}_n + \mathrm{BD}_n + \sum_{k=n}^{N} (\mathrm{WIP}_k + \mathrm{FG}_k) - (\mathrm{OD}_{N+1} + \mathrm{BD}_{N+1}),\ n = 1, 2, \ldots, N, \tag{27}$$

and are related as follows:

$$I_n = \sum_{k=n}^{N} i_k,\ n = 1, 2, \ldots, N, \tag{28}$$

$$i_n = I_n - I_{n+1},\ n = 1, 2, \ldots, N\text{-}1, \text{ and } i_N = I_N. \tag{29}$$

With the above definitions in mind, the decision to place an order at each stage is based on local information in an IS policy, and on global information in an ES policy. The parameters

Q_n and r_n are in general different for each stage. We make the common assumption that the order lot sizes satisfy:

$$Q_n = j_n \cdot Q_{n+1},\ n = 1, 2, \ldots, N, \text{ and } Q_{N+1} = 1, \tag{30}$$

for some positive integers j_n. Assumption (30) is necessary if the rationing policy is to satisfy all or nothing of a production order, because then the installation stock at every stage should always consist of an integer number of downstream lot sizes (except for the last stage where the rationing policy allows the partial satisfaction of a customer order as long as stock is available). Besides simplifying material handling, the integer ratio constraint (30) also simplifies analysis significantly. The cost increase due to constraint (30) is likely to be insignificant due to the insensitivity of inventory costs to the choice of order quantities.

A queuing network model representation of a two-stage production-inventory system operating under an IS (Q, r) policy with ADI is shown in Figure 11.

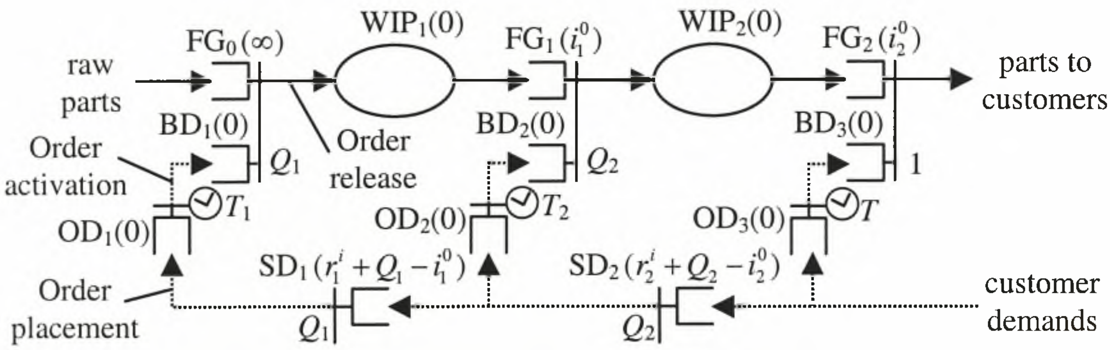

Figure 11: Queuing network model representation of a two-stage production-inventory system operating under an IS (Q, r) policy with ADI.

The symbolism used in Figure 11 and all other similar figures that follow in the rest of the thesis is as follows. The ovals represent WIP facilities, and the queues followed by vertical bars represent synchronization stations. Queues are labeled according to their content, and their initial value is indicated inside parentheses. For example, the queue representing the FG output store of stage 1 is labeled FG_1 and its initial value is i_1^0. Every synchronization station has a marking on its side. This marking indicates either the lot size needed to activate the synchronization station, i.e., the minimum number of customers that must be present in each queue to activate the synchronization station, or the time delay before the synchronization station may be activated. In the second case, the synchronization station has a clock next to it.

To clarify matters, let us look at some examples. Queues FG_1 and BD_2 are linked in a synchronization station marked with "Q_2." This means than as soon as there are at least Q_2 parts in FG_1 and Q_2 demands in BD_2, then exactly Q_2 parts depart from FG_1 and are released into WIP_2. At the same time, exactly Q_2 backordered demands depart from BD_2 and are discarded since they are satisfied. Another example is the synchronization station consisting of a single queue, SD_1, which is marked with "Q_1." This marking means that as soon as there are at least Q_1 demands in SD_1, then exactly Q_1 demands depart from SD_1 and are placed into queue BD_1. A final example is the synchronization station consisting of a single queue, OD_1, which is marked with a clock followed by "T_1." This marking means that when an order enters queue OD_1 it stays there for exactly T_1 time units before it is activated, i.e., before it departs from OD_1 and enters into BD_1.

The queuing network model representation of a two-stage production-inventory system operating under an ES (Q, r) policy with ADI is shown in Figure 12.

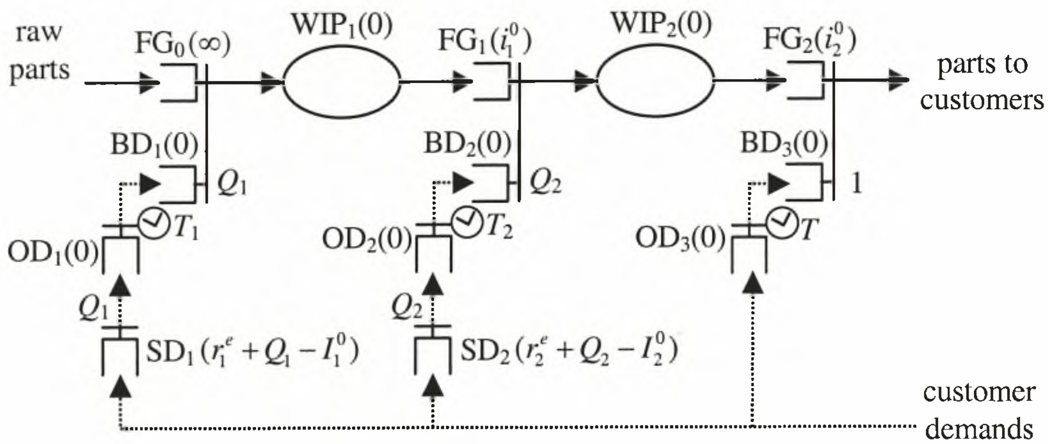

Figure 12: Queuing network model representation of a two-stage production-inventory system operating under an ES (Q, r) policy with ADI.

The local information nature of IS (Q, r) policies with ADI and the global information nature of ES (Q, r) policies with ADI is reflected in the way customer demand information is communicated to all stages, as can be seen in Figure 11 and Figure 12, respectively. In an IS policy, customer demand information is communicated from a stage to its previous upstream stage only when an order is placed at the former stage. In an ES policy, on the other hand, a customer demand is communicated to all stages immediately upon its arrival to the system.

Notice that if the demand lead time T is equal to zero, i.e., if there is no ADI, then all the delays T_n are also equal to zero by (25). In this case, the models in Figure 11 and Figure 12 can be further simplified by merging queue OD_n into queue BD_n, for n = 1, 2. Even if the demand lead time T is not equal to zero, however, i.e., if there is ADI, the models in Figure 11 and Figure 12 have exactly the same structure as the corresponding models with no ADI, once we view queues OD_n and BD_n as one combined queue with timer in it. This is an important observation, because many of the properties of IS and ES (Q, r) policies with no ADI developed in Axsater and Rosling (1993) and reinterpreted in Liberopoulos and Dallery (2002) carry over to the situation where there is ADI, as we will see next.

3.2.2. Behavior and properties of IS and ES (Q, r) policies with ADI

First, let us take a closer look at IS (Q, r) policies with ADI. We assume that the initial installation stock FG inventory positions in an IS (Q, r) policy with ADI satisfy $r_n^i < i_n^0 \leq r_n^i + Q_n$ for all n, where r_n^i are the installation stock reorder points. These conditions will anyway be satisfied as soon as an order has been placed by every stage. The following property is evident by looking at Figure 11.

Property 7. *An IS (Q, r) policy with ADI is always nested in the sense that when an order is placed at stage n, then orders must simultaneously be placed at all downstream stages as well.*

Without loss of generality, we also assume that

$$i_n^0 = r_n^i + k_n \cdot Q_{n+1},\ n = 1, 2, \ldots, N, \tag{31}$$

where k_n is an integer such that $1 \leq k_n \leq j_n$. Assumption (31) guarantees that the inventory of stage n is at the reorder point exactly when placing an order. Under some fairly non-restrictive assumptions on the customer demand arrival process, it must happen at some time that just enough customer demands have arrived to the system so that the inventory position of stage N is at the reorder point r_N^i and all stages place their orders simultaneously. This means that after ordering, all installation stock inventory positions will be at their maximum levels, i.e., $i_n = r_n^i + Q_n$, $n = 1, 2, \ldots, N$, irrespectively of the initial installation stock inventory positions. Moreover, if no customer demands arrive for some time, all FG buffers will end up at their maximum levels, i.e., at $r_n^i + Q_n$, $n = 1, 2, \ldots, N$, and all other queues will be empty. This state is regenerative in that it does not depend on the initial installation stock positions, and it implies the following property.

Property 8. *The behavior of an IS (Q, r) policy does not depend on the initial installation stock positions i_n^0, but only on the echelon planned lead times L_n, the reorder points r_n^i and the reorder quantities Q_n.*

Even though by Property 8 the behavior of an IS (Q, r) policy with ADI does not depend on the initial installation stock positions, the initial installation stock positions do play a role. Namely, the initial installation stock positions given by (31) determine the number of demands for stage-n FG that must arrive before a replenishment order of size Q_n is placed at stage n. There are two extreme cases: one where $k_n = j_n$ and another where $k_n = 1$. If $k_n = j_n$, the initial installation stock positions are equal to their maximum levels, i.e., $i_n^0 = r_n^i + j_n Q_{n+1} = r_n^i + Q_n$. In this case, exactly Q_n demands, i.e., j_n lots of demands of size Q_{n+1}, must arrive before a replenishment order of size Q_n is placed at stage n. On the other hand, if $k_n = 1$, the initial installation stock positions are equal to their minimum levels, i.e., $i_n^0 = r_n^i + Q_{n+1}$. In this case, exactly one lot of demands of size Q_{n+1} must arrive before a replenishment order of size Q_n is placed at stage n. This is exactly how an MRP system with fixed order quantity as its lot sizing rule works in a continuous review setting, as is stated by the following property.

Property 9. *If the flow time of every replenishment order though the WIP facility of stage n is constant and equal to l_n, and $k_n = 1$ so that $i_n^0 = r_n^i + Q_{n+1}$, for n = 1, 2,..., N, the resulting IS (Q, r) policy with ADI behaves exactly like an MPR system with fixed order quantity as its lot sizing rule.*

Property 9 is an important observation because it states that an MRP system with fixed order quantity is equivalent to an IS (Q, r) policy with ADI.

If the rationing policy is to satisfy all or nothing of a production order, we must assume that not only does the integer constraint (30) hold, but also that r_n^i is an integer multiple of Q_{n+1}, for $n = 1, 2, \ldots, N-1$, i.e. that $r_n^i = b_n Q_{n+1}$, where b_n is a positive integer.

Finally, the following property concerns the relationship between IS and ES (Q, r) policies with ADI.

Property 10. *An IS (Q, r) policy with ADI (with initial installation stock positions equal to their maximum level) can always be replaced by an equivalent ES (Q, r) policy with ADI with initial echelon stock positions*

$I_n^0 = \sum_{k=n}^{N} i_k^0 = r_n^e + Q_n$, *n = 1, 2, ..., N,*

where r_n^e are the echelon stock reorder points in the equivalent ES policy and are given by

$$r_n^e = r_n^i + \sum_{k=n+1}^{N} (r_k^i + Q_k),\ n = 1, 2, \ldots, N-1, \text{ and } r_N^e = r_N^i .$$

A similar analysis can be performed for ES (Q, r) policies with ADI. Here again we assume that the initial echelon stock FG inventory positions in an ES (Q, r) policy with ADI satisfy $r_n^e < I_n^0 \leq r_n^e + Q_n$, for all *n*, where r_n^e are the echelon stock reorder points. These conditions will anyway be satisfied as soon as an order has been placed at every stage. Unlike IS (Q, r) policies with ADI, ES (Q, r) policies with ADI generally depend on the initial echelon stock positions I_n^0 as well as on the echelon planned lead times L_n, the echelon stock reorder points r_n^e, and the reorder quantities Q_n. Also, unlike IS (Q, r) policies with ADI, ES (Q, r) policies with ADI are not always nested. If an ES (Q, r) policy with ADI is nested, however, then it can be replaced by an equivalent IS (Q, r) policy with ADI; otherwise, it can not. Axsater and Rosling (1993) give a necessary and sufficient condition for an ES (Q, r) policy to be nested when there is no ADI. The same condition holds when there is ADI. This condition is given by the following property.

Property11.*An ES (Q, r) policy with ADI is nested if the initial installation stock inventory positions satisfy*

$$i_n^0 = I_n^0 - I_{n+1}^0 = r_n^e - r_{n+1}^e + (k_n - 1) \cdot Q_{n+1},\ n = 1, 2, \ldots, N\text{-}1, \tag{32}$$

for some positive integers k_n such that $1 \leq k_n \leq j_n$. If condition (32) holds, the resulting nested ES policy can be replaced by an equivalent IS policy with initial installation stock positions

$$i_n^0 = I_n^0 - I_{n+1}^0 = r_n^i + k_n \cdot Q_{n+1},\ n = 1, 2, \ldots, N\text{-}1, \text{ and } i_N^0 = I_N^0,$$

where r_n^i are the reorder points in the equivalent IS policy and are given by

$$r_n^i = r_n^e - r_{n+1}^e - Q_{n+1},\ n = 1, 2, \ldots, N\text{-}1, \text{ and } r_N^i = r_N^e.$$

Clearly, if condition (32) holds, the resulting nested ES policy does not depend on the initial installation stock positions and therefore on the initial echelon stock positions. We end our discussion of IS and ES (Q, r) policies with ADI with some concluding remarks.

IS (Q, r) policies with ADI are nested ES (Q, r) policies with ADI and are therefore special cases of the letter policies. An important implication of the preceding analysis is that the behavior of an IS policy does not depend on the initial FG inventory positions. The behavior of an ES policy, on the other hand generally depends on the initial FG inventory positions, except when condition (32) holds. In the latter case the resulting ES policy is nested and can be replaced by an equivalent IS policy. In other words, IS policies with ADI depend on three parameters per stage, L_n, r_n and Q_n, i.e. they have three degrees of freedom on the choice of parameters per stage, whereas ES policies generally depend on four parameters per stage, L_n, r_n, Q_n, and I_n^0, i.e., they have four degrees of freedom per stage.

An MRP system with fixed order quantity as its lot sizing rule is equivalent to an IS (Q, r) policy with ADI. Given that an IS (Q, r) policy with ADI is a special case of an ES (Q, r) policy with ADI, this means that an MRP system is a special case of an ES (Q, r) policy with ADI. In a sense, an ES (Q, r) policy with ADI may be viewed as a broader definition of an MRP system. It appears that it is this definition of an MRP system that Asxater and Rosling (1994) invoke when they claim that any IS (Q, r) policy and any ES (Q, r) policy (with no ADI) can be duplicated by an MRP system (i.e., an ES (Q, r) policy with ADI).

Finally, in case of unit lot sizes, i.e., when $Q_n = 1$, $n = 1,2,\ldots,N+1$, an IS (Q, r) policy with ADI is identical to an ES (Q, r) policy with ADI, and they are both equivalent to a *base stock policy with a release time parameter* (Karaesmen et al, 2002).

3.3. Installation and echelon kanban policies with A.D.I

The original kanban system developed at Toyota's automobile production lines is the single technique most closely associated with JIT practices. The last two decades have seen a surge in the literature on kanban systems, but there seems to be no agreed upon definition of what a kanban system is (Liberopoulos and Dallery, 2000). Liberopoulos and Dallery (2002) used the notions of installation kanbans and echelon kanbans to define installation kanban and echelon kanban policies, respectively, in cases where there is no ADI. In both policies the placement a replenishment production order to the facility of each stage, triggered by the arrival of a customer demand, is initiated after the consumption of an end-item from FG inventory. In the case where there is ADI, since the consumption of an end-item from FG inventory is activated T time units after the arrival time of the demand, the demand lead time T is totally unexploited as far as the placement of the replenishment policy is concerned. Hence, by their nature, installation kanban and echelon kanban policies can not take advantage of ADI, and therefore there is not much to say about kanban policies with ADI. Nevertheless, in the rest of this section, we will recall some of the basic facts concerning kanban policies in cases where there is no ADI presented in Liberopoulos and Dallery (2002) because we will use them in a later section in our discussion of hybrid policies with ADI.

3.3.1. Definition of installation and echelon kanban policies with A.D.I

In a multi-stage production-inventory system controlled by an *installation kanban* (IK) or an *echelon kanban* (EK) policy, every stage n has associated with it a finite number of authorization cards or kanbans. This number is equal to an integer multiple of the stage lot size Q_n. A kanban may be either free or attached onto a part. A free stage-n kanban is used to signal a customer demand for one part at stage n. Kanbans, like parts, move in lots of size Q_n. Specifically, when Q_n free stage-n kanbans have accumulated at stage n, an order of equal size, i.e., Q_n, is placed at stage n. If Q_n parts are available in stage-(n-1) FG inventory, the free kanbans are attached onto the parts and the combined lot, i.e., the Q_n parts plus their kanbans, is released into the WIP facility of stage n. The kanbans remain attached to the parts until the combined lot reaches a certain *final* FG output store. When a part exits the FG output store, because it is consumed by the next downstream stage or by a customer (if the final FG output store is the output store of the last stage), the kanban that was attached to it is detached and becomes free. This free kanban is used once again to signal a customer demand for one part at stage n so that when Q_n free kanbans have accumulated, an order of equal size is placed at stage n.

The difference between IK and EK policies lies in the definition of the *final* FG output store, i.e., the point after which kanbans are detached from parts. In an IK policy, the final FG output store at stage n is the FG output store of stage n. In an EK policy, it is the FG output store of the last stage, i.e., stage N. This means that in an IK policy, a stage-n kanban follows a part through the WIP facility and the FG output store of stage n and is detached from the part after the part leaves the FG output store of stage n. In an EK policy, on the other hand, a stage-n kanban follows a part through the WIP facilities and FG output stores of stages n through N and is detached from the part after the part leaves the FG output store of stage N. This implies that in an IK policy, the decision to place an order at each stage is based on local information, whereas in an EK policy it is based on global information from all downstream stages. The kanbans used in IK and EK policies are referred to as *installation kanbans* and *echelon kanbans*, respectively. Note that in an IK policy, every part in the WIP facility or FG output buffer at stage n has attached onto it a stage-n installation kanban. In an EK policy, on the other hand, every part in the WIP facility or FG output buffer at stage n has attached onto it one echelon kanban from each of stages 1 through n. This means that in an EK policy, when an end item is consumed by a customer, N echelon kanbans are detached from the part and become free.

The above definition of IK policies is closely related to most of the definitions of kanban systems encountered in the literature. The only difference between this definition and most other definitions is that in this definition we assume that each kanban is attached onto a single part, whereas in most other definitions it is assumed that each kanban is attached onto a container that carries an entire lot of parts. This difference, however, has no effect on the behavior of the policy, since in both definitions a replenishment order for a new lot of Q_n parts is placed at stage n, when an entire lot of Q_n parts in FG inventory has been consumed. The reason for which we assume that each kanban is attached onto a single part rather than on a lot of parts is because it helps us better accommodate the definition of EK policies.

The original queuing network model representations of a two-stage production-inventory system operating under an IK and an EK policy are shown in Figure 13 and Figure 14 respectively.

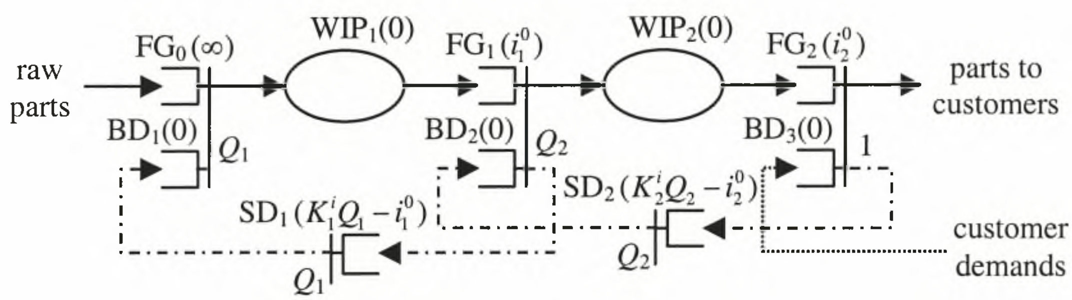

Figure 13: Original queuing network model representation of a two-stage production-inventory system operating under an IK policy.

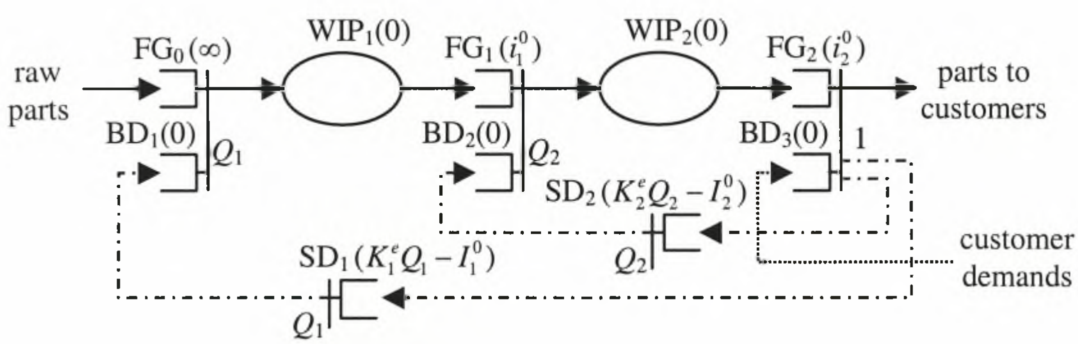

Figure 14: Original queuing network model representation of a two-stage production-inventory system operating under an EK policy.

The models of Figure 13 and Figure 14 can be further simplified by merging queue SD_n into queue BD_n, for n = 1, 2. The simplified models are shown in Figure 15 and Figure 4 16, respectively.

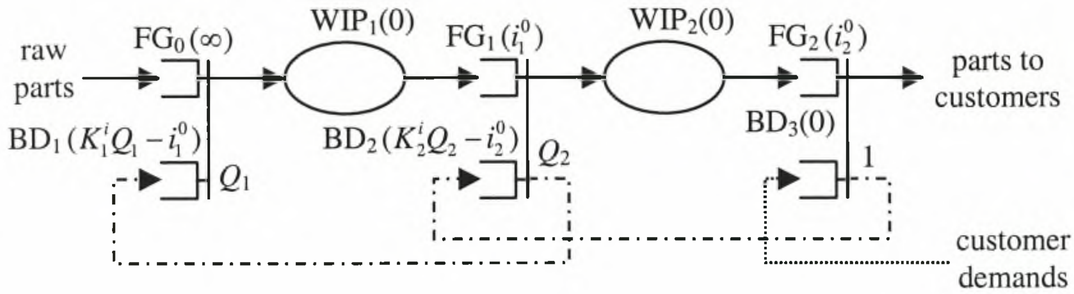

Figure 15: Simplified queuing network model representation of a two-stage production-inventory system operating under an IK policy.

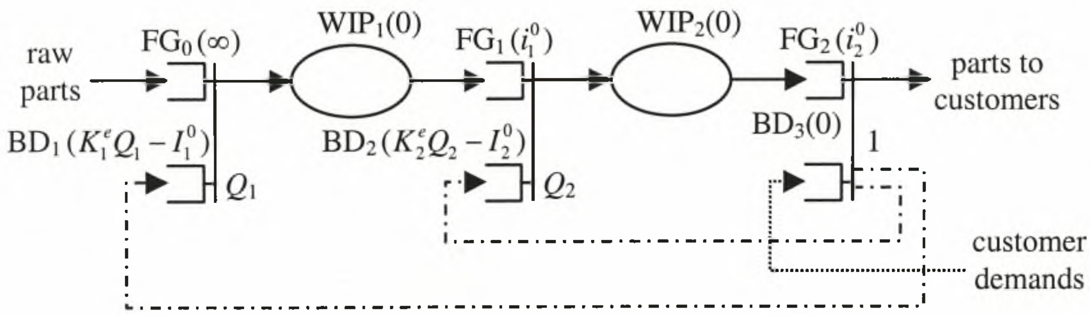

Figure 16: Simplified queuing network model representation of a two-stage production-inventory system operating under an EK policy.

Next, we discuss IK policies in some more detail.

3.3.2. Behavior and properties of installation kanban policies with A.D.I

Axsater and Rosling (1993) view IK policies (they call them 'original kanban' policies) as being inherently IS (Q, r) policies, where a) backorders are not subtracted from the definition of the installation stock inventory position, and the reorder point at stage n is defined as $r_n^i = (K_n^i - 1)\cdot Q_n$, where K_n^i is an integer such that $K_n^i \geq 1$ and $K_n^i \cdot Q_n$ is the number of installation kanbans at stage n, or b) the inventory position is defined exactly as in IS policies, and the reorder point is occasionally decreased (when there are backorders). It is important to note that in an IK policy the reorder point r_n^i is an integer multiple of the stage lot size Q_n, whereas in an IS (Q, r) policy it need not be.

We assume that the initial installation stock FG inventory positions in an IK policy satisfy $(K_n^i - 1)\cdot Q_n < i_n^0 \leq K_n^i \cdot Q_n$, for all n, as was the case in an IS (Q, r) policy. Without loss of generality we also assume that $i_n^0 - (K_n^i - 1)\cdot Q_n = k_n \cdot Q_{n+1}$, where k_n is an integer such that $1 \leq k_n \leq j_n$. This assumption guarantees that the inventory of stage n is at the reorder point exactly when ordering. It also guarantees that $i_n^0 \geq Q_{n+1}$, so that the system will never come to a deadlock. Under some fairly non-restrictive assumptions on the customer demand arrival process, it must happen at some time that just enough customer demands have arrived to the system so that an order is placed at stage 1. Suppose that it also happens that no additional customer demands arrive for some time. Then, all WIP facilities will be cleared out of parts and all installation stock inventory positions will be at their maximum level. Therefore, without loss of generality, we may assume that the initial installation stock positions are equal to their maximum level, i.e., $i_n^0 = K_n^i \cdot Q_n$. In this case all the initial SD positions will be zero. This means that an IK policy does not depend on the initial installation stock positions but only on the reorder quantities Q_n and the integers K_n^i, which together with the Q_n define the reorder points r_n^i.

Axsater and Rosling (1993) conclude that IK policies are inherently IS (Q, r) policies with some limitations and are therefore inferior to IS (Q, r) policies, although in a more recent paper they recognize that this conjecture is not always correct (Axsater and Rosling, 1999). In our view there is a fundamental difference between IS (Q, r) policies and IK policies. In an IK policy, demand is communicated at a stage only when FG inventory is consumed by the next downstream stage or by a customer. In an IS (Q, r) policy, on the other hand, demand is communicated at a stage irrespectively of whether FG inventory is consumed or not. This difference is quite evident when one compares Figure 3 to Figure 1.

A consequence of this difference is that an IK policy can not take advantage of ADI, as was mentioned earlier. Another consequence is that an IK policy is never nested in the sense that an IS (Q, r) policy is. A third consequence of the difference between IK and IS (Q, r) policies is that in an IK policy, the WIP + FG inventory at every stage is always bounded by the number of installation kanbans. In an IS (Q, r) policy, on the other hand, although the FG inventory at every stage is bounded by the initial FG inventory position, the WIP inventory is unbounded.

3.3.3. Disadvantages of installation and echelon kanban policies with A.D.I

Liberopoulos and Dallery (2002) mention that an important advantage of kanban policies over their reorder point counterpart policies is that the former policies impose an upper bound on the WIP + FG inventory. This advantage implies inventory holding cost savings. One of the disadvantages of kanban policies, however, is that they do not communicate customer demand information to all upstream stages as quickly as their corresponding RPPs. This is because in kanban policies customer demand information is communicated only when a lot of kanbans is detached, and kanbans are detached only when FG parts are consumed. This disadvantage has a direct impact on customer service since it implies longer customer response times, particularly if customer demand is highly variable. It also implies that the capacity of the system depends on the number of kanbans. Another disadvantage of kanban policies is that kanban policies can not exploit ADI, as was mentioned earlier.

One way to overcome these disadvantages and increase customer service and system capacity would be to uncouple a) the actions of detaching a kanban and communicating demand information and b) the initial FG inventory and reorder point from the number of kanbans at every stage. This approach can be implemented by combining an IK or an EK policy with an IS or an ES (Q, r) policy with ADI to form a more sophisticated hybrid policy. In the rest of this thesis we will discuss combinations IK policies with IS or ES (Q, r) policies with ADI.

3.4. Hybrid Installation Kanban/Reorder Point (Q, r) policies with ADI

A hybrid *installation kanban/reorder point* (IK/RP) (Q, r) policy with ADI is a combination of an IK policy with an IS or an ES (Q, r) policy with ADI. In a hybrid IK/RP (Q, r) policy with ADI, installation kanbans trace a loop within each stage and are detached from the FG output store of the stage as in an IK policy. However, when an installation kanban is detached from a part in FG inventory, it does not carry with it customer demand information, as in an IK policy. Instead, demand is communicated according to the RPP with ADI in place

3.4.1. Definition of hybrid IK/IS nd IK/ES (Q, r) policies with ADI

We differentiate between the following types of IK/RP (Q, r) policies with ADI: *independent, synchronized & Delay Before Synchronization (DBS)* and *synchronized & Delay After Synchronization (DAS).*

In both synchronized and independent IK/RP (Q, r) policies with ADI, the actions of detaching a kanban and communicating demand are uncoupled. Moreover, in both cases, the initial FG inventory and the reorder point are not determined by the number of kanbans, as is the case in IK policies. Finally, in both cases, customer demands are communicated according to the RPP in place. The difference between the two cases is that in a synchronized IK/RP (Q,

r) policy with ADI, when a stage-n installation kanban is detached from a part in stage-n FG inventory, it is used to *authorize the placement* of a replenishment order for one part at stage n. In an independent IK/RP (Q, r) policy with ADI, on the other hand, when a stage-n installation kanban is detached from a part in stage-n FG inventory, it is used to *authorize the release* of a replenishment order for one part at stage n (the difference between the placement and the release of an order was clarified in previous section). In other words, in a synchronized IK/RP (Q, r) policy with ADI, the placement of orders is synchronized with the trajectory of installation kanbans, whereas in an independent IK/RP (Q, r) policy with ADI the placement of orders is independent of the trajectory of installation kanbans. In both synchronized and independent IK/RP (Q, r) policies with ADI, the decision to authorize the placement or release of an order at each stage is based on local information, since it depends on the availability of installation kanbans. The decision to place an order at each stage, on the other hand, is based on local information, if the RPP in place is an IS (Q, r) policy with ADI, and on global information, if the RPP is an ES (Q, r) policy with ADI.

Especially for the case of *synchronized* policies, arises another matter; where to place the delay mechanisms. Before or after the synchronization of Kanban and order information;

With the above definitions in mind, there are six hybrid IK/RP (Q, r) policies with ADI to consider:

1. Independent IK/ES (Q, r) policies with ADI
2. Synchronized DBS IK/ES (Q, r) policies with ADI
3. Synchronized DAS IK/ES (Q, r) policies with ADI
4. Independent IK/IS (Q, r) policies with ADI
5. Synchronized DAS IK/IS (Q, r) policies with ADI
6. Synchronized DBS IK/IS (Q, r) policies with ADI

The only really distinct hybrid IK/RP policies are:

1. Independent IK/ES (Q, r) policies with ADI
2. Synchronized DAS IK/ES (Q, r) policies with ADI
3. Synchronized DAS IK/IS (Q, r) policies with ADI

This is because 1) an independent IK/IS (Q, r) policy with ADI is equivalent to a nested independent IK/ES (Q, r) policy with ADI and therefore a special case of the latter policy, and 2) a synchronized DBS IK/ES (Q, r) policy with ADI is identical to an independent IK/ES

(Q, r) policy with ADI and 3) a synchronized DBS IK/IS (Q, r) policy with ADI is identical to a synchronized DBS IK/ES (Q, r) policy with ADI which is identical to an independent IK/ES (Q, r) policy with ADI. We will therefore focus out attention only on synchronized DAS IK/IS (Q, r) policies with ADI, independent IK/ES (Q, r) policies with ADI and Synchronized DAS IK/ES (Q, r) policies with ADI.

3.4.2. Behavior and properties of Syn/zed DAS IK/IS (Q, r) policies with ADI

Queuing network model representations of a two-stage production-inventory system operating under a Synchronized DAS Installation Kanban/Installation Stock (Q, r) policies with ADI is shown in Figure 17:

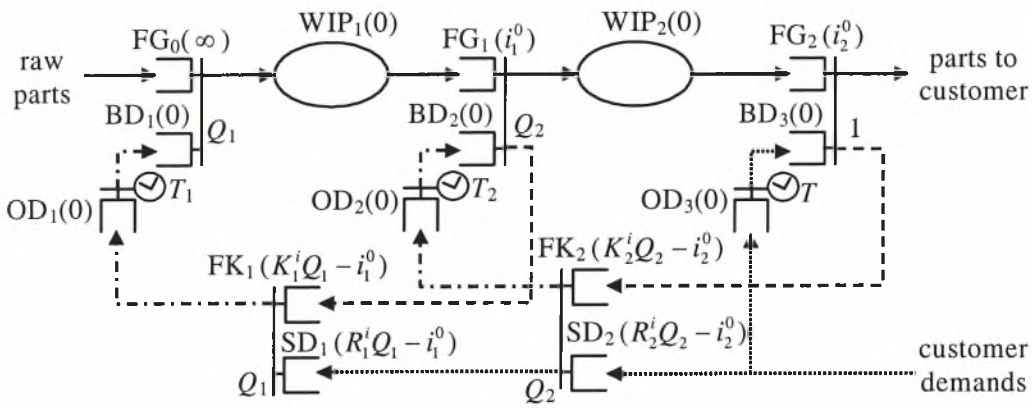

Figure 17: Queuing network model representation of a two-stage production-inventory system operating under a synchronized DAS IK/IS (Q, r) policy with ADI.

We assume that the initial installation stock FG inventory positions in a synchronized DAS IK/IS (Q, r) policy with ADI satisfy $(R_n^i - 1)\cdot Q_n < i_n^0 \le R_n^i \cdot Q_n$, for all *n*, as was the case in IS (Q, r) policies with ADI, where R_n^i are integers such that $1 \le R_n^i \le K_n^i$. Without loss of generality, we also assume that $i_n^0 - (R_n^i - 1)\cdot Q_n = k_n \cdot Q_{n+1}$, where k_n is an integer such that $1 \le k_n \le j_n$. This assumption guarantees that the inventory of stage *n* is at the reorder point exactly when ordering. It also guarantees that $i_n^0 \ge Q_{n+1}$, so that the system will never come to a deadlock. Similarly to our analysis of IS (Q, r) policies with ADI, under some fairly non-restrictive assumptions on the customer demand arrival process, we may assume that the initial installation stock positions are equal to their maximum level, i.e., $i_n^0 = R_n^i \cdot Q_n$. In this case, all the initial SD positions will be zero. This means that a synchronized DAS IK/IS (Q, r) policy with ADI does not depend on the initial installations stock positions but only on parameters L_n, Q_n, K_n^i, and R_n^i.

A synchronized DAS IK/IS (Q, r) policy with ADI is never nested in the sense that an IS (Q, r) policy with ADI is, i.e., in the sense that when an order is placed at stage *n*, then orders must simultaneously be placed at all downstream stages as well (except when $K_n^i = \infty$, for all *n*, as we will see below).

A synchronized DAS IK/IS (Q, r) policy with ADI includes IK policies and IS (Q, r) policies with ADI as special cases. Specifically, a synchronized DAS IK/IS (Q, r) policy with ADI with $K_n^i = R_n^i$, for all *n* is equivalent to an IK policy, i.e., a policy which does not exploit ADI, as was mentioned earlier. A synchronized DAS IK/IS (Q, r) policy with ADI with $K_n^i = \infty$, for all *n*, is equivalent to an IS (Q, r) policy with ADI (and hence, an MRP system) with installation stock reorder points equal to $r_n^i = (R_n^i - 1) \cdot Q_n$, and is therefore nested. Any other synchronized DAS IK/IS (Q, r) policy with ADI with $R_n^i < K_n^i < \infty$ is never nested, imposes an upper bound on the WIP + FG inventory, while exploiting ADI for better replenishment control; however, it does not take full advantage of ADI, since the communication of demands from a stage *n* to the previous upstream stage *n*-1 may be blocked due to the lack of free stage-*n* kanbans in queue FK_n.

The dynamics of kanban and material flow in the synchronized DAS IK/IS (Q, r) policy with ADI, is essential in order to determine the impact of the choice of system parameters on the departure times of parts from various points in the system. The dynamics of the independent IK/ES (*Q*, *r*) policy, can be described by recursive evolution equations that utilize operators " + " and " max " only. These equations relate the timing of a particular event in the policy to the timings of events that must precede it.
The timing of events in a syn/zed DAS IK/IS (Q, r) policy with ADI is shown in Figure 18:

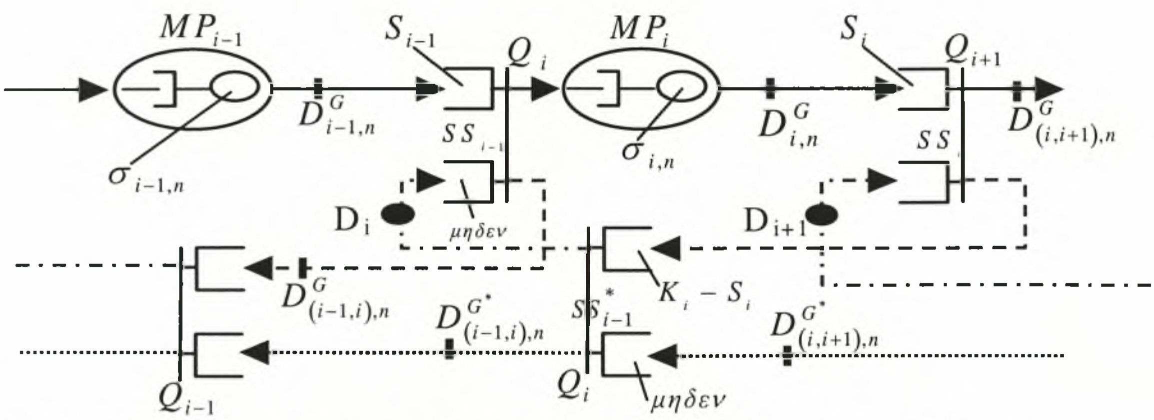

Figure 18: Timing of events in a synchronized DAS IK/IS (Q, r) policy with ADI

The evolution equations of the above policy are the following:

$$D_{i,n}=\sigma_{i,n}+ max\left(D_{i,n-1},D_{(i-1,i),n}\right),i=1,...,N$$

$$D_{(i-1,i),n}= max\left(D_i+D^*_{(i-1,i),n},D_{i-1,\left\lceil\frac{n}{Q_i}\right\rceil Q_i-S_{i-1}}\right),i=2,...,N+1$$

Where: $$D^*_{(i-1,i),n}= max\left(D_{(i,i+1),\left\lceil\frac{n}{Q_i}\right\rceil Q_i-(K_i-S_i)},D^*_{(i,i+1),\left\lceil\frac{n}{Q_i}\right\rceil Q_i}\right),i=2,...,N+1$$

As in we see the above equation is recursive in that it expresses $D^*_{(i-1,i),n}$ in terms of $D^*_{(i,i+1),\left\lceil\frac{n}{Q_i}\right\rceil Q_i}$. The expansion of this recursive equation, unfortunately doesn't lead as in a compact non-recursive form. This is because the timing of events in the synchronized DAS IK/IS (Q, r) policy with ADI, is much too complicated. This inevitably leads to the conclusion that synchronized DAS IK/IS (Q, r) policy with ADI, is a very complex policy and hence difficult to implement.

3.4.3. Behavior and properties of Independent IK/ES (Q, r) policies with ADI

Queuing network model representations of a two-stage production-inventory system operating under a independent Installation Kanban/Echelon Stock (Q, r) policies with ADI is shown in Figure 19:

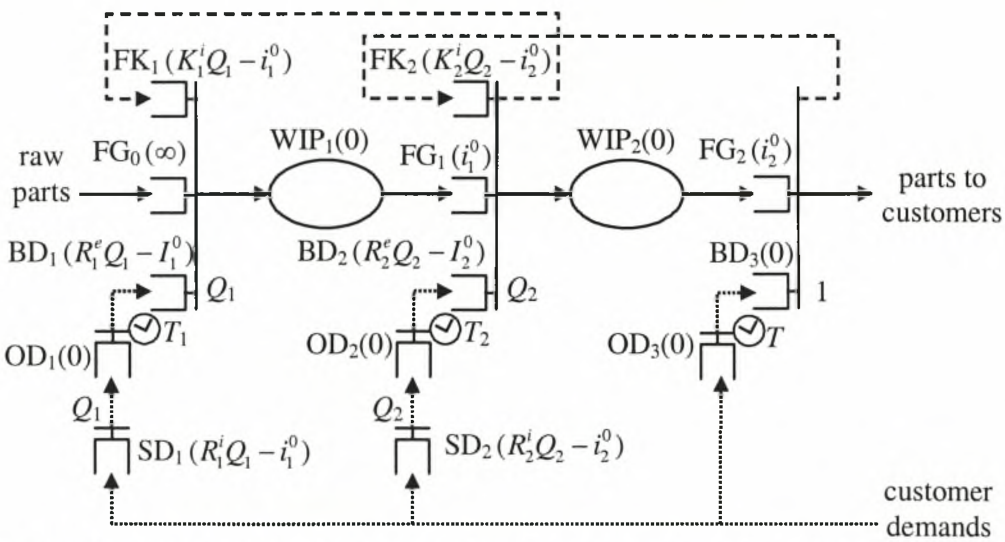

Figure 19: Queuing network model representation of a two-stage production-inventory system operating under an independent IK/ES (*Q*,*r*) policy with ADI.

A similar analysis as in the case of a synchronized DAS IK/IS (Q, r) policy with ADI can be carried out on independent IK/ES (Q, r) policies with ADI. We assume that in an independent IK/ES (Q, r) policy with ADI the initial echelon stock FG inventory positions satisfy $(R_n^e - 1)\cdot Q_n < I_n^0 \le R_n^e \cdot Q_n$, for all *n*, as is the case in an ES (Q, r) policy, where R_n^e are integers such that $R_n^e \cdot Q_n - R_{n+1}^e \cdot Q_{n+1} \le K_n^i \cdot Q_n$, or $R_n^e \cdot j_n - R_{n+1}^e \le K_n^i \cdot j_n$, for *n* = 1, 2, …, N-1, and $R_N^e \le K_N^i$.

A very important issue is to describe in detail the dynamics of kanban and material flow in the independent IK/ES (*Q*, *r*) policy, with ADI, in order to determine the impact of the choice of system parameters on the departure times of parts from various points in the system. The dynamics of the independent IK/ES (*Q*, *r*) policy, can be described by recursive evolution equations that utilize operators " + " and " max " only. These equations relate the timing of a particular event in the IK/ES (*Q*, *r*) policy to the timings of events that must precede it.

The timing of events in an independent IK/ES (*Q*, *r*) policy, with ADI, is shown in Figure 20:

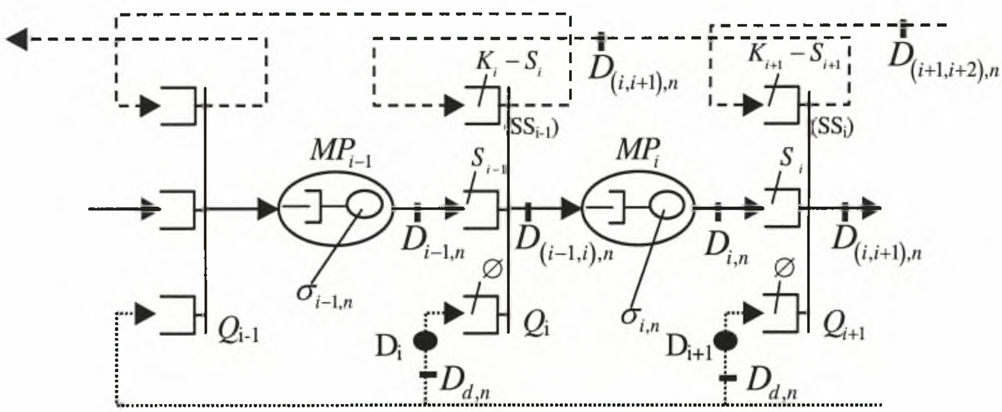

Figure 20: Timing of events in an independent IK/ES (*Q*, *r*) policy, with ADI

Proposition 3. *In a independent IK/ES (Q,r) policy, with ADI, in which MP_i consists of a single machine, the timings of events are related by the following evolution equations:*

$$D_{i,n} = \sigma_{i,n} + \max\left(D_{i,n-1}, D_{(i-1,i),n}\right), i = 1,\dots,N \tag{33}$$

$$D_{(i-1,i),n} = \max\left(D_i + D_{d,\left\lceil \frac{n}{Q_i}\right\rceil Q_i}, D_{(i,i+1),\left\lceil \frac{n}{Q_i}\right\rceil Q_i - (K_i - S_i)}, D_{i-1,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - S_{i-1}}\right), i = 2,\dots,N+1 \tag{34}$$

Proof. Equation (33) can be explained in a similar way as Equation (17) in the EKCS in Dallery's and Liberopoulos (2000). Equation (34) can be explained in a similar way as equation (18) in the EKCS in Dallery's and Liberopoulos (2000).

The symbolism $\left\lceil \frac{n}{Q_i} \right\rceil Q_i$ has to do with the batching of parts. Equation (34) is recursive in that it expresses $D_{(i-1,i),*}$ in terms of $D_{(i,i+1),*}$. Expanding this recursion yields:

i=N+1:

$$D_{(N,N+1),n} = \max\left(D_{N+1} + D_{d,\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}}, D_{(N+1,N+2),\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}-(K_{N+1}-S_{N+1})}, D_{N,\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}-S_N} \right) \Rightarrow$$

$$\boldsymbol{D_{(N,N+1),n} = \max\left(D_{N+1} + D_{d,\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}}, D_{N,\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}-S_N} \right), i = 2,\dots,N+1}$$

i=N

$$D_{(N-1,N),n} = \max\left(D_N + D_{d,\left\lceil \frac{n}{Q_N} \right\rceil Q_N}, D_{(N,N+1),\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}, D_{N-1,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-S_{N-1}} \right),$$

$\textit{οπου}$:

$$D_{(N,N+1),\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)} = \max\left(D_{N+1} + D_{d,\left\lceil \frac{\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}{Q_{N+1}} \right\rceil Q_{N+1}}, D_{N,\left\lceil \frac{\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}{Q_{N+1}} \right\rceil Q_{N+1}-S_N} \right),$$

$\textit{οπου}$:

$$\left\lceil \frac{\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}{Q_{N+1}} \right\rceil Q_{N+1} = \left\lceil \frac{\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(\nu-\xi)Q_N}{Q_{N+1}} \right\rceil Q_{N+1} = \left\lceil \left\{ \left\lceil \frac{n}{Q_N} \right\rceil -(\nu-\xi) \right\} \frac{Q_N}{Q_{N+1}} \right\rceil Q_{N+1} =$$

$$= \left\lceil \left\{ \left\lceil \frac{n}{Q_N} \right\rceil -(\nu-\xi) \right\} \mu \right\rceil Q_{N+1} = \left\lceil \frac{n}{Q_N} \right\rceil Q_N - (K_N - S_N),$$

$\textit{αρα}$:

$$D_{(N,N+1),\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)} = \max\left(D_{N+1} + D_{d,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}, D_{N,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-K_N} \right),$$

$\textit{αρα}$:

$$\boldsymbol{D_{(N-1,N),n} = \max\left(D_N + D_{d,\left\lceil \frac{n}{Q_N} \right\rceil Q_N}, D_{N+1} + D_{d,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}, D_{N,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-K_N}, D_{N-1,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-S_{N-1}} \right)}$$

Hence the final form of $D_{(i-1,i),n}$ is:

$$D_{(i-1,i),n} = max\left\{ D_{i-1,\left\lceil \frac{n}{Qi}\right\rceil Q_i - S_{i-1}}, \max_{j=i}^{N}\left(D_{j,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - S_j - \sum_{m=i}^{j}(K_m - S_m)}\right), \max_{j=i}^{N+1}\left(D_j + D_{d,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - \sum_{m=i}^{j-1}(K_m - S_m)}\right)\right\}, \text{ and}$$

so the final evolution equations have the following form:

$$\left\langle \begin{array}{l} D_{i,n} = \sigma_{i,n} + max\left(D_{i,n-1}, D_{(i-1,i),n}\right), i = 1,...,N \\ D_{(i-1,i),n} = max\left\{ D_{i-1,\left\lceil \frac{n}{Qi}\right\rceil Q_i - S_{i-1}}, \max_{j=i}^{N}\left(D_{j,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - S_j - \sum_{m=i}^{j}(K_m - S_m)}\right), \max_{j=i}^{N+1}\left(D_j + D_{d,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - \sum_{m=i}^{j-1}(K_m - S_m)}\right)\right\} \\ or \\ D_{(i-1,i),n} = \max_{j=i-1}^{N}\left\{ D_{j,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - S_j - \sum_{m=i}^{j}(K_m - S_m)}, D_{j+1} + D_{d,\left\lceil \frac{n}{Q_i}\right\rceil Q_i - \sum_{m=i}^{j}(K_m - S_m)}\right\} \end{array} \right\rangle \quad (35)$$

Unlike synchronized DAS IK/IS (Q, r) policies with ADI, independent IK/ES (Q, r) policies with ADI generally depend on the initial echelon stock positions I_n^e as well as on the parameters Q_n, K_n^e and R_n^e. An independent IK/ES (Q, r) policy with ADI may be nested in the sense that an ES (Q, r) policy with ADI may be nested, i.e., in the sense that when an order is placed at stage *n*, then orders must simultaneously be placed at all downstream stages as well. The condition for this to happen is exactly the same as in an ES (Q, r) policies with ADI and is given by expression (10), where $r_n^e = (R_n^e - 1)\cdot Q_n$. If this condition holds, the nested independent IK/ES (Q, r) policy with ADI can be replaced by an equivalent independent IK/IS (Q, r) policy with ADI, just as an ES (Q, r) policy with ADI can be replaced by an equivalent IS (Q, r) policy with ADI. The resulting nested independent IK/IS (Q, r) policy with ADI does not depend on the initial installation stock positions and therefore on the initial echelon stock positions. A nested synchronized (or equivalently independent) IK/ES (Q, r) policy, on the other hand, can not be replaced by an equivalent synchronized DAS IK/IS (Q, r) policy, because as was already mentioned above, a synchronized DAS IK/IS (Q, r) policy is never nested.

An independent IK/ES (Q, r) policy with ADI includes IK and ES (Q, r) policies with ADI as special cases. Specifically, an independent IK/ES (Q, r) policy with ADI with $K_n^i \cdot Q_n = R_n^e \cdot Q_n - R_{n+1}^e \cdot Q_{n+1}$, or $K_n^i \cdot j_n = R_n^e \cdot j_n - R_{n+1}^e$, n = 1, 2, ..., N-1, and $K_N^i = R_N^e$, is equivalent to an IK policy. An independent IK/ES (Q, r) policy with ADI with $K_n^i = \infty$ for every stage *n*, is equivalent to an ES (Q, r) policy with ADI (and hence an MRP system) with echelon stock reorder points equal to $r_n^e = (R_n^e - 1) \cdot Q_n$. Any other independent IK/ES (Q, r) policy with ADI with $R_n^e \cdot j_n - R_{n+1}^e < K_n^i \cdot j_n < \infty$ imposes an upper bound on the WIP + FG inventory, and still exploits ADI for better replenishment control, just as an ES (Q, r) policy with ADI does.

There are two limiting cases where an independent IK/ES (Q, r) policy with ADI is equivalent to an IK policy and to an ES (Q, r) policy with ADI, respectively. In fact, these are the only cases where a synchronized DAS IK/IS (Q, r) policy with ADI and an independent IK/ES (Q, r) policy with ADI are equivalent to each other. In any other case, a synchronized IK/IS (Q, r) policy with ADI and an independent IK/ES (Q, r) policy with ADI are never equivalent to each other. This means that if we take an IS (Q, r) policy with ADI and an equivalent nested ES (Q, r) policy with ADI, superimpose on each policy the same IK policy and synchronize the trajectory of installation kanbans with the placement of orders, the resulting synchronized IK/IS (Q, r) policy with ADI and synchronized IK/ES (Q, r) policy with ADI (which is equivalent to an independent IK/ES (Q, r) policy with ADI) will not be equivalent to each other. For this reason, we cannot say with certainty whether an independent IK/ES (Q, r) policy with ADI performs better or worse than a synchronized IK/IS (Q, r) policy with ADI. It would not be surprising, however, if in many cases an independent IK/ES (Q, r) policy with ADI turned out to perform no worse than a synchronized IK/IS (Q, r) policy with ADI because the former policy uses global information, whereas the latter policy uses only local information.

Moreover, as we observed earlier, synchronized DAS IK/IS (Q, r) policies with ADI have the drawback that they appear to be more complicated than independent IK/ES (Q, r) policies with ADI and most importantly that they cause an indirect coupling between the actions of detaching a kanban and communicating demand. This coupling may cause delays in communicating demand information and therefore not take full advantage of ADI.

3.4.4. Behavior and properties of Synchronized DAS IK/ES(Q, r) policies with ADI

Queuing network model representations of a two-stage production-inventory system operating under a synchronized DAS IK/ES (Q,r) policy with ADI is shown in Figure 21:

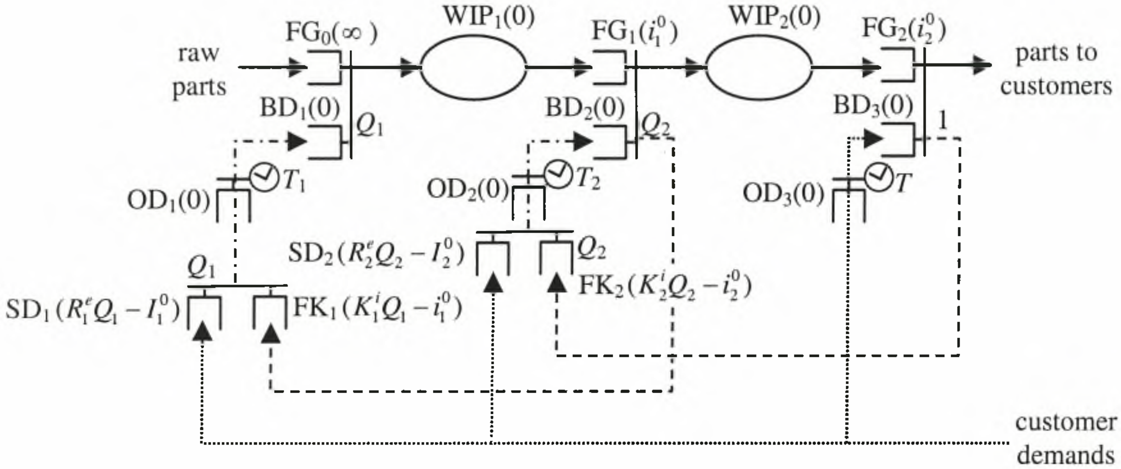

Figure 21: Queuing network model representation of a two-stage production-inventory system operating under a synchronized DAS IK/ES (Q,r) policy with ADI.

The timing of events in an synchronized DAS IK/ES (Q,r) policy with ADI, is shown in Figure22:

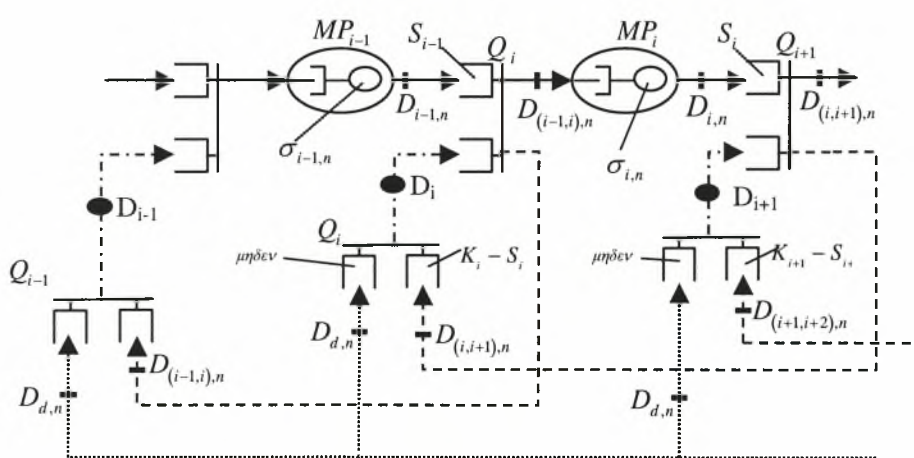

Figure 22: Timing of events in a synchronized DAS IK/ES (Q,r) policy with ADI

As before we will write down the evolution equations in the following proposition:

Proposition 4. *In a independent IK/ES (Q,r) policy, with ADI, in which MP_i consists of a single machine, the timings of events are related by the following evolution equations:*

$$D_{i,n} = \sigma_{i,n} + max\left(D_{i,n-1}, D_{(i-1,i),n}\right), i = 1,...,N \tag{36}$$

$$D_{(i-1,i),n} = max\left(D_i + \max\left(D_{d,\left\lceil \frac{n}{Q_i} \right\rceil Q_i}, D_{(i,i+1),\left\lceil \frac{n}{Q_i} \right\rceil Q_i - (K_i - S_i)}\right), D_{i-1,\left\lceil \frac{n}{Q_i} \right\rceil Q_i - S_{i-1}}\right), i = 2,...,N+1 \tag{37}$$

Proof. Equation (36) can be explained in a similar way as Equation (17) in the EKCS in Dallery's and Liberopoulos (2000). Equation (37) can be explained in a similar way as equation (18) in the EKCS in Dallery's and Liberopoulos (2000).

The symbolism $\left\lceil \frac{n}{Q_i} \right\rceil Q_i$ has to do with the batching of parts. Equation (37) is recursive in that it expresses $D_{(i-1,i),*}$ in terms of $D_{(i,i+1),*}$. Expanding this recursion yields:

i=N+1:

$$D_{(N,N+1),n} = max\left(D_{N+1} + \max\left(D_{d,\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}}, D_{(N+1,N+2),\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}-(K_i-S_i)} \right), D_{N,\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}-S_N} \right) \Rightarrow$$

$$\boldsymbol{D_{(N,N+1),n} = max\left(D_{N+1} + D_{d,\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}}, D_{N,\left\lceil \frac{n}{Q_{N+1}} \right\rceil Q_{N+1}-S_N} \right)}$$

i=N

$$D_{(N-1,N),n} = max\left(D_N + \max\left(D_{d,\left\lceil \frac{n}{Q_N} \right\rceil Q_N}, D_{(N,N+1),\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)} \right), D_{N-1,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-S_{N-1}} \right),$$

$\textit{οπου}$:

$$D_{(N,N+1),\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)} = max\left(D_{N+1} + D_{d,\left\lceil \frac{\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}{Q_{N+1}} \right\rceil Q_{N+1}}, D_{N,\left\lceil \frac{\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}{Q_{N+1}} \right\rceil Q_{N+1}-S_N} \right)$$

$\textit{οπου}$:

$$\left\lceil \frac{\left\lceil \frac{n}{Q_N} \right\rceil Q_N - (K_N - S_N)}{Q_{N+1}} \right\rceil Q_{N+1} = \left\lceil \frac{\left\lceil \frac{n}{Q_N} \right\rceil Q_N - (\nu-\xi) Q_N}{Q_{N+1}} \right\rceil Q_{N+1} = \left\lceil \left\{ \left\lceil \frac{n}{Q_N} \right\rceil - (\nu-\xi) \right\} \frac{Q_N}{Q_{N+1}} \right\rceil Q_{N+1} =$$

$$= \left\lceil \left\{ \left\lceil \frac{n}{Q_N} \right\rceil - (\nu-\xi) \right\} \mu \right\rceil Q_{N+1} = \left\lceil \frac{n}{Q_N} \right\rceil Q_N - (K_N - S_N),$$

$\textit{αρα}$:

$$D_{(N,N+1),\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)} = max\left(D_{N+1} + D_{d,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}, D_{N,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-K_N} \right),$$

$\textit{αρα}$:

$$\boldsymbol{D_{(N-1,N),n} = max\left(D_N + D_{d,\left\lceil \frac{n}{Q_N} \right\rceil Q_N}, D_{N+1} + D_N + D_{d,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-(K_N-S_N)}, D_N + D_{N,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-K_N}, D_{N-1,\left\lceil \frac{n}{Q_N} \right\rceil Q_N-S_{N-1}} \right)}$$

Hence the final form of $D_{(i-1,i),n}$ is:

$$D_{(i-1,i),n} = \max_{j=i-1}^{N} \left\{ \sum_{m=i}^{j} D_m + D_{j, \left\lceil \frac{n}{Q_i} \right\rceil Q_i - S_j - \sum_{m=i}^{j} (K_m - S_m)}, D_{j+1} + \sum_{m=i}^{j} D_m + D_{d, \left\lceil \frac{n}{Q_i} \right\rceil Q_i - \sum_{m=i}^{j} (K_m - S_m)} \right\},$$

and so the final evolution equations have the following form:

$$\left\langle \begin{array}{l} D_{i,n} = \sigma_{i,n} + \max\left(D_{i,n-1}, D_{(i-1,i),n}\right), i = 1, \ldots, N \\ D_{(i-1,i),n} = \max_{j=i-1}^{N} \left\{ \sum_{m=i}^{j} D_m + D_{j, \left\lceil \frac{n}{Q_i} \right\rceil Q_i - S_j - \sum_{m=i}^{j} (K_m - S_m)}, D_{j+1} + \sum_{m=i}^{j} D_m + D_{d, \left\lceil \frac{n}{Q_i} \right\rceil Q_i - \sum_{m=i}^{j} (K_m - S_m)} \right\} \end{array} \right\rangle \qquad (38)$$

A synchronized DAS IK/ES (Q, r) policy with ADI is never nested in the sense that an IS (Q, r) policy with ADI is, i.e., in the sense that when an order is placed at stage *n*, then orders must simultaneously be placed at all downstream stages as well (except when $K_n^i = \infty$, for all *n*, as we will see below).

A synchronized DAS IK/ES (Q, r) policy with ADI includes IK policies and IS (Q, r) policies with ADI as special cases. Specifically, a synchronized DAS IK/ES (Q, r) policy with ADI with $K_n^i = R_n^i$, for all *n* is equivalent to an IK policy, i.e., a policy which does not exploit ADI, as was mentioned earlier. A synchronized DAS IK/ES (Q, r) policy with ADI with $K_n^i = \infty$, for all *n*, is equivalent to an ES (Q, r) policy with ADI. Any other synchronized DAS IK/ES (Q, r) policy with ADI with $R_n^i < K_n^i < \infty$ is never nested, imposes an upper bound on the WIP + FG inventory, while exploiting ADI for better replenishment control; however, it does not take full advantage of ADI, since the communication of demands from a stage *n* to the previous upstream stage *n*-1 may be blocked due to the lack of free stage-*n* kanbans in queue FK_n.

By observing closely the evolution equations of a synchronized DAS IK/ES (Q, r) policy with A.D.I (38), and comparing them with the corresponding evolution equations of an independent IK/ES (Q, r) policy with A.D.I (35), we can come to the conclusion that an independent IK/ES (Q, r) policy with A.D.I is ***"quicker"*** than a synchronized DAS IK/ES (Q, r) policy with A.D.I. This means that if an order arrives simultaneously in both a synchronized DAS IK/ES & an independent IK/ES (Q, r) policy with A.D.I (with identical initial conditions & delay mechanisms), the independent IK/ES (Q, r) policy will respond quicker. This observation does not mean that an independent IK/ES (Q, r) policy with A.D.I is superior to a synchronized DAS IK/ES (Q, r) policy with A.D.I, because we don't have data about the corresponding holding & backorder costs of the two systems.

Chapter 4

Conclusions

4.1. Introduction

We presented a unified framework, which is based on a queuing network modeling representation, for *describing, comparing and contrasting* simple and hybrid multi-stage production-inventory control policies with lot-sizing and advance demand information (ADI). The simple policies that we considered are reorder point and kanban policies. The hybrid policies are combinations of the simple policies, which can be materialized in a synchronized or an independent way, leading to synchronized and independent hybrid policies, respectively.

We then, attempted to describe their basic operations and functions, emphasizing on their advantages and disadvantages. We developed evolution equations, so as to describe in detail the dynamics of each system. Using the above analysis we attempted a comparison of the above systems, emphasizing in equivalencies and superiorities.

The basic findings of this effort is summarized in the next section

4.2. Conclusions

A. Let us first consider the case where ADI isn't available:

A1. We have first considered IS & ES (Q, r) control policies. If an ES (Q, r) policy is nested, i.e. condition (10) holds, it can be replaced by an equivalent IS (Q, r) policy. This means that ES (Q, r) policies are superior to IS (Q, r) policies. In the limiting case where $Q_n=1$, n=1,2,…, the two policies are identical and form the well know *base stock control policy*

A2. IK and EK policies are never equivalent to each other, except in the trivial case where there is a single stage. For this reason, it is not simple to determine whether EK policies are superior to IS policies or vice versa. It should be noted, however, that an advantage of EK policies over IK policies is that the former polices use global information, whereas the latter policies use only local information

A3. Kanban policies have an important advantage over their reorder point counterpart policies is that the former policies impose an upper bound on the WIP + FG inventory. This advantage implies inventory holding cost savings

A4. Kanban policies do not communicate customer demand information to all upstream stages as quickly as their corresponding reorder point policies. This fact has a direct impact on customer service since it implies longer customer response times, particularly if customer demand is highly variable. It also implies that the capacity of the system depends on the number of kanbans

A5. We can not say with certainty whether an independent IK/ES (Q, r) policy is superior to a synchronized IK/IS (Q, r) policy or vice versa. It would not be surprising, however, if in many cases an independent IK/ES (Q, r) policy turned out to perform better than a synchronized IK/IS (Q, r) policy because the former policy uses global information, whereas the latter policy uses only local information

A6. Synchronized IK/IS (Q, r) policies have the drawback that they appear to be more complicated than independent IKES (Q, r) policies

A7. Synchronized IK/IS (Q, r) policies cause an indirect coupling between the actions of detaching a kanban and communicating demand. This coupling may cause delays in communicating demand information

A8. In an independent IK/ES (Q, r) policy, demands are satisfied earlier than in the synchronized IK/IS (Q, r) policy, but it does not necessarily also mean that the independent IK/ES (Q, r) policy has an overall better performance than the synchronized IK/IS (Q, r) policy, since inventory storage costs are not taken into account. In fact, the independent IK/ES (Q, r) policy is likely to incur higher inventory storage costs than does the synchronized IK/IS (Q, r) policy

A9. The production capacity of an independent IK/ES (Q, r) policy is higher than the production capacity of the synchronized IK/IS (q, r) policy, with the same parameters

B. We then considered the case where ADI is available:

B1. IS (Q, r) policies with ADI are nested ES (Q, r) policies with ADI and are therefore special cases of the latter policies

B2. An MRP system with fixed order quantity as its lot sizing rule is equivalent to an IS (Q, r) policy with ADI. Given that an IS (Q, r) policy with ADI is a special case of an ES (Q, r) policy with ADI, this means that an MRP system is a special case of an ES (Q, r) policy with ADI.

B3. An IS (Q, r) policy with ADI & unit lot sizes is identical to an ES (Q, r) policy with ADI

& unit lot sizes, and they are both equivalent to a *base stock policy with a release time parameter*

B4. Installation & Echelon kanban policies, by their nature, can not take advantage of ADI

B5. The timing of events in a synchronized DAS IK/IS (Q, r) policy with ADI, is much too complicated and in general is a very complex policy and hence difficult to implement

B6. A synchronized DAS IK/IS (Q, r) policy with ADI cause an indirect coupling between the actions of detaching a kanban and communicating demand. This coupling may cause delays in communicating demand information and therefore not take full advantage of ADI.

B7. An Independent IK/ES (Q, r) policy with A.D.I is superior to an Independent IK/IS (Q, r) policy with A.D.I.

B8. We cannot say with certainty whether an independent IK/ES (Q, r) policy with ADI performs better or worse than a synchronized DAS IK/IS (Q, r) policy with ADI. It would not be surprising, however, if in many cases an independent IK/ES (Q, r) policy with ADI turned out to perform no worse than a synchronized DAS IK/IS (Q, r) policy with ADI because the former policy uses global information, whereas the latter policy uses only local information

B9. In an independent IK/ES (Q, r) policy with A.D.I, demands are satisfied earlier than in the synchronized DAS IK/ES (Q, r) policy with A.D.I. This fact does not necessarily mean that the independent IK/ES (Q, r) policy with A.D.I has an overall better performance than the synchronized DAS IK/ES (Q, r) policy with A.D.I, since inventory storage costs are not taken into account.

References

1. AXSATER, S. and ROSLING, K., 1993, Installation vs. ES policies for multilevel inventory control. *Management Science*, 39 (10), 1274-1280.
2. AXSATER, S. and ROSLING, K., 1994, Multilevel production-inventory control: Material requirements planning or reorder point policies? *European Journal of Operations Research*, 75, 405-412.
3. AXSATER, S. and ROSLING, K., 1999, Ranking of generalized multi-stage KANBAN policies. *European Journal of Operations Research*, 113, 560-567.
4. AXSATER, S. (2000) *Inventory Control*, Kluwer: International Series in Operations Research and Management Science, Norwell, MA.
5. BEDWORTH, D.D. and BAILEY, J.E. (1987) *Integrated Production Control Systems*, Wiley, New York, NY.
6. BENTON, W.C. and SHIN, H. (1998) Manufacturing planning and control: The evolution of MRP and JIT integration, *European Journal of Operations Research*, 110, 411-440.
7. BUZACOTT, J. A., 1989, Queuing models of kanban and MRP controlled production systems. *Engineering Costs and Production Economics*, 17, 3-20.
8. BUZACOTT, J. A. and SHANTHIKUMAR J. G., 1993, *Stochastic Models of Manufacturing Systems* (Englewood Cliffs, NJ: Prentice-Hall).
9. BUZACOTT, J.A. (1989) Queueing models of kanban and MRP controlled production systems, *Engineering Costs and Production Economics*, 17, 3-20.
10. BUZACOTT, J.A. and SHANTHIKUMAR J.G. (1994) Safety stock versus safety time in MRP controlled production systems, *Management Science*, 40 (12), 1678-1689.
11. CACHON, G.P. and ZIPKIN, P.H. (1999) Competitive and cooperative inventory policies in a two-stage supply chain, *Management Science*, 45 (7), 936-953.
12. CHEN, F.(1998) Echelon reorder points, installation reorder points, and value of centralized demand information, *Management Science*, 44 (12), 592-602.
13. CHEN, F., DREZNER, Z., RYAN, J.K. and SIMCHI-LEVI, D. (2000) Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times and information, *Management Science*, 46 (3), 436-443.
14. DALLERY, Y. and LIBEROPOULOS, G., 2000, Extended kanban control system: Combining kanban and base stock. *IIE Transactions*, 32 (4), 369-386.
15. DURI, C., FREIN, Y., and DI MASCOLO, M. (2000) Comparison among three pull control policies: kanban, base stock and generalized kanban, *Annals of Operations Research*, 93, 41-

69.

16. ELSAYED, E.A. and BOUCHER, T.O. (1994) *Analysis and Control of Production Systems*, Prentice Hall, Englewood Cliffs, NJ.

17. FEDERGRUEN, A. (1993) Centralized planning models for multi-echelon inventory systems under uncertainty, in *Logistics of Production and Inventory*, S.C. Graves *et al.* (eds.), North-Holland: Handbooks in OR & MS, Vol. 4, 133-173, Amsterdam.

18. HARDLEY, G. and WHITIN, T. (1963) *Analysis of Inventory Systems*, Prentice Hall, Englewood Cliffs, NJ.

19. HOPP, W.J. and SPEARMAN, M.L. (1996) *Factory Physics*, McGraw Hill, Boston, MA.

20. KARAESMEN, F., BUZACOTT, J. A., and DALLERY, Y., 2002, Integrating advance order information in make-to-stock production systems. *IIE Transactions*, 34 (8), 649-662.

21. KARAESMEN, F. and DALLERY, Y. (2000) A performance comparison of pull control mechanisms for multi-stage manufacturing systems, *International Journal of production Economics*, 68, 59-71.

22. KARAESMEN, F., LIBEROPOULOS, G., and DALLERY, Y. (2001) The value of advance information on demands for make-to-stock manufacturing systems, working paper, Laboratoire Productique-Logistique, Ecole Centrale de Paris.

23. KARMARKAR, U. (1989) Getting control of just-in-time, *Harvard Business Review*, 67, September-October, 122-131.

24. LEE, H.L., PADMANABHAN, V. and WHANG, S. (1997) Information distortion in a supply chain: The bullwhip effect, *Management Science*, 43 (4), 546-558.

25. LIBEROPOULOS, G. and KOUKOUMIALOS, S. (2001) Numerical investigation of tradeoffs between base stock levels, numbers of kanbans and production lead times in production-inventory supply chains with advance demand information, working paper, Production Management Laboratory, Department of Mechanical & Industrial Engineering, University of Thessaly.

26. LIBEROPOULOS G. and DALLERY, Y., 2000, A unified framework for pull control mechanisms in multi-stage manufacturing systems. *Annals of Operations Research*, 93, 325-355.

27. LIBEROPOULOS G. and DALLERY, Y., 2002, Queuing network representation of multi-stage production-inventory policies with lot-sizing. *International Journal of Production Research*, (submitted).

28. LIBEROPOULOS G. and DALLERY, Y. (2001) Base stock versus WIP cap in single-stage make-to-stock production-inventory systems, *IIE Transactions* (to appear).

29. PYKE, D.F. and COHEN, M.A. (1990) Push and pull in manufacturing and distribution systems, *Journal of Operations Management*, 9 (1), 24-43.

30. RICE, J.W. AND YOSHIKAWA, T. (1982) A comparison of Kanban and MRP concepts for the control of repetitive manufacturing systems, *Production and Inventory Management Journal*, First Quarter, 1-13.

31. SILVER, E.A., PYKE, D.F. and PETERSON, R. (1998) *Inventory Management and Production Planning and Scheduling*, Wiley, New York, NY.

32. SIMCHI-LEVI, D., KAMINSKY, P., and SIMCHI-LEVI, E. (2000) *Designing and Managing the Supply Chain*, McGraw-Hill: Business Management & Organization Series, Boston, MA.

33. SIPPER, D. and BULFIN, JR. R.L. (1997) *Production: Planning, Control, and Integration*, MgGraw Hill, New York, NY.

34. SPEARMAN, M.L., WOODRUFF, D.L. and HOPP W.J. (1990) CONWIP: A pull alternative to kanban, *International Journal of Production Research*, 28, 879-894.

35. SPEARMAN, M.L. and ZAZANIS, M.A. (1992) Push and pull production systems: issues and comparisons, *Operations Research*, 40 (3), 521-532.

36. VEACH, M.H. and WEIN, L.M. (1994) Optimal control of a two-station tandem production-inventory system, *Operations Research*, 42 (2), 337-350.

37. VEINOTT, A., JR., (1965) The optimal inventory policy for batch ordering, *Operations Research*, 13, 424-432.

38. VOLLMANN, T.E., BERRY, W.L. AND WHYBARK, D.C. (1997). *Manufacturing Planning and Control Systems*, McGraw Hill, Boston, MA.

39. ZIPKIN, P., 1989, A kanban-like production control system: analysis of simple models. Research Working Paper No. 89-1, Graduate School of Business, Columbia University, New York, USA.

40. ZIPKIN, P. (2000) *Foundations of Inventory Management*, McGraw Hill: Management & Organization Series, Boston, MA.