A SYSTEM DYNAMICS PERSPECTIVE ON JIT-KANBAN

by Ramon O'Callaghan

(doctoral candidate)

Anderson 25
Harvard Business School
Boston, Mass. 02163

Paper presented at the

1986 International Conference of the System Dynamics Society

A SYSTEM DYNAMICS PERSPECTIVE ON JIT-KANBAN

by Ramon O'Callaghan

Abstract

Just-In-Time (JIT) production is the notion of producing the necessary products in the necessary quantities in the necessary time in every process of a factory and also among companies. It is not uncommon to find JIT used synonymously with "Kanban," which is the name for a specific inventory replenishment system developed by Toyota to accomplish JIT production. The Kanban system employs cards (kanbans) to signal both the need to deliver more parts and the need to produce more parts. A unique feature that distinguishes the kanban-based JIT system is its unique "pull" nature.

The paper begins with a review of JIT production and the Kanban system. Then, using the structuring principles of System Dynamics, a simulation model of a kanban-based JIT production system is developed. The formulation effort begins with the "simple structure" of one production stage. By connecting a few of these "basic structures" and adding a market interface module, a complete multi-stage manufacturing system is developed later.

To test the internal consistency of the model, several simulation experiments are conducted. The unifying theme in these experiments is the issue of flexibility: How well does the system adapt to changes. The simulations are thus designed to show, for different management policies, the behavior of the system in response to unexpected circumstances. The following cases are considered: normal response, changing the number of kanbans, a breakdown, small and large demand increases, bottlenecks, and capacity planning. Finally, the results of these simulations are used to point out some of the managerial trade-offs involved in JIT production.

Although the major contribution is the conceptualization and formulation of the system dynamics model, the paper lays the groundwork for subsequent normative research in the field of operations management.

BACKGROUND ON JUST-IN-TIME PRODUCTION AND THE KANBAN SYSTEM

Just-in-Time production is the notion of producing the necessary products in the necessary quantities in the necessary time in every process of a factory and also among companies.

In ordinary production control systems, various production schedules are issued to all the processes. Parts and assemblies are produced according to these schedules, employing the method of the preceding process supplying materials and parts to its following process (see figure 1). This is known as a PUSH system. This method makes it difficult to promptly adapt to changes caused by demand fluctuations. For adapting to these fluctuations, the company must change each production schedule for each process simultaneously, and this approach makes it difficult to change the schedules frequently. As a result the company must hold inventory among processes to absorb troubles and demand changes.

By contrast, Just-in-Time (JIT) is a PULL system, in the sense that the subsequent process withdraws parts from the preceding process. Only the final assembly line knows accurately the necessary timing and quantity of parts required. The final assembly line goes to the preceding process to obtain the necessary quantity at the necessary time for assembly (fig. 2). Thus, the preceding process produces the parts withdrawn by the subsequent process. The procedure is repeated further down the line. The beauty of this method is that no production schedules need to be issued simultaneously to all processes (at least in the short run). Additionally, inventory levels are indeed quite low because nothing is produced that has not been requested (by the subsequent process) for immediate use. (Hall, 1983).

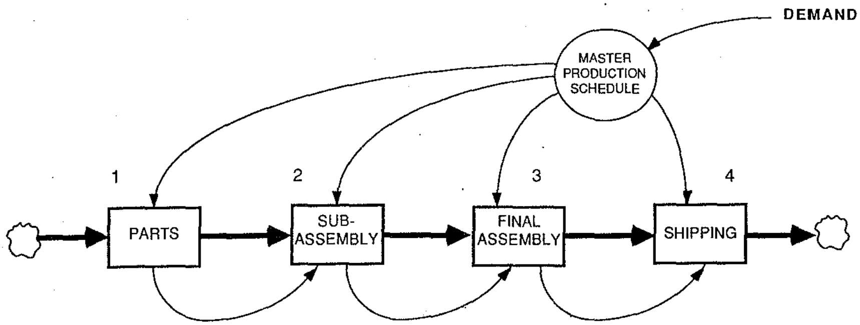
The Kanban system

"Kanban" (pronounced kahn-bahn), literally translated, means "visible record" or "visible plate." More generally, kanban is taken to mean "card."

The Kanban system can be viewed as an information system that controls Just-in-Time production.

The Kanban System employs cards ("kanbans") to signal the need to deliver more parts, and the need to produce more parts. Kanbans are attached to units or containers holding a given number of units. When a unit (or container) is used up, its associated kanban is detached. Then, the detached kanban becomes an order

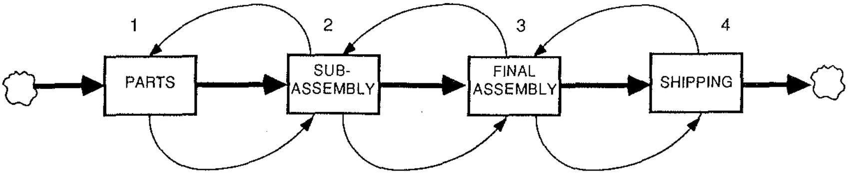
A PUSH SYSTEM : MRP



- TRANSMISSION OF INFORMATION
  - FORWARD : BY AVAILABILITY OF INVENTORIES
  - NO BACKWARD INFORMATION
  - SIMULTANEOUS INFO. FROM **M.P.S.**

Figure - 1

A PULL SYSTEM : JIT - KANBAN



- TRANSMISSION OF INFORMATION
    - FORWARD : BY AVAILABILITY OF INVENTORIES
    - BACKWARD : BY KANBAN (PRODUCTION ORDERS)

Figure - 2

requesting a new unit.

Two kinds of Kanbans are used: 1) WITHDRAWAL Kanban that specifies the kind and quantity of product to be withdrawn from the subsequent process, and 2) PRODUCTION Kanban that specifies the kind and quantity of product to be produced. See appendix A. The withdrawal Kanban is used as an INTER-PROCESS signal: to move physical units from one process to another process that takes place at a different location; whereas the production Kanban is an INTRA-PROCESS signal: to issue production orders in a particular process.

As result of this dual chain of kanbans, the rate of production of the succeeding process is transmitted to the preceding process and every process receives the necessary units at the necessary time in the necessary quantities. The Just-in-Time ideal is realized at each process.

CONDITIONS FOR THE KANBAN SYSTEM

Smoothing of production:

The JIT ideal is to make one piece just in time for the next operation. The kanban system is the information system that carries out this ideal by ensuring that the preceding processes continuously produce their product in the quantities withdrawn by the subsequent processes. Since the subsequent processes will require a single unit or a small lot size, the preceding processes must make frequent setups according to the frequent requisitions by the subsequent processes. The Japanese use engineering to cut machine set-up times so that it is economical to run very small batches.

Traditionally, the Economic Order Quantity (EOQ) concept has been used as the "optimal" lot size. The EOQ is a compromise between inventory carrying costs and set-up costs. But, as the Japanese experience demonstrates, these are only the obvious costs. The Japanese have found that producing and carrying smaller lots results in many benefits other than savings on inventory carrying costs. The main benefits are in quality, worker motivation and productivity (Schonberger, 1982).

For a small-lot withdrawal and a small-lot production, the smoothing of production (leveled daily production) is a necessary pre-requisite. The Kanban system itself is merely a dispatching means for actual production actions during each day at each process. Before entering the phase of dispatching the jobs by Kanban, overall planning throughout the plan must be made in advance. Toyota, for example, informs each process and each supplier each month of a predetermined monthly production quantity for the next month's production so each process and each supplier can prepare in advance its cycle time, necessary workforce, necessary number of materials, etc. (Monden, 1983). Based on such overall plans, all processes in the plant can start Just-in-Time production (according to the new schedule) the first day of the month.

Production planning for smooth production:

The objective of production planning in JIT is to prepare to execute a level schedule. A level schedule is one that has as even a distribution as possible of material requirements as well as labor requirements.

Prior to JIT production, most companies operated on either a monthly ordering system, or on the basis of MRP. With monthly ordering, parts schedules are based on forecasts, many of them independently made for each part number. With MRP they are most generally made by back scheduling due dates for parts based on an explosion of requirements from the Master Production Schedule. If production is repetitive, these schedules are normally converted into daily schedules for the plant floor. (fig. 1). (Morecroft, 1983).

With JIT production, planning the final assembly schedule is critical. The final assembly schedule is the key that triggers the whole system. JIT production requires development of the ability to synchronize everything from the final assembly schedule. All other schedules are only in preparation for this. Except for final assembly, actual production is executed in response to a pull signal, not a schedule.

The pre-planning of final assembly schedules may start a few months before the final assembly schedules (actual runs) are given daily to the lines. Planning is approximate in the early stages. A "master production schedule" (MPS) is developed as a summary in daily buckets of "expected" final assembly schedules. The preplanning (the MPS) is done based on a forecast, and the closer the plan comes to the time of execution, the more it is revised, based on a combination of forecast and actual demand. The final assembly schedules which will really drive the pull system are the last revisions, and these may be developed as little as one day before they are run.

The initial stage of final assembly scheduling is to fix the line rates. If final assembly lines are to be balanced and fabrication balanced to run with them, initially fixing the overall line rate is important to allow everyone to plan ahead for the most significant changes in equipment configuration and manning. If equipment is flexible and workers are cross-trained to work at several different positions with skill, the possibility of a plant rebalancing itself to operate at a different rate is increased, but that change cannot be made frequently. What is most often done in Japan is to adjust the length of the planned work day, usually by overtime.

In summary, these are the main points in planning production for a JIT system:

1 - The overall rate of production and an approximate model mix are fixed so that everyone can prepare in advance.

2 - Within the confines of the overall fixed rate, total output and model mix can be adjusted, within limits, from that which was first planned.

"Freezing" the production schedule:

The length of time required for production planning depends on how much time is required physically and organizationally for preparation. During this planning period, the current production schedules are held "frozen" (i.e. they are not changed). The amount of time over which the overall production rate is "frozen" ranges from 5 to 25 working days. The fabrication schedules are developed by exploding the master production schedule using the bills of materials. Since the production schedule is "frozen," what results is a set of fabrication schedules: one for each identical day in the planning interval (a 15 day period, for example). Again, the purpose of these schedules is to allow fabrication, subassembly, and supplier supervisors to have advance warning about the scope of the schedules to be run. This allows pre-planning of the workplace, manning, and tooling organization required to balance the operations, move the material, and perform preventive maintenance. The idea is that advanced planning for JIT should provide for capacity in excess of what is required well in advance. This idea is captured in figure 3.

Note however, that the planned fabrication schedule only ADVISES departments of the impact of the PLANNED final assembly schedule on each product. ACTUAL fabrication takes place only IN RESPONSE to the PULL system coming from final assembly.

Coping with demand changes:

In general, the number of Kanbans is kept constant. Therefore, when daily demand increases, the lead time must be reduced. This requires reducing the time of standard operations by changing the allocation of workers in the line. However, a workshop incapable

THE " MARRIAGE " OF MRP AND JIT

(YAMAHA' S SYNCHRO)

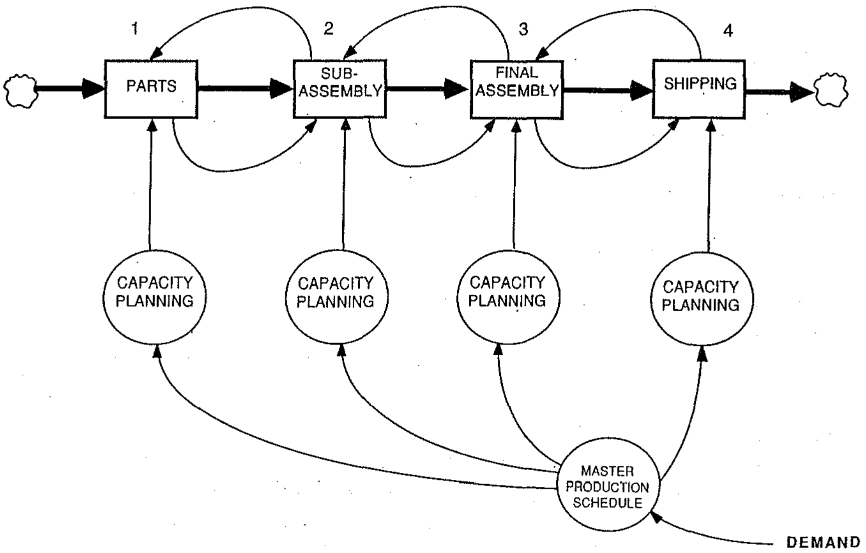


Figure - 3

of such improvements will suffer line-stops or force the use of overtime. Shops may increase the number of Kanbans to adapt to demand increase.

To illustrate this adaptability (known as "volume flexibility"), it is worth examining what happens in companies not using Kanban. These companies lack the means to deal smoothly with sudden, unexpected demand changes. The ordinary control system centrally revises the current production schedules, determines the new production schedules and issues them simultaneously to all production processes. Typically, this is a task that requires seven to ten days. As a result, the various processes are faced, from time to time, with abrupt changes in production requirements.

Instead, companies using the Kanban system do not issue detailed production schedules simultaneously to the preceding processes during a month; each process can only know what to produce when the production-ordering Kanban is detached from the container at its store. It is only the final assembly line that receives a schedule for a day's production. The way the Kanban system achieves volume flexibility is by means of "fine-tuned production". That is: the system is able to produce a few more units than the number predetermined by schedule without actually revising the schedule. Such fine-tuning of production by Kanban can only adapt to small fluctuations in demand. According to Toyota, demand variations of 10% can be handled without revising the schedule and without changing the number of Kanban (that is without increasing the inventory levels).

However, in case of larger seasonal changes in demand, or in the case of an increase or decrease of the actual monthly demand over the predetermined load or the preceding month's load, all the production lines must be rearranged. The cycle time of each workstation and the number of workers in each process must be changed. Otherwise, the number of Kanbans must be increased or decreased (allowing more or less inventory in the system).

A SYSTEM DYNAMICS MODEL OF JIT-KANBAN

The model presented in this paper is a multi-stage manufacturing system. It consists of a transfer line including the following processes: 1) parts procurement, 2) sub-assembly, 3) final assembly, and 4) shipments.

This structure is shown in figure 2.

The development of such a relatively large model begins with the understanding and formulation of its integral components. We will start with the formulation of one "basic structure." The model of the "basic structure" will be that of an individual stage (production process) within the chain of processes that constitutes the entire manufacturing system. Later, we will develop the complete multi-stage model by chaining several of these "basic structures" .

The "basic structure"

In the "basic structure," a workstation stands in between two inventories: the inbound stock and the outbound stock. The inbound stock consists of parts or assemblies that are inputs to the production process. The outbound stock results from the accumulation of the assemblies or products that are manufactured by the workstation (its output). Thus, the production process at a given workstation depletes its inbound inventory and increases its outbound inventory. As we have seen, the production Kanbans regulate this process.

When consequent workstations are located apart from each other, the output of one station has to be physically transported to its succeeding station (where it becomes the input to the next production process). This transportation process takes time and holds some inventory ("in-transit" inventory). In this transportation process, "In-transit" inventory results from the "transportation lead time" (delay due to the time to move a unit), and a "withdrawal" kanban is used to trigger the moving of units. This is analogous to the production process where "In-process" inventory results from the "production lead time" (delay due to the time to complete a unit), and a "production" kanban is used for triggering production. The analogy is so strong that most modelers of kanban systems do not make any distinction and consider transportation as another process that is interleaved in the manufacturing chain. (Kimura and Terada, 1981).

Given that the transportation process does not add anything conceptually new to the "basic structure", it will be assumed that the workstations in our model are so close to each other that no physical transportation is required (the effect of transportation is neglected, and the "withdrawal kanbans" are not used). Our discussion will thus revolve around the "production" kanbans only.

Each time a unit is withdrawn from inventory, its kanban is detached and placed in the collection box (See Appendix A). Periodically (every few hours) the detached kanbans that have accumulated in the box are taken to the dispatching post, where they become production orders. The time interval at which the kanbans are taken from the collection box and moved to dispatching is fixed. This interval is called the "Kanban cycle." The kanbans in the collection box become production orders when they are dispatched in the next time interval (kanban cycle).

It is important to realize that the "kanban cycle" determines how fast the system will react to changes in production rate. Using the system dynamics terminology, the "kanban cycle" is a "time constant." In the real world this parameter will be determined by different factors. In the case of suppliers, it is established by a contract whereby the supplier agrees to deliver once or twice a day, for example. In the case of a production process, it will be determined by the flexibility to transfer workers to the workstation when required, by the policies on the use of overtime, and especially, by the production lot size (which in turn will be determined by the set-up times).

From the previous section that has introduced JIT-kanban, we can summarize the essential points determining the structure and operation of the Kanban system:

- Production is not instantaneous: it takes time to produce a unit. That means there will be a production "lead time", and, consequently, some "work-in-process" inventory at each manufacturing stage.

- The number of Kanbans determines the maximum inventory. In fact, any Kanban has to be either attached to a container (in Work-in-process inventory, or in output inventory), or in the Kanban receiving box, or in the dispatching post.

- The kanbans in the receiving box (that have been detached by the succeeding process each time a container has been used), will eventually get to dispatching. So, all the detached kanbans that have not yet joined the production process, constitute a BACKLOG of PRODUCTION ORDERS. The size of this backlog is determined by the kanban cycle (in equilibrium the number of kanbans in this backlog is the number of kanbans detached during a kanban cycle, that is = normal production rate x kanban cycle). See figure 4.

- This backlog of production orders determines the DESIRED production rate (in units/day). I say "desired" because the ACTUAL production rate will also depend on other factors like: availability of inventory from the preceding stage, availability of enough workers, or constraints on overtime.

- The total number of Kanbans is given (a system's parameter) and equals the sum of the number of "work-in-process" containers plus all containers in "output" inventory (units completed) plus backlog of production orders.

THE KANBAN SYSTEM

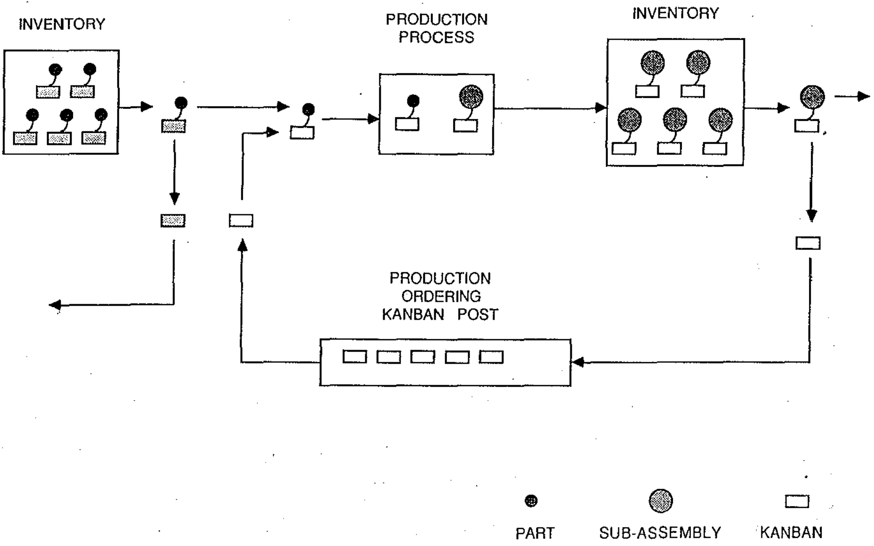


Figure - 4

Formulation of the "basic structure"

With the preceding points in mind, let's now move to the stock and flow diagram which is shown in the figure 5.

The figure shows the backlog of kanbans that have been detached (prod. orders, PO2). The inflow rate of this production orders backlog is controlled by the rate at which units are withdrawn from inventory (I2), which is the production rate of the succeeding stage (PR3). The outflow rate is controlled by the actual production rate of the current stage (PR2).

All production kanbans accumulated in the production orders backlog (PO2) are supposed to be dispatched during the time interval of the Kanban cycle (TI2). Thus, the desired production rate (DPR2) is determined by the production orders backlog (PO2) and the Kanban cycle (TI2).

But actual product rate is not always the "desired" production rate. There are some constraints that prevent production rate (PR2) from being the "desired" production rate (DPR2) at all times. Figure 5 also shows those constraints:

1) availability of inventory from the preceding stage (I1): we cannot produce if do not have enough inventory at the input. Instead of using a step function (that would abruptly change from zero to one, or one to zero) to model the effect of the availability of inventory from the preceding stage, I have preferred to use a continuous function (SW2). A reason for doing so is to avoid the technical problems associated with a discontinuity. Another reason is that the inventory in our model may in fact represent the aggregate inventories of different parts, assemblies or products (and not one single product). The global effect can thus be seen as being more gradual.

2) production capacity ("Maximum production rate" MPR2) (i.e. number of workers, or maximum overtime allowed, etc...). A way to formally model capacity is in terms of the number of workers assigned to the station, their productivity (output/hour), and the maximum overtime allowed. For the sake of generality, however, production capacity in our model will be expressed in the form of "maximum production rate." That is: without especifying the source that limits capacity.

Finally, there is the product completion rate (PCR2) which is the production rate delayed by the production lead time LT2 (the time to produce a unit) and the work-in-process inventory (WIP2).

A slight variation of the "basic structure" of the kanban system is presented in figure 6. Here, the backlog of kanbans

KANBAN SYSTEM : BASIC STRUCTURE

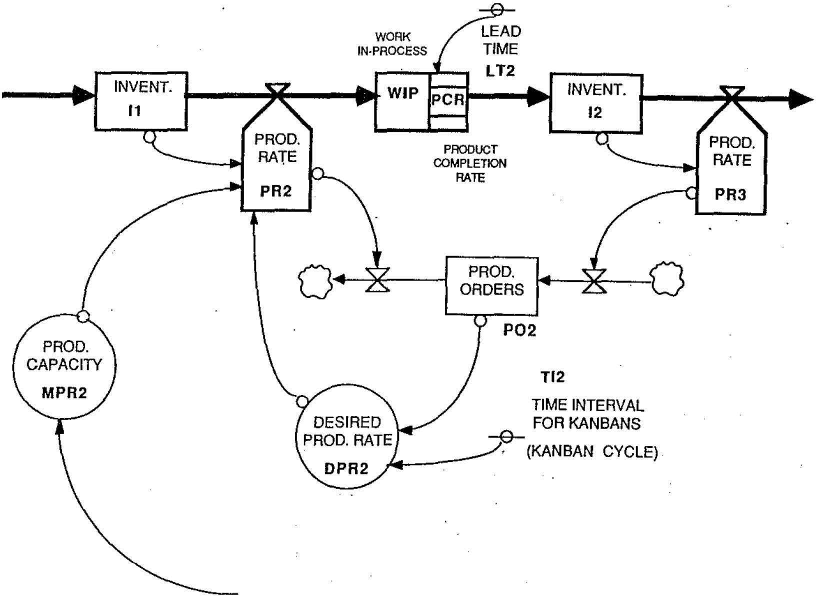


Figure - 5

representing production orders has been replaced by an information structure whose central point is the auxiliary variable PO2 (production orders). In this new structure, the production rate is expressed as a function of the total number of kanbans (maximum inventory allowed), and the existing inventories (work-in-process and finished units). This means using equation (I) instead of equation (II) (appendix B).

Recall that, at all times, a kanban is either in the backlog of production orders, or in the work in-process (WIP) inventory, or in the inventory of finished units. The sum of kanbans in inventory, kanbans in WIP, and kanbans in backlog is constant. This sum is the total number of kanbans for the production process in question. Since the total number of kanbans is kept constant in practice, we can always express the kanbans in the backlog as a function of the kanbans in WIP, and inventory.

Prod. orders = Total # of kanbans - # of kanbans attached to WIP
- # of kanbans attached to finished units inventory

As commented earlier, the number of kanbans determines the maximum inventory in the system. This is illustrated with a simple example. Let's imagine that the succeeding stage does not withdraw any unit from inventory. Then no new kanbans will go into the production orders backlog. WIP will eventually be completed and accumulated into the inventory of finished units. Then, the few orders still in backlog will go to WIP, and finally into inventory of finished units. In the final state, backlog will be zero, WIP will also be zero, there will be no production and all kanbans will be in the inventory of completed units.

Summarizing, the essence of the kanban system can be explained by this basic structure which is no more than a "GOAL SEEKING" FEEDBACK LOOP, with the GOAL being the MAXIMUM INVENTORY ALLOWED in the system (which is entirely determined by the number of kanbans).

It is worth mentioning at this point the subtlety of the kanbans: their double mission. Individually taken, each kanban is a signaling device that triggers production (a kanban becomes a production order). But taken together, all kanbans constitute the goal of the basic feedback loop: the maximum inventory allowed. This will be illustrated later with some simulation experiments. It is important to realize that the two missions are simultaneous. If there is a small number of kanbans (a tight inventory policy) there is little flexibility for reacting to desired changes in production rate (in case of an increase of demand for example, there may not be enough kanbans to generate the number of production orders per kanban cycle required to get a higher production rate). If you want more flexibility you need more kanbans to have the necessary slack in terms of production orders.

KANBAN SYSTEM : BASIC STRUCTURE

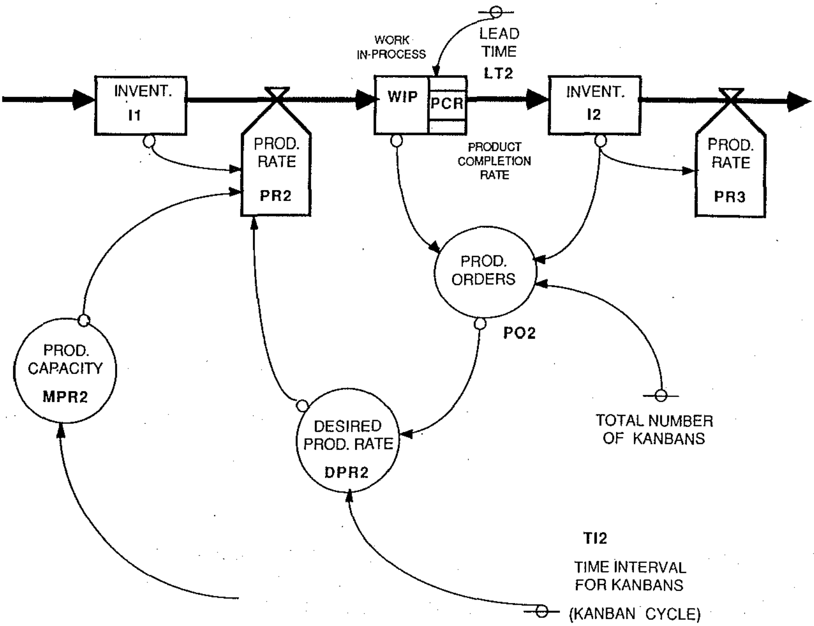


Figure - 6

But then, of course, you have to allow higher levels of inventory.

This flexibility-versus-inventory trade-off stems from the very essence of the kanban system.

A Dynamo version of the "basic structure"

As an example of a possible implementation, the Dynamo equations of one such "basic structure" are included here.

A word about the notation. In this model, Stage 1 represents the preceding process (parts), stage 2 represents the current process (sub-assembly), and stage 3 represents the succeeding process (final assembly). The subscripts (1,2, or 3) in the equations indicate the stage each variable refers to.

```
*
*
*    STAGE (2) :  Sub-Assembly
*
*  Stocks
*
L     I2.K=I2.J+(DT)(PCR2.JK-PR3.JK)       Inventory of finished units
N     I2=NK2*CC2-(ND*LT2)-(ND*TI2)
L     WIP2.K=WIP2.J+(DT)(PR2.JK-PCR2.JK)   Work-in-Process inventory
N     WIP2=ND*LT2
*
*  flows
*
R     PCR2.KL=DELAY3(PR2.JK,LT2)       Product completion delay
C     LT2=.5  DAYS                       due to production lead time.
R     PR2.KL=CLIP(MPR2.K,DPR2.K,DPR2.K,MPR2.K)*SW2.K
A     MPR2.K=1.1*PP.K            max. prod. (capacity) = 10% above prod. plan
A     SW2.K=TABLE(TSW2,CI1.K,0,10,1)     effect of inventory of preced. stage
T     TSW2=0/.85/.95/1/1/1/1/1/1/1/1
A     CI1.K=I1.K/CC1
C     CC1=10   UNITS/CONTAINER
*
*  Information feedback and policies
*
A     PO2.K=NK2.K*CC2-I2.K-WIP2.K    prod. orders (as a function of
                                      Number of Kanbans, WIP, and Inventory)
A     DPR2.K=PO2.K/TI2
C     TI2=0.5 DAY
```

where PR = production rate
PCR = product completion rate
I = inventory

NK = number of Kanbans
CC = container capacity
LT = lead time (production)
TI = time interval (Kanban cycle)
WIP = work-in-process inventory
ND = normal demand
MPR = maximum production rate
DPR = desired production rate
SW = effect of inventory availability (from preceding stage)
PP = production plan
PO = production orders

Although the model documentation is meant to be self-explanatory, the following points are worth further comments.

In equilibrium, production rate (PR) and product completion rate (PCR) are constant and equal to "normal demad" (ND). WIP inventory is also constant, and it is equal to the amount of units kept in process during the production lead time (=ND*LT2). The backlog of production orders is also constant, and it equals the usage (ND) during the kanban cycle (number of kanbans detached per cycle) (=ND*TI2). The level of finished-units inventory is determined by the total number of kanbans minus the number of kanbans attached to WIP, minus the number of kanbans in the backlog of production orders. Therefore, the initial value of finished-units inventory is unequivocally determined by the expression:

```
N   I2 = NK2*CC2 - (ND*LT2) - (ND*TI2)
```

Production rate (PR2) is perhaps the most complex equation. A special Dynamo function, CLIP, is used to mean the following: if "Desired production rate" (DPR2) is lower than the "Maximum production rate" (MPR2), then let "Production rate" be the "Desired production rate." Otherwise, let "Production rate" be the "Maximum production rate." This how the effect of a capacity constraint is introduced. But, there is another effect to be taken into account: the availability of inventory from the preceding stage (SW2). This effect is introduced as a multiplicative effect on production rate. The auxiliary variable SW2 acts as a switch: when the preceding stage has enough inventory (I1) its value is one. When the level of inventory approaches zero, SW2 gets values lower than one, and eventually becomes zero. This has been implemented by means of a table function.

Finally, the number of product orders to be dispatched (PO2) is determined by the number of kanbans that are not attached to physical units (or containers). This variable might be expressed as a backlog: a level whose inflow would be the usage of finished units by the succeding stage, and whose outflow would be production rate at the current stage. However, this is not how it is has been expressed in our model. PO2 has been written, instead, as the difference between the "maximum inventory allowed" (as given by the total number of kanbans, NKK2*CC2) and the existing inventories (sum of WIP and finished units).

PO2.K = NK2.K*CC2 - I2.K - WIP2.K

This has the advantage of reflecting the goal-seeking nature of the information feedback loop: the short-fall between actual inventory and "maximum inventory allowed" signals the need for more production. Also, having the total number of kanbans (NK2) included explicitly in the expression for PO2 is quite useful, because it allows us to treat this parameter (NK2.K) as another variable (whose value may be changed in the course of a simulation run). This will be helpful later when we will conduct several experiments with the model.

The complete multi-stage model:

A multi-stage model is developed by connecting a series of "basic structures" like the one described above. Our model of a complete manufacturing system is a simple three stage transfer line that includes parts, sub-assembly, final assembly. The model also incorporates a market-interface module to take care of shipments and customer orders backlog. Figure 7 shows one such stage at the end of the manufacturing line.

The structure contains a typical customer order backlog that determines "desired" shipping rate given a "normal delivery delay" (the days of backlog is assumed to be a company's policy). Shipping a unit depletes both the "backlog" of customer orders and the inventory of "finished products." At the same time, taking a unit out of the finished goods inventory leads to the issuance of a production order through the kanban that has been detached. This is how the PULL system is triggered.

"Demand" is the external stimulus that triggers the PULL system. An increase in "demand," leads to an increase in the final production rate. This is the causal sequence:

Demand ---> + Backlog ---> + DSR ---> + SR ---> - F. Products

F. Products ---> - Prod. orders ---> + DASR ---> + F. Assmbly rate

SHIPMENTS AND BACKLOG

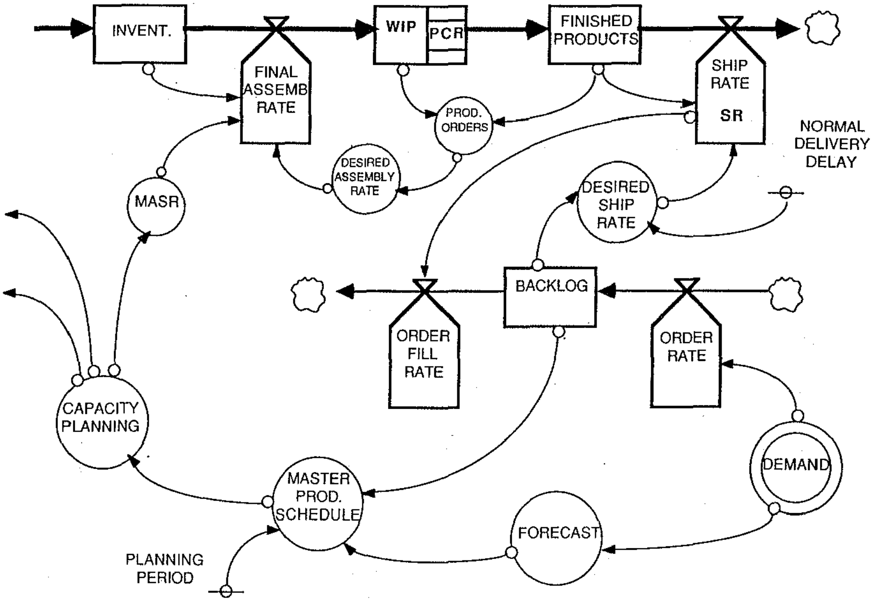


Figure - 7

where "+" stands for increase, and "-" stands for decrease.

Finally, note that shipping rate may also be affected by the availability of finished products inventory in a way similar to how production rates are affected by the availability of inventory in the preceding stages affects.

The figure also shows the master production schedule developed as a function of the backlog (desired ship rate, in fact) and a forecast of demand. This production schedule will be used for developing the monthly "frozen schedule" which will then be used to plan capacity in the different stages (a resource allocation process). Later, with the simulation results, we will have an opportunity to expand on the issue of capacity planning.

A dynamo version of the complete multi-stage model is shown in the appendix.

SIMULATION EXPERIMENTS

The objective of this section is to test the internal consistency of the model by looking at its behavior under different circumstances. In order to do that, a series of different experiments are run on the model. This helps us understand the dynamics of a kanban-based manufacturing system and illustrate ways in which the model can be used for further research.

Each experiment involves one or more simulation runs. The results of each simulation are presented as a set of three graphs, each plotting different variables over time. Each set will be referred to as one single figure. These are the graphs and the variables plotted in each figure:

* Production Rates

- PR3: Final Assembly rate (stage 3)
- PR2: Sub-assembly rate (stage 2)
- PR1: Parts arrival rate (stage 1)
- D: Demand (shown only as a reference)

* Inventories

- INV3: Finished products (stage 3)
- INV2: Sub-assemblies (stage 2)

- INV1: Parts (stage 1)

* Backlog and Shipments

- D: Demand
- B: Customer orders backlog (stage 4)
- SR: Shipping rate (stage 4)

Note that the first 3 stages of the manufacturing chain (1 to 3) are concerned with procurement and production processes. The variables characterizing their behavior are: production rates and level of inventories. The results regarding these variables are found in the first two graphs. The stage at the end of the chain (stage 4) is not concerned with manufacturing activities. Stage 4 is the market interface: it receives, accumulates and dispatches customer orders by shipping units from the inventory of finished products. The results that have to do with this stage are found in the third graph ("Shipment and Backlog").

The series of experiments follows.

Normal response

The first simulation shows the behavior of the system in response to a one-time step increase in demand. The increase in demand is not too big, and falls within the range of changes that the system can handle (there is enough production capacity and enough kanbans for the new production rate).

The results are presented in fig. 8. Note the pull nature of the system. The increase in demand triggers a chain reaction with some delay between stages: ship rate goes up, then final assembly rate follows, then sub-assembly rate, and finally parts arrival rate. As production rates increase, inventory levels decrease. They do so in the same sequence (from end to beginning of the chain): first, finished products; then, sub-assemblies; and finally, parts. The main reason why inventories (of completed units) diminish is because more units are now held as "work-in-process" inventory due to the increase in production rate, and also, because the backlog of production ordering kanbans needs to be larger to signal a higher production rate. (Remember: the sum of WIP inventory, completed units inventory and backlog of production orders is fixed).

Changing the number of kanbans

This experiment is aimed at understanding what happens when the

Fig. 8 - Normal response

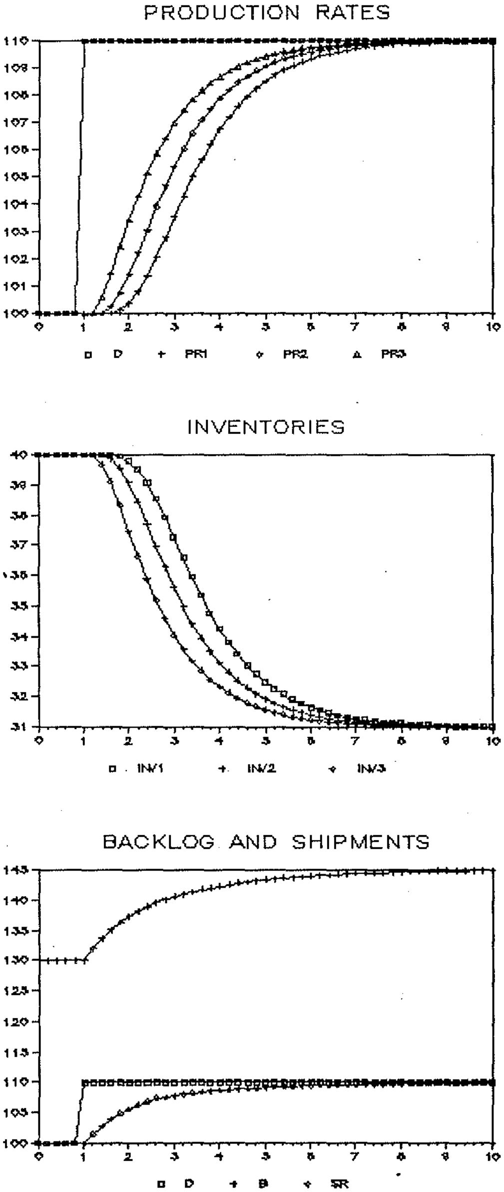
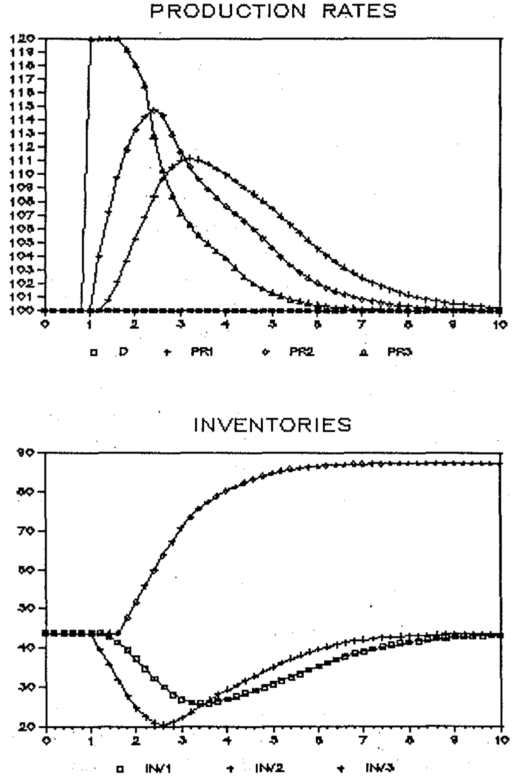


Fig. 9 - Increase in number of kanban



number of kanbans is changed suddenly while other things remain equal. In this simulation run, we are not changing demand. This simulation illustrates some of the subtleties of the kanban system that we saw before.

Figure 9 shows the result of increasing the number of kanbans in stage 3 (final assembly) by 20% in day 1. We see that the immediate result is an abrupt peak in production rate due to the fact that new kanbans enter the system as production ordering kanbans (thus increasing the level of production orders backlog). Note again how the pull system works: it transmits the production peak to the preceding stages of the chain (sub-assembly, and parts).

This sudden increase in production rate is artificial. Since demand stays the same, we know that in the long run production rates will resume their normal value (= demand). What happens is clearly illustrated by the graphs plotting inventories. The introduction of more kanbans in stage 3 (final assembly) has raised the level of "inventory allowed". Due to the goal seeking nature of the main feedback loop (of the "basic structure") the system reacts so as to build up more inventory in stage 3 (finished products inventory). This is why in the new equilibrium this inventory has a higher level than before. During the transient period, the inventories of the preceding stages are temporarily depleted (levels go below their normal value) due to the higher-than-normal rate of production of the succeeding stages (the succeeding stages are withdrawing more units than normal).

Breakdowns

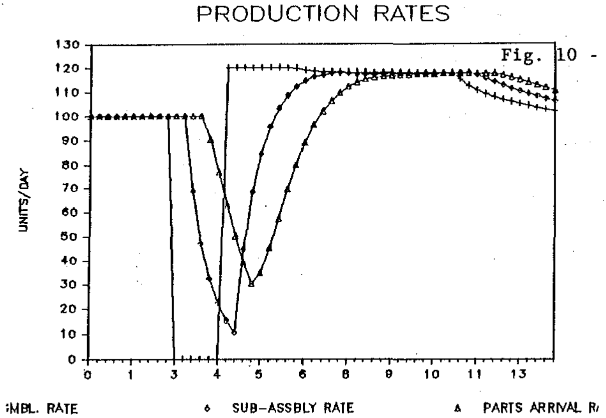
Simulating a machine breakdown is an interesting way to test a pull system.

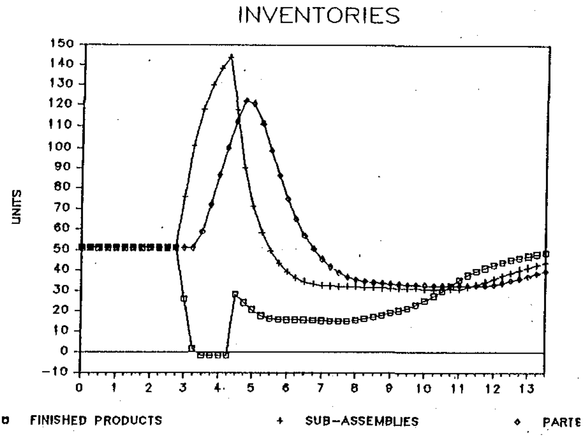
In a "pure" push system, the stages preceding the "broken-down" station continue their production according to their schedule even after the breakdown has occurred. The result is that inventory accumulates at the input of the "broken-down" station.

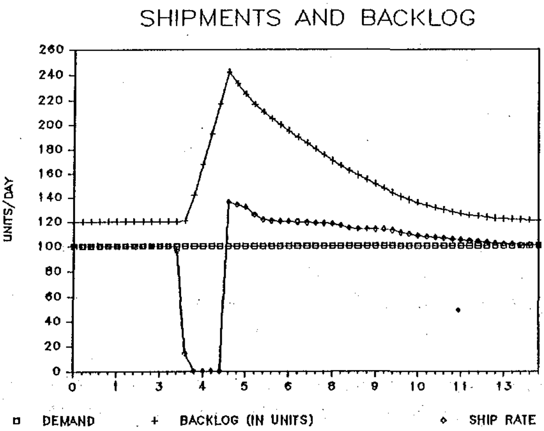
In a pull system, the preceding stage does not produce anything that has not been requested (recently) by the succeeding stage. Therefore, in the event of a breakdown, the flow of production ordering kanbans to the preceding stages ceases. This stops production in the preceding stages.

Figure 10 shows a one-day breakdown in final assembly (the system was previously in equilibrium). We observe how the production rates of the preceding stages do go down after some delay (due to the duration of the kanban cycle). After station 3 has been repaired, its production rate goes up as much as it can go

Fig. 10 - Breakdown







(remember that it is limited by the capacity constraint: "maximum production rate"). The reason for doing so is the big accumulation of customer orders (backlog) that has taken place during the period in which stage 3 was out of order (and therefore the corresponding pressure for delivering finished products). After a few days producing at the "maximum prod. rate" to clear the abnormally high level of customer orders in backlog, production rates resume their normal level (= normal demand) in day 13 and beyond.

It is interesting to see what happens to inventories. Finished products inventory goes to zero once the few units in inventory have been shipped. The preceding stages are not notified instantaneously about the breakdown. Due to the delay of the kanban cycle, they are still producing what they were asked for in the periods before the breakdown. If the situation persisted, these inventories would reach the "maximum inventory allowed" level (determined by the total number of kanbans as we have seen before). But this situation is, of course, temporary. As soon as production activities resume (once stage 3 has been repaired), inventories are being used up again by the succeeding stages in order to produce at what it is now the "maximum production rate". At this point, inventory levels fall below the normal level, because the current production rates are higher than normal. Later, in the steady state all inventories regain their normal, equilibrium level.

The "shipments and backlog" graph shows what happens at the market interface stage. After the breakdown, and once the available inventory of finished products has been exhausted, shipping rate falls to zero. In the meantime, customer orders keep arriving. But, since we are not able to dispatch them, the backlog grows and grows rapidly. After the break-down has been fixed, the natural tendency of the system is to ship as much as possible. This pressure is captured by the variable "desired ship rate" (not shown in this graph). However, there is a constraint on shipments (a "maximum ship rate") which in our model plays a role similar to the capacity constraint in production. In the long run, backlog is reduced to its normal level (1.3 days), and shipping rate matches demand.

Bottlenecks

There are several situations in which we may encounter a bottleneck. The case in which a bottleneck in one stage prevents the system from adequately responding to an increase in demand is particularly interesting.

Two kinds of bottlenecks are possible in one stage of a JIT-kanban system. One is obvious, the other one is not so obvious.

The obvious bottleneck is the classical capacity constraint of a particular stage: when its value happens to be lower than the capacity of the other stages causing the line to be unbalanced. The not-so-obvious bottleneck is due to a constraint in the number of kanbans at a given production stage.

Let's analyze the former case first.

Bottleneck caused by capacity constraint:

This type of bottleneck is simulated in stage number 1 of our model (Parts). That is: "maximum production rate" (MPR) in stage 1 is only 10% above the normal production rate (= normal demand), whilst MPR is 20% above normal. The step increase in demand is 20%.

Figure 11 shows the results of the simulation run. The bottleneck is clearly seen in the graph of production rates. We observe the tendency of production rates to go up in a "chain" response to the increase in demand (in the typical "pull" fashion). The problem arises when parts arrival rate (PR1) reaches its constraint level ("maximum" level = 10% above normal demand). Beyond this point, stage 1 (Parts) does not produce enough to meet the needs of stage 2 (Sub-assembly), and its lack of inventory forces the rate of stage 2 to drop. Similar considerations apply to the effect of stage 2 on stage 3 with some delay. So, in the long run all production rates (and eventually the shipping rate) converge to the value of the production rate of the stage experiencing the bottleneck.

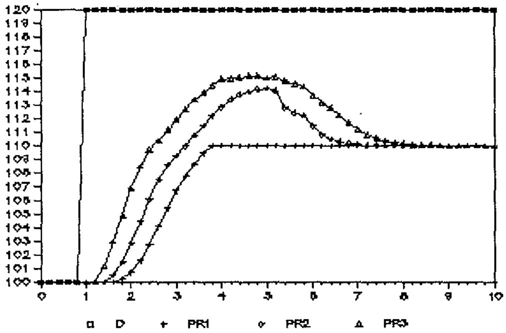
Note the effect on the market interface (stage 4). For a while, ship rate is able to meet demand. But later, when the effect of the bottleneck has been transmitted to the end of the chain, shipping rate drops and backlog grows hopelessly.

Bottleneck caused by Kanban constraint:

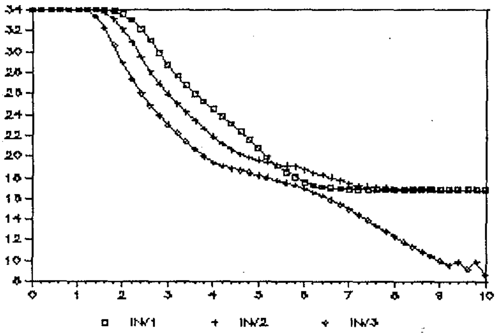
This is the "not-so-obvious" kind of bottleneck. As mentioned in the preceding section, the number of kanbans does not only establish the "maximum inventory allowed" but does in fact determine the slack (flexibility) to place more production orders when required (by an increased in demand, for example). We may have a system with enough production capacity, but a tight inventory policy (small number of kanbans) at the same time. In such a situation, production rate is not limited by capacity. Instead, the lack of kanbans prevents the system from placing more production orders. Production rate is thus limited by the number of kanbans as well.

Fig. 11 - Bottleneck due to capacity constraint

PRODUCTION RATES



INVENTORIES



BACKLOG AND SHIPMENTS

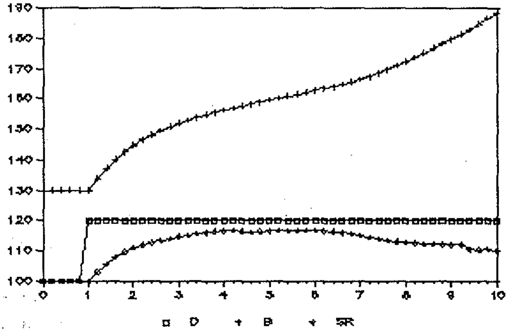
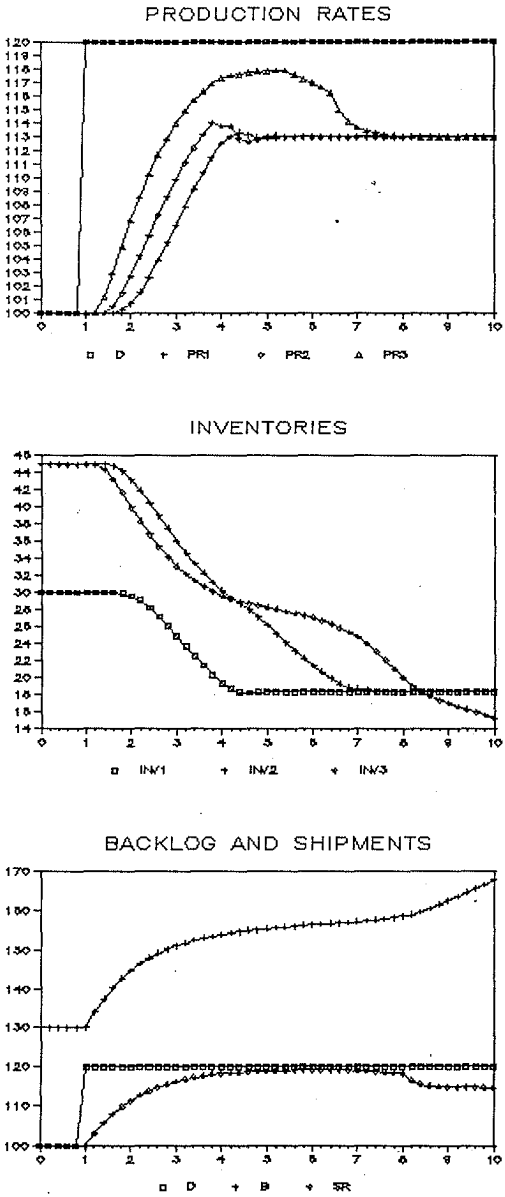


Fig. 12 - Bottleneck due to kanban constraint



This kind of bottleneck is simulated in our model, again, in stage 1. The results of the simulation are shown in figure 12. We can see that the behavior of the system now is similar to the case of a bottleneck caused by a capacity constraint. The causes are, however, quite different.

REFINING THE MODEL TO INCLUDE CAPACITY PLANNING

Until now, we have been assuming that the capacity is fixed. The next set of simulations consider the case in which capacity is in fact changed each time a new master production schedule is developed. As seen in the introductory section on JIT, each month (or every few weeks) a "frozen production schedule" is developed. This schedule is supposed to be the fixed: the "leveled" production rate that will prevail during the month. Some deviations can be permitted later. In practice, the purpose of this pre-established schedule is to allow fabrication, sub-assembly, and supplier supervisors to have advance warning about the scope of the schedule to be run. This allows planning of the workplace: manning, tooling, materials, and preventive maintenance.

This kind of planning has finally been included in our model. There are two main modules (information structures) to consider:

1) the development of a "master production schedule" (based on a forecast of demand and the customer orders backlog), and

2) the assignment of more capacity to the different stations (both in terms of "maximum production rate", and total number of kanbans in the production process).

Figure 7 shows these two blocks.

It is important to realize that we are now departing from what it is the basic kanban system. Things can get very complex as we attempt to link the logic of a push system (like MRP) for planning, with the effectiveness of a JIT system for executing production. New factors now enter the picture. The flexibility of the system to respond to changes in demand may be affected by system parameters like:

* length of the planning period (amount of time over which the overall production rate is "frozen")

* which in turn depends on how much time is required physically and organizationally for preparation and allocation of resources.

* the constraint on capacity change from one planning period to the next. How much can we change in one period? What flexibility do we have in capacity change?

* forecasting : Is forecast just the smoothing of past demand? or, is it based on other exogenous factors that anticipate long term trends?

* the way in which the master production schedule is developed: is it based on the forecast? or, is it based on the backlog of accumulated customer orders? or, both?

The link of MRP and JIT is an entire subject per se. Here, we do not pretend to explore all of the above points. The objective is to show that the model can certainly be used to get some insights into some of these issues.

So, we will be tackling just one issue of the above list: the effect of the "master production schedule" on the flexibility of the entire system to respond to increases in demand (beyond the range of what the "basic" kanban system is able to handle).

Simulations experiments with capacity planning:

The simulation starts with a one-time step increase in demand which is higher than the existing "maximum production rate". As we have seen in the study of the bottleneck, backlog keeps growing in the steady state because there is not enough production capacity to meet demand. In the next planning period, new capacity will be added. The question is how much? What should be the next "leveled" production schedule?

If we look at the backlog the pressure is tremendous. If we set the next "leveled" schedule to the current "desired rate", we may end up with excess capacity once the backlog has been reduced. On the other hand, if we look at the forecast (which is basically a smoothing of past demand) it will tell us that demand seems to be increasing and will report just a fraction of the actual step increase (because it will still be averaging over time). If we set the "leveled" schedule at the value given by the forecast we will be below the real need.

A linear combination of both has been chosen to develop the "frozen" schedule for the next planning period. That is:

Production plan = X * Forecast + (1-X) * Desired Ship Rate

Note that when X = 1 , the production schedule is determined exclusively by the forecast. When X = 0 , the production schedule is entirely determined by "desired ship rate".

Figures 13 and 13 show the result of simulations using different values for X.

We can see that X = 1 leads to a very reactive response. The production plan ("leveled" schedule) that results at each planning period always falls short of real needs. As a consequence, we have to use overtime almost at all times. It is interesting to see how, in fig. 18, production rates are above the production plan (the square symbol line) staying at their "maximum production rate" (a 5% above the production plan).

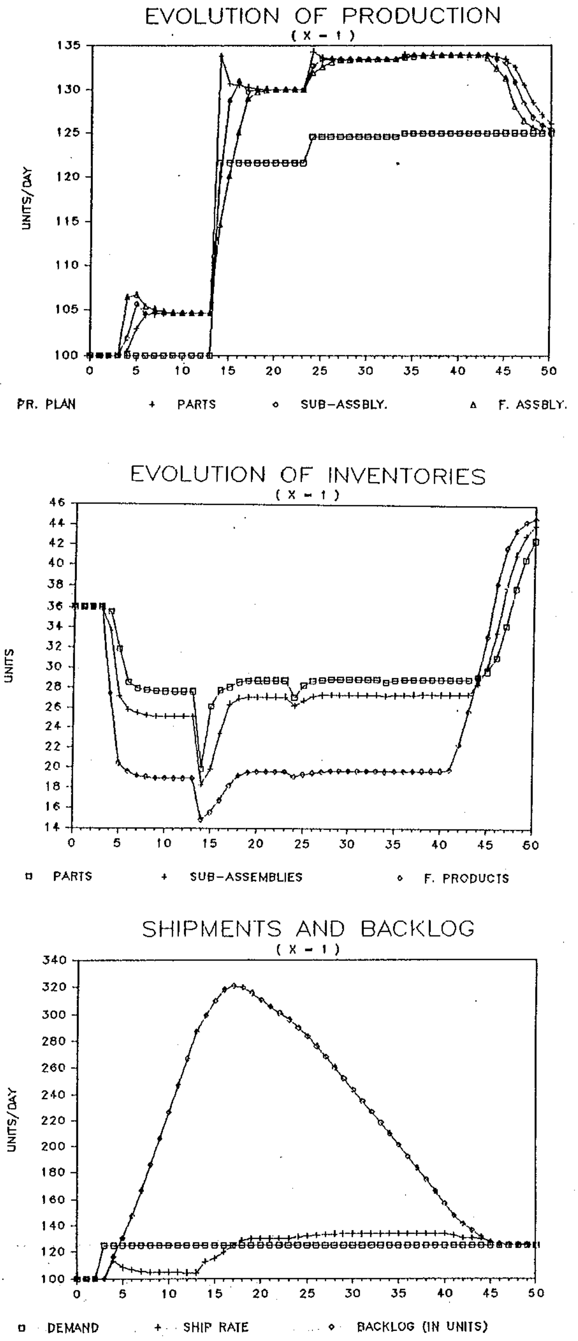
The situation is different when we consider the case of X = 0.8 What happens here is an "overshoot:" the production schedule jumps to the level of 155 units/day which is above the long term equilibrium of 120 units/date (= new demand). This level represents 55% increase in capacity. We may well question whether this would be feasible in real world (this goes back to one of the points in the above list: the constraints on capacity change). Note also in the same figure, that some overtime takes place at the beginning of the planning period (days 14-20), but then production rates begin to fall below the "leveled" production schedule. Production rates would get down to the new demand (120 units/day), if the "leveled" schedule remained at the high level of 155 units/day. In the next planning period (day 24), the new "leveled" production schedule is finally set to 120 units/day. At that point, and despite the reduction of production capacity, we observe considerable undertime. What is the reason for this apparent contradiction?

The reason has to do with the subtleties of the kanban system: when the new production schedule is developed and issued to all stages, the capacity planning module does not only change the capacity constraint of each stage but also the number of kanbans. In our case, production capacity and number of kanbans are reduced in day 24. The withdrawal of kanbans from the system at that point gets translated into an immediate goal: reduction of inventory! The "maximum inventory allowed" is now much less than what we had in the previous period (days 14-23). Therefore production rates are slowed down until this "all-of-a-sudden" excess inventories is cleared.

The difference between the two ways of developing the production schedule is dramatic when we compare the backlogs in figure 13 and figure 14.

For X = 1 (a reactive production schedule), backlog goes up to 320 orders and it takes 45 days to recover. Instead, for X = 0.8 (a

Fig. 13


EVOLUTION OF PRODUCTION
(X - 1)
UNITS/DAY
135
130
125
120
115
110
105
100
0
5
10
15
20
25
30
35
40
45
50
PR. PLAN
+ PARTS
◇ SUB-ASSBLY.
△ F. ASSBLY.
EVOLUTION OF INVENTORIES
(X - 1)
UNITS
46
44
42
40
38
36
34
32
30
28
26
24
22
20
18
16
14
0
5
10
15
20
25
30
35
40
45
50
□ PARTS
+ SUB-ASSEMBLIES
◇ F. PRODUCTS
SHIPMENTS AND BACKLOG
(X - 1)
UNITS/DAY
340
320
300
280
260
240
220
200
180
160
140
120
100
0
5
10
15
20
25
30
35
40
45
50
□ DEMAND
+ SHIP RATE
◇ BACKLOG (IN UNITS)

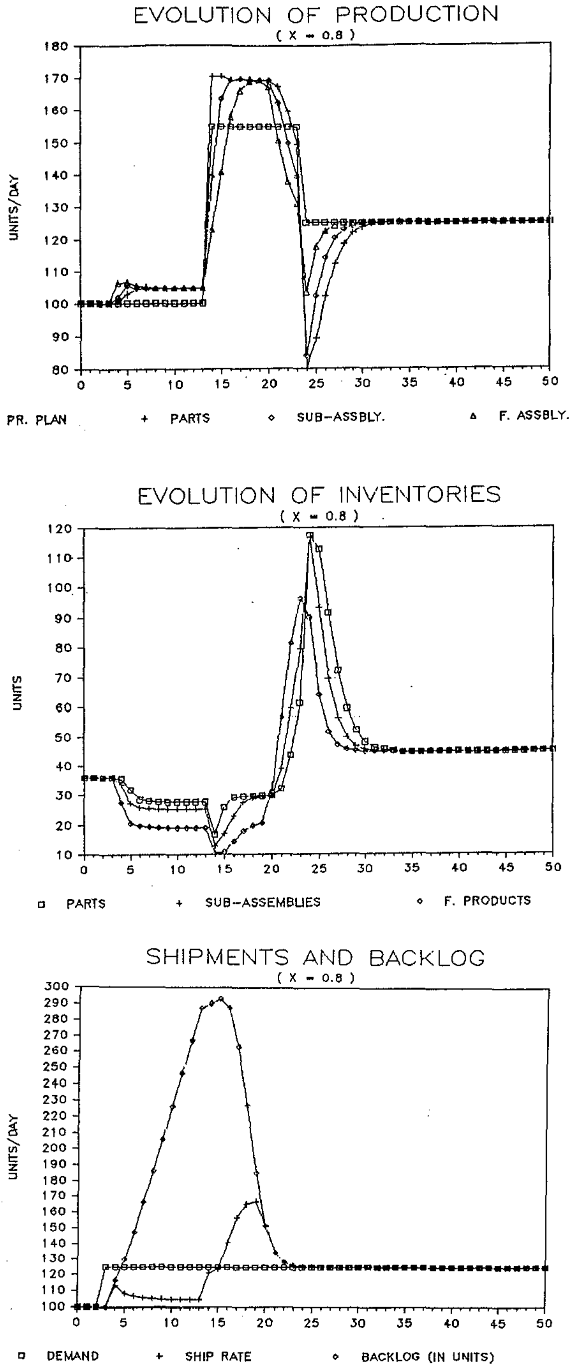

EVOLUTION OF PRODUCTION
(X = 0.8)
UNITS/DAY
180
170
160
150
140
130
120
110
100
90
80
0
5
10
15
20
25
30
35
40
45
50
PR. PLAN
+ PARTS
◇ SUB-ASSBLY.
△ F. ASSBLY.
EVOLUTION OF INVENTORIES
(X = 0.8)
UNITS
120
110
100
90
80
70
60
50
40
30
20
10
0
5
10
15
20
25
30
35
40
45
50
□ PARTS
+ SUB-ASSEMBLIES
◇ F. PRODUCTS
SHIPMENTS AND BACKLOG
(X = 0.8)
UNITS/DAY
300
290
280
270
260
250
240
230
220
210
200
190
180
170
160
150
140
130
120
110
100
0
5
10
15
20
25
30
35
40
45
50
□ DEMAND
+ SHIP RATE
◇ BACKLOG (IN UNITS)

Fig. 14

more aggressive, more "nervous" way to develop the production schedule), backlog peaks at 290 orders and it takes only 22 days.

From a service point of view the "aggressive way" may be more interesting than the "reactive". But, has the company enough flexibility to change capacity so much, so often? what are the costs?

The answers to these questions are beyond what this model can provide.

Our model has shown, however, that the flexibility of a company adopting JIT-kanban will not depend so much on the kanban system as it does on other factors, such as the ability to change capacity, and the ability to plan production. Only when demand fluctuations are small does the basic kanban system seem to be an effective way to respond to demand uncertainties.

CONCLUSION

A system dynamics model of a kanban-based just-in-time (JIT) production system has been developed. The production process is a simple three stage transfer line. The modeling emphasis has been on classical production scheduling and production smoothing. The model has been used to examine the response of the production system to small shocks, such as small changes in demand, when the system is run on "automatic" mode, i.e. management is allowed to make only very routine responses (such as overtime). The shocks are simulated with different levels of kanbans in the system. The second use of the model has begun to explore larger shocks, when management is allowed to change capacity.

Deliberately, this paper has been quite technique oriented, rather than problem oriented. Its objective was to show that a JIT-kanban system could be built using the structuring principles of system dynamics modeling. Hence, the paper has devoted a long extension to check the internal consistency of the model by exposing it to a series of experiments.

The model has limitations that could be overcome in future versions. For example, the transfer line is very simple: one product, one linear sequence of material flow. Given that set-up times have been used to motivate some of the formulas, it seems that a multi product model would be more appropriate. This is true. Mixed production should definitely be the enhancement to include in the next version of the model. Then, issues like product mix flexibility (and not just volume flexibility) could be explored.

Finally, from a Production and Operations Management perspective, the contribution of the model (as presented in this paper) is more latent than real. However, the paper lays the groundwork for more normative research.

By concentrating on management decisions and policies, research based on this model could be useful to practitioners. For a given manufacturing system, what are the objectives that JIT-Kanban tries to accomplish? What are its problems? What are the levers available to managers of the transfer line? How should they be used? How does the system dynamics model give them insight into what to do?

In summary, subsequent research ought to focus on the "burning" issues in the management of JIT/kanban systems, and use the model to shed some light on them.

APPENDIX-A: OPERATION OF THE KANBAN SYSTEM

This is how the Kanbans are used:

1 - Production Kanban:

The parts processed at a certain stage are put in a container. A Kanban is attached or hung on the container and then stored at the location designated by the Kanban (1) (see figure 15).

When a succeeding process withdraws this part or material, a worker lifts off the Kanban and puts it into a Kanban box (2). Kanbans are collected from the box at regular intervals and hung on hooks on a schedule board. The sequence of various Kanbans on the board shows-workers the dispatching order of jobs in the process (3).

A worker produces various items, in accordance with the sequence of the various Kanbans on the board, as indicated by the Kanban, at the rate which is set in advance. The Kanban itself moves in the process with the first unit of the batch (4).

The procedure (1) through (4) is repeated and the production is continued effectively.

Note that it is probable in procedure (2) that if the succeeding process never withdraws material from the preceding process then the Kanban is neither collected from the Kanban box nor is it hung on a hook of the schedule board. Consequently the item is never processed at this shop.

2 - Withdrawal Kanban:

The withdrawal Kanban is used for moving materials from the output of one process to the input of another (distant) process. The withdrawal Kanban is handled in a way similar to the production Kanban, with the difference that the process is transportation instead of manufacturing (see figure 16).

We should keep in mind that it is also the rule that withdrawals are equal to what the Kanban indicates and nothing will be withdrawn unless a Kanban is in the box.

In figure 16, broken lines imply a production Kanban and its movement in the preceding shop. When material or parts are withdrawn from storage, a production Kanban on the container is exchanged with a withdrawal Kanban. The production Kanban removed will be transferred to a production Kanban (collection) box.

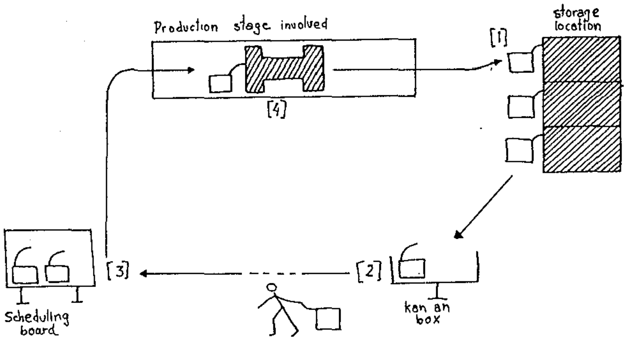

Production stage involved
[1]
storage location
[4]
[3]
[2]
Scheduling board
kan an box

Fig. 15

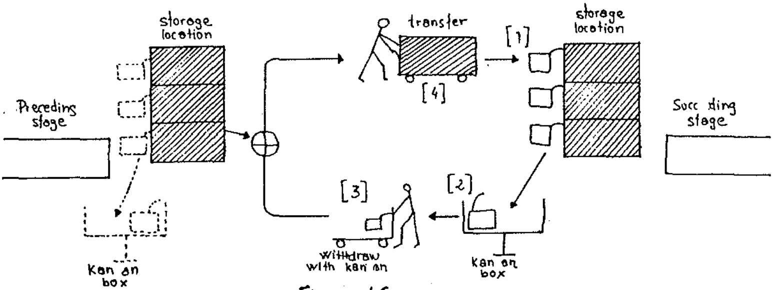

storage location
Preceding stage
transfer
[1]
storage location
[4]
Succ ding stage
[3]
[2]
Kan an box
withdraw with kan an
kan an box

Fig. 16

APPENDIX - B

DETERMINATION OF THE NUMBER OF PRODUCTION KANBANS:

In a "constant cycle" inventory control system, the re-order date is fixed and the quantity ordered depends on the usage since the previous order was placed and on the outlook during the lead time.

The following formula is used to calculate the maximum inventory:

Maximum Inventory =

= daily demand x (order cycle + lead time) + safety stock

where the order cycle is the time interval between an order time and the next order time and the lead time is simply the time interval between placing an order and receiving delivery.

Theoretically, the order cycle is determined by the formula:

order cycle = (economic lot size) / daily demand

In practice however, the order cycle is often determined by external constraints such as steps in the monthly production scheduling or a contract between the supplier and the manufacturer.

The following formula is used for computing the total number of Kanbans:

$$\text{Total number of Kanban} = \frac{\text{Maximum Inventory}}{\text{Container Capacity}} =$$

$$= \frac{[\text{ daily demand x (order cycle + lead time + safety period) }]}{\text{container capacity}}$$

DETERMINATION OF THE ORDER QUANTITY:

The order quantity in a "constant cycle" inventory control system is given by the formula:

(eq. I) Order Quantity = (maximum inventory - existing inventory) - (orders placed but not yet received)

With a Kanban system, this order quantity is automatically specified by the number of Kanbans detached by the time of regular Kanban collection since the previous collection. That is:

(eq. II) Order Quantity =

= (number of Kanban detached by the time of regular Kanban collection since the previous collection) x container capacity

The reason that validates this expression for order quantity is due to this relationship:

(Number of Kanbans detached since the previous collection of Kanbans) =

= (Total number of Kanbans) - (number of Kanbans attached to the existing inventory at the subsequent store) - (Number of Kanban still kept in the preceding process)

The kanbans move in a circular fashion. A production kanban can only be in three places: either 1) attached to work-in-process inventory at the current store, or 2) attached to the inventory of finished units at the subsequent store, or 3) detached and put into a kanban collection box where they become production orders. Since, the total number of kanbans is fixed, the number of kanbans detached (which become production orders) can be expressed as the above difference. So, there is no need to compute the order quantity by using the formula above (eq.I). The order quantity is automatically given by the kanbans detached by the subsequent process (eq. II).

APPENDIX - C

A DYNAMO VERSION OF THE MULTI-STAGE MODEL

```
*
*
*
*  K14: 3 Stage JIT - Kanban system
*
*                                by Ramon  O'Callaghan, Nov. 1985
*
*
*  Demand
*
A     D.K=ND*(1+STEP(SD,TSD))                       Demand
C     ND=100 UNITS/DAY                              Normal demand
C     SD=.20                                        Step increase
C     TSD=2  DAYS                                   Time of step increase
*
*  Backlog (stock and flow)
*
L     B.K=B.J+(DT)(OR.JK-OFR.JK)                    Backlog
N     B=TI4*D
R     OR.KL=D.K                                     Customer order rate
R     OFR.KL=SR.JK                                  Order fill rate
*
* Production Planning (developing a frozen schedule)
*
L     PP.K=PP.J+(DT)(CPP.JK)                        Production plan
N     PP=DSR
R     CPP.KL=PULSE(1,TSD,PPT)*(NPP.K-PP.K)/DT       Change in production plan
A     NPP.K=(X)*F.K+(1-X)*DSR.K
C     X=.8
A     F.K=SMOOTH(D.K,TAD)                            Forecast
C     TAD=5 DAYS
C     PPT=10  DAYS                                   Planning period
A     NK1.K=((PP.K/CC1)*(LT1+TI1)*(1+ALPHA))         Changing kanbans according
A     NK2.K=((PP.K/CC2)*(LT2+TI2)*(1+ALPHA))            to production plan
A     NK3.K=((PP.K/CC3)*(LT3+TI3)*(1+ALPHA))
C     ALPHA=.3                                       Safety coefficient
*
*  Shipping policies
*
R     SR.KL=CLIP(MSR.K,DSR.K,DSR.K,MSR.K)*SW4.K     Ship rate
A     MSR.K=2*PP.K                                   Max. ship rate
A     DSR.K=B.K/TI4                                  Desired ship rate
C     TI4=1.3   DAYS                                 Normal delivery delay
A     SW4.K=TABLE(TSW4,CI3.K,0,10,1)                 Effect of inventory
T     TSW4=0/.8/.95/1/1/1/1/1/1/1/1                      on ship rate
```

```
A     CI3.K=I3.K/CC3
C     CC3=10  UNITS/CONTAINER
*
*
*  3) Final Assembly (PR3, PCR3)
*
*   stocks
*
L     I3.K=I3.J+(DT)(PCR3.JK-SR.JK)          Inventory (completed units)
N     I3=NK3*CC3-(ND*LT3)-(ND*TI3)
L     WIP3.K=WIP3.J+(DT)(PR3.JK-PCR3.JK)     Work-in-process inventory
N     WIP3=ND*LT3
*
*  flows
*
R     PCR3.KL=DELAY3(PR3.JK,LT3)             Product completion rate
C     LT3=.5  DAYS                           Production lead time
R     PR3.KL=CLIP(MPR3.K,DPR3.K,DPR3.K,MPR3.K)*SW3.K     Prod. rate
A     MPR3.K=1.1*PP.K                        Max. prod. rate
A     SW3.K=TABLE(TSW3,CI2.K,0,10,1)         Effect of inventory of
T     TSW3=0/.65/.9/1/1/1/1/1/1/1/1            preceding stage
A     CI2.K=I2.K/CC2
C     CC2=10  UNITS/CONTAINER
*
*  information feedback and policies
*
A     PO3.K=NK3.K*CC3-I3.K-WIP3.K                 Production orders
A     DPR3.K=PO3.K/TI3                            Desired prod. rate
C     TI3=.5   DAY                                 Kanban cycle
*
*
*  2) Sub-Assembly  (PR2, PCR2)
*
L     I2.K=I2.J+(DT)(PCR2.JK-PR3.JK)
N     I2=NK2*CC2-(ND*LT2)-(ND*TI2)
L     WIP2.K=WIP2.J+(DT)(PR2.JK-PCR2.JK)
N     WIP2=ND*LT2
*
*  flows
*
R     PCR2.KL=DELAY3(PR2.JK,LT2)
C     LT2=.5  DAYS
R     PR2.KL=CLIP(MPR2.K,DPR2.K,DPR2.K,MPR2.K)*SW2.K
A     MPR2.K=1.1*PP.K
A     SW2.K=TABLE(TSW2,CI1.K,0,10,1)
T     TSW2=0/.65/.9/1/1/1/1/1/1/1/1
A     CI1.K=I1.K/CC1
C     CC1=10   UNITS/CONTAINER
*
*  information feedback and policies
*
```

```
A      PO2.K=NK2.K*CC2-I2.K-WIP2.K
A      DPR2.K=PO2.K/TI2
C      TI2=.5 DAY
*
*
* 1) Parts  (PR1, PCR1)
*
L      I1.K=I1.J+(DT)(PCR1.JK-PR2.JK)
N      I1=NK1*CC1-(ND*LT1)-(ND*TI1)
L      WIP1.K=WIP1.J+(DT)(PR1.JK-PCR1.JK)
N      WIP1=ND*LT1
*
*  flows
*
R      PCR1.KL=DELAY3(PR1.JK,LT1)
C      LT1=.5 DAYS
R      PR1.KL=CLIP(MPR1.K,DPR1.K,DPR1.K,MPR1.K)
A      MPR1.K=1.1*PP.K
*
*  information feedback and policies
*
A      PO1.K=NK1.K*CC1-I1.K-WIP1.K
A      DPR1.K=PO1.K/TI1
C      TI1=.5 DAY
*
*
*  Control statements
*
SPEC LENGTH=50/DT=.05/PRTPER=1
PRINT D,SR,B,PP,PR1,PR2,PR3,I1,I2,I3
RUN
```

BIBLIOGRAPHY AND REFERENCES

- "Readings in Zero Inventories", Transcripts of papers presented at the APICS 27th Annual International Conference, Las Vegas, Oct. 1984
- "A Simulation Analysis of the Japanese Just-In-Time Technique (with Kanbans) for a Multiline, Multistage Production System", Decision Sciences, 1983.
- Hall. "Zero Inventories," Homewood Ill.: Irwin, 1983
- Kimura and H. Terada. "Design and Analysis of Pull System, A method of Multi-Stage Production Control," International Journal of Production Research, 1981, vol 19, no 3, p241
- Monden. "The Toyota Production System," Atlanta Ga.: Industrial Engineering and Management Press, 1983
- Morecroft. "A Systems Perspective on Material Requirements Planning," Decision Sciences, 1983, vol.14, p.1-18.
- Schonberger. "Some Observations on the Advantages and Implementation Issues of JIT production Systems," Schonberger, Journal of Operations Management, Vol 3, No. 1, Nov. 1982, p 1
- Schonberger. "Japanese Manufacturing Techniques," New York: The Free Press, 1982.
- Sepehri. "How Kanban System is used in an American Toyota Motor Facility," I.E. , February 1985, p50-56