A Comparison of Production-Line Control Mechanisms[1]

Asbjoern M. Bonvik[2] Christopher Couch[3]
Stanley B. Gershwin[4]

April 5, 1996

[1]Accepted for publication in the *International Journal of Production Research.* This work was supported in part by the Advanced Research Projects Agency under contract N00174-93-C-0035, by the National Science Foundation under grant DDM-9216358, and by the Norwegian Research Council under contract 100250/410.

[2]Operations Research Center, MIT E40-149, 77 Massachusetts Avenue, Cambridge, MA 02139-4307

[3]Sloan School of Management, MIT E56-290, 77 Massachusetts Avenue, Cambridge, MA 02139-4307 (on leave from Vehicle Production Engineering, Toyota Motor Corporation, Toyota City, Aichi, Japan)

[4]Laboratory for Manufacturing and Productivity, MIT 35-331, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, *to whom correspondence should be addressed.*

Abstract

We study the performance of the kanban, minimal blocking, basestock, CONWIP, and hybrid kanban-CONWIP control policies in a four-machine tandem production line making parts for an automobile assembly line. Cases with both constant and changing demand rates are studied. The main performance measures are the service level and the amount of work-in-progress. We also consider other performance measures such as variability amplification along the line. The results are obtained by extensive simulations.

We find that the best parameter choices for the hybrid policy decrease inventories by 10% to 20% over the best kanban policy while maintaining the same service levels. The inventory difference grows as the demands on service level increase. The performance of basestock and CONWIP policies falls between those of the kanban and hybrid policies.

The CONWIP and hybrid policies also give significantly better response to changes in the demand rate.

1 Introduction

Despite its significant success, kanban control is not flawless. Kanban means card in Japanese, and refers to a shop floor control policy synchronizing manufacturing processes that make parts with those processes that consume the parts. In the spirit of *kaizen* — continuous improvement — we try to find similar control policies that improve the system behavior while retaining the features of kanban control that makes it such an effective part of the overall production system. This paper presents results from a study of a typical component production line in Toyota Motor Company.

There has been much work on individual operating disciplines, but relatively few comparison studies have been done. This may be partly because the different approaches have not been described within the same nomenclature. Common frameworks have recently been developed (Buzacott and Shantikumar 1992; Frein, Di Mascolo, and Dallery 1994), making it easier to compare different policies. The policies we study are basestock, CONWIP, kanban, and hybrids thereof. These can be implemented by circulating kanban-like cards, but differ in the pattern of information flow and the resulting behavior of inventory.

The classical reference on basestock is Clark and Scarf (1960), where optimality is shown for a system with constant leadtimes and unlimited machine capacities. A paper by Kimball (1988) presents this policy in a more accessible form, and Lee and Zipkin (1992) develop an approximate performance analysis of this policy under the assumption of exponential service times and Poisson demand. One interesting observation is that a production stage controlled by basestock maintains a certain inventory ahead of demand. If the demand process is deterministic, this is equivalent to the hedging point policies developed from an optimal control perspective by Kimemia and Gershwin (1983) and others. These policies are optimal for a single unreliable machine in constant demand. See, for instance, Bielecki and Kumar (1988).

There is an immense kanban literature, and one survey paper is Berkley (1992). Tabe, Muramatsu, and Tanaka (1980) compare systems described as *push* and *pull*. The pull system is similar to the Toyota kanban system, and the push system is driven by unreliable demand forecasts. The main performance measure in this paper is the amplification of ordering quantity and inventory level variance along the line. This is shown to be quite sensitive to forecasting errors. Kimura and Terada (1981) describe the Toyota kanban system in detail. They again compare this to the so-called push system, using variance amplification as a performance measure. They show that the batch sizes and production lead time of each stage have a major impact on the inventory fluctuations under kanban control, whereas the main factor in the push system again is the degree of forecasting error. We will use more direct performance measures like throughput and inventory levels, but will also report on variability amplification to connect with these influential papers. All the policies we study are driven by actual demand realizations, not forecasts.

Optimizing the number of kanban in a line has been a popular research topic, but it seems that most kanban implementations set these parameters by rules of thumb or simple formulas. An example of such a formula is Toyota's $n \geq DL(1+\alpha)/a$, where n is the number of kanban, D is the demand rate, L the replenishment lead time, α a safety factor, and a the number of parts in a container (Sugimori, Kusunoki, Cho, and Uchikawa 1977). During factory operation, the kanban numbers are steadily decreased by reducing the safety factor. A less than satisfying aspect of this formula is that it is based on standard leadtimes, and thus does not reflect the lead time consequences of shop floor congestion and limited machine capacities.

Berkley (1991) shows that a common model of kanban systems is equivalent to a traditional tandem production line with finite buffers. This model assumes that kanban travel instantly to their destinations when they are detached from a part, and that both kanban and parts travel in quantities of one. If the line operates by blocking before service (information blocking) the buffer size is the same as the number of kanban, and if it operates by blocking after service, the buffer size is one less. The usual assumption in the transfer line literature is that the last machine in the line is never blocked, which is not the case in a pull system. However, the transfer line literature is applicable to kanban systems in saturating demand. An overview is given by Dallery and Gershwin (1992).

Another kanban model, which we call *minimal blocking*, has been studied by several authors, starting with So and Pinault (1988). They state that the minimal blocking model "may be different from the operating concept of a typical pull system", but justify this to facilitate machine recovery from failures and to keep bottlenecks working even if there are failed machines downstream. We will show that the hybrid policy we propose is a better way of achieving this result. Buzacott and Shantikumar (1992) define kanban control as equivalent to the minimal blocking model. Mitra and Mitrani (1990) show that the minimal blocking model dominates the tandem buffer model in terms of output. Mitra and Mitrani (1991) extends this study to a line with Poisson demand, which has also been studied by Di Mascolo, Frein, and Dallery (1994).

Spearman, Woodruff, and Hopp (1990) introduce the constant work-in-progress (CONWIP) policy. The main justification is to extend the scope of just-in-time manufacturing to systems where kanban is inappropriate. Kanban is only suitable for a high volume production environment with relatively few part types. It is not useful in an environment with expensive items that are rarely ordered, since it would require at least one of each kind of item to be in inventory at all times. CONWIP instead limits the total inventory of all part types in the system and allows the part type mix and inventory location to vary as appropriate.

Van Ryzin, Lou, and Gershwin (1993) develop a control scheme that combines the basestock and kanban approach. They study a two-machine tandem line in constant demand by numerically solving an optimal control dynamic programming problem and observe that the optimal solution can be closely approximated by two boundaries: a hedging point for each machine and a finite buffer. This work has been followed up by Yan, Lou, Sethi, Gardel, and Deosthali (1994), where the two-boundary policies behave very well in a simulation study of semiconductor manufacturing. These policies are particularly attractive in systems subject to demand changes.

Veatch and Wein (1994) show that the basestock policies are never exactly optimal for a two-machine line with Poisson demand and exponential processing times. The difficulty with such policies is the possibility of accumulating large inventories that stay in the system for extended periods, because of capacity limitations. They also solve a dynamic program numerically and exhibit optimal policies similar to those found by Van Ryzin, Lou, and Gershwin. In the simulation experiments of Veatch and Wein, the best basestock policies come close to optimal. Depending on the parameters of the system, basestock policies may or may not be better than kanban policies. Kanban gave better results when the downstream machine is the bottleneck, while basestock gave better results when the bottleneck is upstream. In an earlier simulation study of semiconductor manufacturing, Wein (1988) demonstrated that the release policy into a manufacturing system has a greater impact on the system performance than the dispatch policy followed within the system.

2 Methods

2.1 Model

We approach this matter through a series of simulation experiments. The model used was written for the purpose in the C programming language. It uses the UNIX system function *random* to generate pseudorandom numbers, and converts them to the needed distributions by methods described in Rubinstein (1981). To supply the computing power needed for our experiments, we distributed the simulations on a network of twelve Sun workstations.

The model was validated by comparing its output to results derived from numerical solutions of Markov chain formulations of the same systems. This was done for one, two, and three stage lines with exponential failure, repair, and operation time distributions. The lines were controlled by kanban, basestock, and hybrid policies in Poisson and saturating demand. This was done through a series of experiments where simulation results with tight 95% confidence intervals were obtained through a large number (50-100) of replications of the same data, each with a different random number seed. Each replication was run for 120,000 units of simulated time, with an additional warm-up period of 9,600 time units before any data logging started. The confidence intervals were estimated by large sample methods described in Arnold (1990).

We make some common assumptions across all scheduling policies:

- The system makes a single part type.
- Material is transported in units of one without delay.
- Information, such as kanban, is transmitted instantly.
- Machines operate asynchronously, so parts can be loaded whenever a part is present and the proper authorization has been received.
- Jobs authorized for loading follow a first come, first serve dispatch policy at all machines.
- The time to fail measures spent working time, not clock time. This is known as *operation dependent failures* (Buzacott and Hanifin 1978), and implies that no machine can fail unless it is working on a part.

- Any demand that cannot be satisfied from finished goods inventory is lost to the system. This reflects an assembly line stockout, where the entire line stops until parts arrive. In a Toyota plant, stoppages are compensated for by working longer hours until the production plan for the day is met. One can therefore say that the backlog information is maintained elsewhere in the factory.

There are several reasons for making these assumptions: One is to simplify our study by reducing the number of variables considered. Another reason is that several of the variables we left out, such as kanban transmittal time and batch size, can be seen as implementation details rather than essential aspects of the different control policies. By creating an ideal implementation, we will illuminate the inherent behavior and limitations of the control policies. Finally, we have adapted the control policies to the system we study, and have made a non-standard assumption about the demand backlog behavior. This assumption is not essential to our control policies; they are easily adapted to different production systems.

2.2 Control policies

2.2.1 Kanban and minimal blocking

Toyota uses a two-card kanban system, where one type of card controls production at a stage and the other type controls transportation between stages. We model this as a line synchronized through finite tandem buffers with blocking before service. This control discipline limits the amount of inventory to a fixed maximum for each cell consisting of a machine and its output buffer, where the maximum is equal to the number of kanban of both types circulating within the cell. Figure 1 shows kanban movement by dashed arrows and the control cells as rectangles, where N_i is the number of kanban circulating in cell i. Each cell receives control information from its immediate downstream neighbor. This information arrives in the form of the kanban cards detached from pieces as they are loaded on the machine. If a machine is down, it will not load material, and no demand information will be propagated upstream.

The minimal blocking policy is a variation of kanban control, where the total inventory in a cell consisting of a machine and its input and output buffers is limited, as indicated in figure 2. The difference between the tandem buffer kanban model and the minimal blocking model is that if the machine

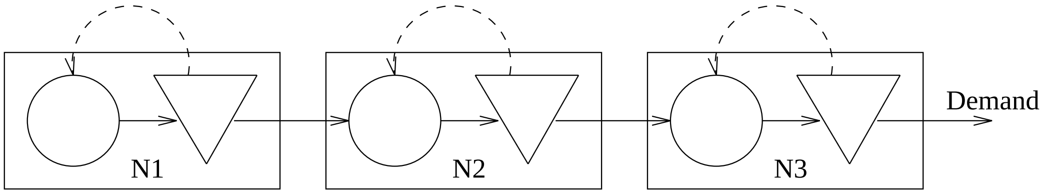

Figure 1: A production line controlled by kanban.

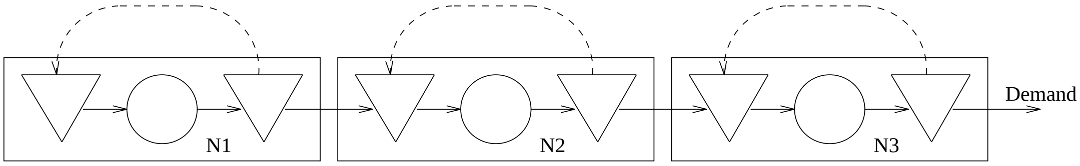

Figure 2: A production line controlled by minimal blocking.

upstream of a buffer finishes its operation before the one downstream of the buffer, and a demand event occurs at the downstream machine in the meantime, the upstream machine can become unblocked and start a new operation. In our tandem buffer kanban model, this will not happen before the downstream machine has finished its operation and loaded another part. This gives the minimal blocking model a higher throughput than the tandem buffer model for any given set of buffer sizes, but the consequences on WIP are not obvious.

Our reason for choosing a tandem buffer model as the primary kanban model is that the minimal blocking model would require a transportation kanban to be dispatched upstream of a machine whenever a piece is taken out of the machine's output buffer. By Toyota shop floor rules, transportation kanban are only dispatched as a consequence of parts actually being loaded on the machine, so the tandem buffer model is a better representation of the operating practice in the plant we study.

2.2.2 Basestock

Basestock control limits the amount of inventory between each production stage and the demand process. Each machine tries to maintain a certain amount of material in its output buffer, subtracting backlogged finished goods demand, if any. This amount is called the *basestock level* of the machine (Kimball 1988).

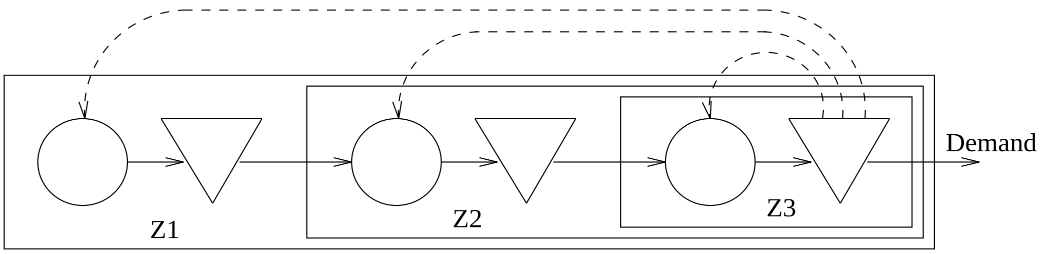

Figure 3: A production line controlled by basestock.

To operate a basestock control, it is necessary to transmit demand information to all production stages as demand occurs. This can be done by using a card-based system similar to kanban control: Attach a set of cards to each unit of finished goods, using as many cards as there are production stages. When a unit is delivered to satisfy demand, remove its cards, and distribute one card to each production stage. This authorizes production in the same way as in kanban control. When an operation is completed, the card is attached to the product, and will follow the product until it leaves the finished goods buffer. If there is only one production stage, this control is identical to kanban. The difference for longer lines is in the flow of information, since basestock control distributes demand information to all stages simultaneously. Different basestock levels are attained by sending out an appropriate number of initialization cards to the various machines to fill buffers to the basestock levels. Basestock control is traditionally implemented this way, with the cards referred to as *work orders* (Lee and Zipkin 1992; Kimball 1988).

When a machine fails, the demand process will continue to remove material from the finished goods inventory, and the machines downstream of the failure will operate normally until they become starved of parts to process. The machines upstream of the failed machine will receive demand information directly from the demand process and will operate as usual. There will therefore be a build-up of inventory in front of the failed machine. If backlogging of demand is permitted, there is no upper limit on this inventory accumulation.

In our experiments, we will be mainly concerned with the total inventory in the line, summing over all machines and buffers. This is the same as applying equal costs to inventory in all locations. Since inventory can only be used to satisfy demand when it is available to the assembly line as finished goods, this leads to a configuration where all basestock levels except the last are zero. Otherwise, inventory would be intentionally held in one or more buffers internal to the system, and there is no incentive in our formulation to do so. We will use this observation in designing our experiments.

2.2.3 CONWIP

CONWIP control strives to maintain a constant work in progress (Spearman, Woodruff, and Hopp 1990). When the preset WIP level is reached, no new jobs are authorized for release to the system before some job leaves. This occurs in response to demand events. A CONWIP line can be seen as controlled by a single kanban cell encompassing all machines. Demand that arrives when the system is full is usually kept on a backlog. If no backlog is allowed in either case, the CONWIP policy becomes equivalent to a basestock policy where only the basestock level for finished goods is non-zero.

If a machine fails in a CONWIP line, the amount of material downstream of it will be eventually be flushed out of the system by the demand process. These demand events will cause the release of new jobs to the system. If the machine stays down long enough, these jobs and the jobs already in the system upstream of the failed machine will accumulate in the buffer immediately upstream of this machine. The release of new jobs to the system will then stop.

CONWIP can be implemented by associating a single card with each part, authorizing its presence in the system. Whenever a part leaves the finished goods buffer, its card is detached and sent to the first production stage, authorizing another part to enter the system. All other stages are always authorized to work on any part released to the system, so passing cards to these machines is not necessary.

2.2.4 Two-boundary hybrid

In some cases, the local inventory build-up in basestock and CONWIP control is excessive. For example, if some machine is a capacity bottleneck, any inventory build-up in front of it will remain in the system for a long time (Veatch and Wein 1994). If the upstream machines are relatively fast and reliable, we may choose to limit the inventory build-up before the maximal level is reached. This is even more important in systems where backlogging is permitted. We therefore also investigate systems that are hybrids of base-

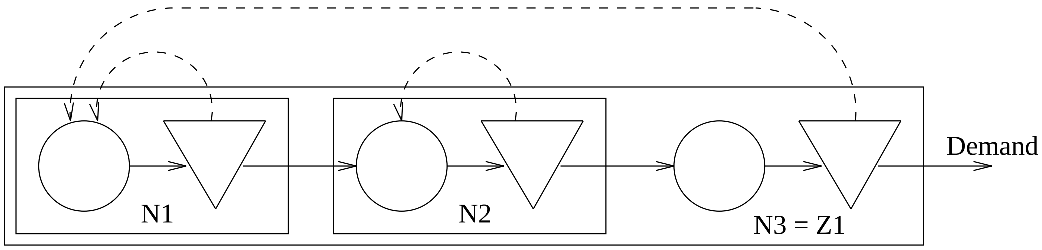

Figure 4: A production line controlled by a kanban-CONWIP hybrid.

stock and kanban control, that is, where demand information is propagated directly as in basestock control, but where there also are inventory limits as in kanban control. Numerical experiments have shown that these control policies are close to optimal for two-stage lines (Van Ryzin, Lou, and Gershwin 1993).

In the system we study, all basestock levels except the last will be zero, as noted in section 2.2.2. The size of the finished goods buffer must be at least as large as the basestock level of the last machine to permit that machine to attain its production target. This makes the hybrid policy particularly easy to implement as a modification of kanban control. Instead of passing kanban from the finished goods buffer to the last production stage when parts are used, send them instead to the *first* stage to authorize loading of another part into the system. This kanban will then follow the part all the way through the system, while other kanban recirculate as usual to limit local inventory accumulations. No separate kanban is needed for the synchronization of the last production stage with the assembly line, since the amount of material in the entire line can never exceed the inventory allowed in this buffer. Equivalently, this control policy can be created from a CONWIP control by limiting inventory levels at the intermediate locations. This is illustrated in figure 4.

3 Results

3.1 Performance measures

One very important performance measure in the factory is the *service level* (or fill rate) at the assembly line. This is the fraction of all demands that find a component ready for use when the demand occurs.

Another important performance measure is the amount of *inventory* in the system. We define the inventory as the amount of material that has been loaded on the first machine, but has not yet been delivered to satisfy demand. We do not consider parts that are authorized for loading at the first machine inventory until they are actually loaded. In the following, we study the relationships between service level and inventory for the different control policies. We do this both for a constant demand rate, and for a case where the demand rate changes over time.

Some influential previous papers, e.g., Tabe, Muramatsu, and Tanaka (1980), use the *variability amplification* of production ordering quantity along the line as their main performance measure. They defined this to be the ratio of order size variance at stage i divided by that of stage $i-1$. Measures like these are important in deciding how long production lines to use when designing a factory, and also indicates what load the system will impose on upstream suppliers. Our work assumes fixed ordering quantities of one, but does not fix the reordering intervals. That is, one unit of material is loaded into the machine whenever the machine is authorized to do this and material is available. Thus, a related performance measure is the variability of the intervals between successive loading events at each machine.

3.2 Constant demand rate

Our starting point is a four-machine line feeding parts to the assembly line. The assembly line is modelled as a deterministic demand process pulling one part per unit of time out of the finished goods buffer. If there is no part, the line stops, and will not restart before parts have been made. The production rate is then equal to the fraction of requests from the line that were satisfied, i.e., the service level.

The line we simulated has operation times with mean 0.98 minutes and a standard deviation of 0.02 minutes. We modelled this by a lognormal distribution with parameters -0.02 and 0.02. A random variable X is said to be lognormally distributed with parameters μ and σ if $X = \exp(Y)$, where the random variable Y is normally distributed with mean μ and standard deviation σ (Rubinstein 1981). We chose the lognormal distribution because it can be sampled very efficiently and can represent low-variability distributions without numerical instabilities. The failure and repair distributions are assumed to be exponential with mean time to fail 1,000 minutes and mean time to repair 3 minutes. This gives each machine an isolated production rate of 1.01695. Even if machine failures are rare in this system, they create an important source of variability when the utilization is this high.

If our system did not have any variability, it would operate at a perfect service level of 1.0 with an inventory of 4 machines $*$ .98 minutes operation time $*$ production rate 1.0 parts per minute = 3.92 parts. In this ideal case, material is only in the system when it is being worked on by a machine. We expect different control policies to have different characteristics with respect to the relationship between throughput and service level in realistic systems with variability.

Our first experiment was to enumerate and simulate all kanban configurations with buffer space less than 5 for buffers 1, 2, and 3, and with buffer space less than 10 for finished goods. This gave 1,250 cases. The reasoning for choosing a larger last buffer is that inventory is not available to satisfy demand unless it is in that location, so we want to keep most of the WIP there.

We did the same experiment for the minimal blocking policy, for another 1,250 cases. Next we ran the same line under CONWIP control, trying all inventory limits in the range 1 to 25. The upper limit corresponds to the total buffer space in the largest kanban configuration we tried.

Another plausible choice is basestock control where the basestock levels reflect the time spent in each production stage times the production rate. For our system, this will be basestock levels of 1, since each part will spend .98 time units in each stage on the average and we aim for a production rate of 1.00. Each part will then on the average spend just .02 units of time in each buffer before entering the next stage. One may surmise that the basestock control will lead to a smoother flow of material than CONWIP, but perhaps at increased inventory levels. We tried this scheme with the same range of overall inventory limits as in CONWIP.

Finally, since we know that basestock and CONWIP can accumulate excessive inventory, we also tried the hybrid policy with limits on total inventory and individual buffer levels. We again enumerated all buffer configurations in the range 1 through 5 for internal buffers, but let the finished goods buffer take on values up to 25. In all cases, each machine except the last had a basestock level of zero, and the last machine had a basestock level equal to the size of the finished goods buffer, indicating that that buffer is the only place where significant inventory is allowed to form. This gave another 3,000 cases to be simulated.

Each case was run for 240,000 minutes of simulated time with an initial warm-up period of 9,600 minutes, starting each one from a different random

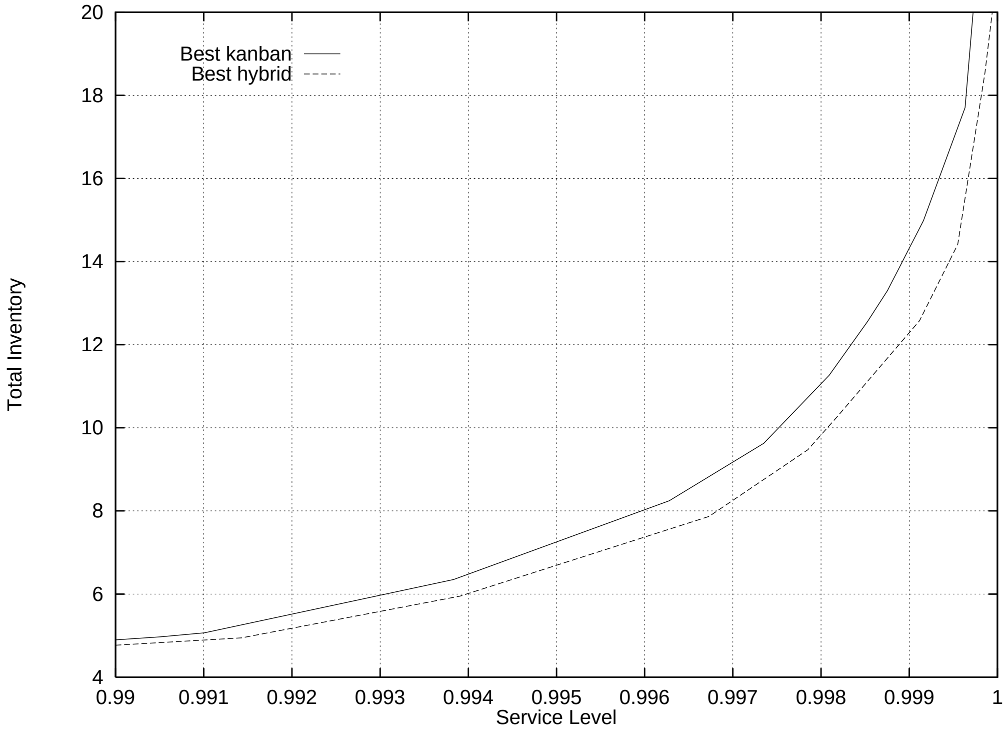

Figure 5: The tradeoff between service level and inventory

number seed.

3.2.1 Inventory results

In figure 5, we have plotted inventory against service level for kanban and hybrid control. Good parameter choices are those that attain high throughput at low inventories, so for each simulated case, we noted this pair of performance measures. The curves in the figure are the convex hulls of all throughput-inventory pairs obtained for the kanban and hybrid policies. Each point on these curves corresponds to the parameter choice within a particular policy that achieved that service level with the least inventory. The preferred polices at all service levels had a relatively large finished goods buffer and small internal buffers.

The hybrid policy makes a better service-inventory compromise than kanban at all service levels. The vertical distance between the curves is the difference in inventory required to attain a particular service level. This dif-

Policy	Description	Buffer sizes				Basestock levels			
Kanban	Finite buffers, blocking before service	2	2	4	10	∞	∞	∞	∞
Min. block.	Movable buffers	2	2	4	9	∞	∞	∞	∞
Basestock	Buffer target levels, infinite buffers	∞	∞	∞	∞	1	1	1	12
CONWIP	Single inventory limit, infinite buffers	∞	∞	∞	∞	0	0	0	15
Hybrid	Single inventory limit, finite buffers	2	3	5	15	0	0	0	15

Table 1: Best configurations that achieve the stated objective

Policy	Service level	Inventory
Kanban	$0.99916 \pm .00006$	$15.82 \pm .05$
Minimal blocking	$0.99909 \pm .00005$	$15.62 \pm .05$
Basestock	$0.99918 \pm .00006$	$14.60 \pm .02$
CONWIP	$0.99922 \pm .00005$	$14.59 \pm .02$
Hybrid	$0.99907 \pm .00007$	$13.93 \pm .03$

Table 2: Performance of best configurations

ference becomes larger as the required service level increases. This graph is an important result: It says that a small change in the flow of information can produce a substantial reduction in inventories compared to kanban control, which is currently considered the state of the art in lean manufacturing.

The other control policies we tested generally fall between the kanban and hybrid curves in this plot, but sometimes exceed the inventory level of kanban at the same throughput. None of the policies have lower inventories than the hybrid policy at any service level.

To investigate this further, suppose we get the following objective: *Find a configuration that gives a 99.9% service level using the least possible total inventory.* This represents the task of finding the best way, by modifying information flow and control parameters, to achieve a management-set service target. We selected the data points from the previous results that came close to this service level, and did 30 replications with different random number seeds of each one. The run length was again 240,000 minutes with a 9,600 minute warm-up period for each replication.

We found the solutions shown in table 1, choosing configurations so that the entire throughput confidence interval was above .999. For the kanban, minimal blocking, and hybrid policies, multiple configurations achieved the service level target with insignificant differences in inventories. Note that under our assumptions, kanban and CONWIP control become special cases of the hybrid control, while the minimal blocking control has a different blocking model. The only difference between basestock and CONWIP in this table is that we constrained the basestock levels to be non-zero in basestock control, but constrained them to be zero for the internal buffers in CONWIP control. Again, we emphasize that the apparent close relationship between CONWIP and basestock is in part due to our assumption of lost demand. Table 2 shows the performance of the best configurations found of each policy. The 95% confidence intervals were found by the formula

$$\overline{X} \pm 1.960 S/\sqrt{N}$$

where $\overline{X}$ is the sample mean, S the sample standard deviation, and N the sample size. This assumes that the sample is "large", but does not assume any underlying distribution of the sample points (Arnold 1990).

In this case, changing from the best kanban configuration to the best hybrid configuration produced an inventory reduction of 12% in an already lean system. As the curve in figure 5 demonstrated, this advantage will increase to near 20% if the service level target is at 99.95%, but will be smaller if the ambitions are lower. Another important observation that can be made from these simulations is that the average inventory locations of the policies are distinctly different. This is shown in figure 6 where we have plotted the average level of each buffer under the different control policies. Kanban maintains more inventory internal to the line than the other policies, because the kanban policy is explicitly to fill up the buffer whenever possible. This behavior is even more pronounced in the minimal blocking model than in the tandem buffer kanban model. The best basestock and CONWIP policies for this system have almost identical distributions of material along the line.

3.2.2 Variability amplification

We now turn to the amplification of variability in the line. Figure 7 shows the standard deviation of the times between loading events at each machine for the different control policies. These results are taken from the same series

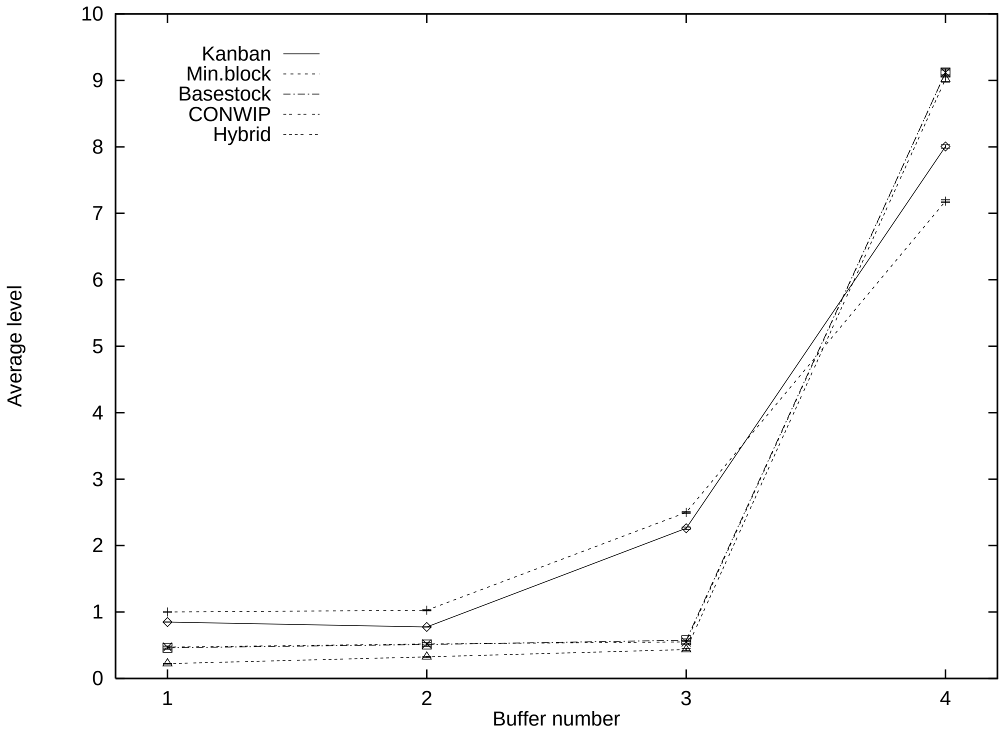

Figure 6: Location of inventory

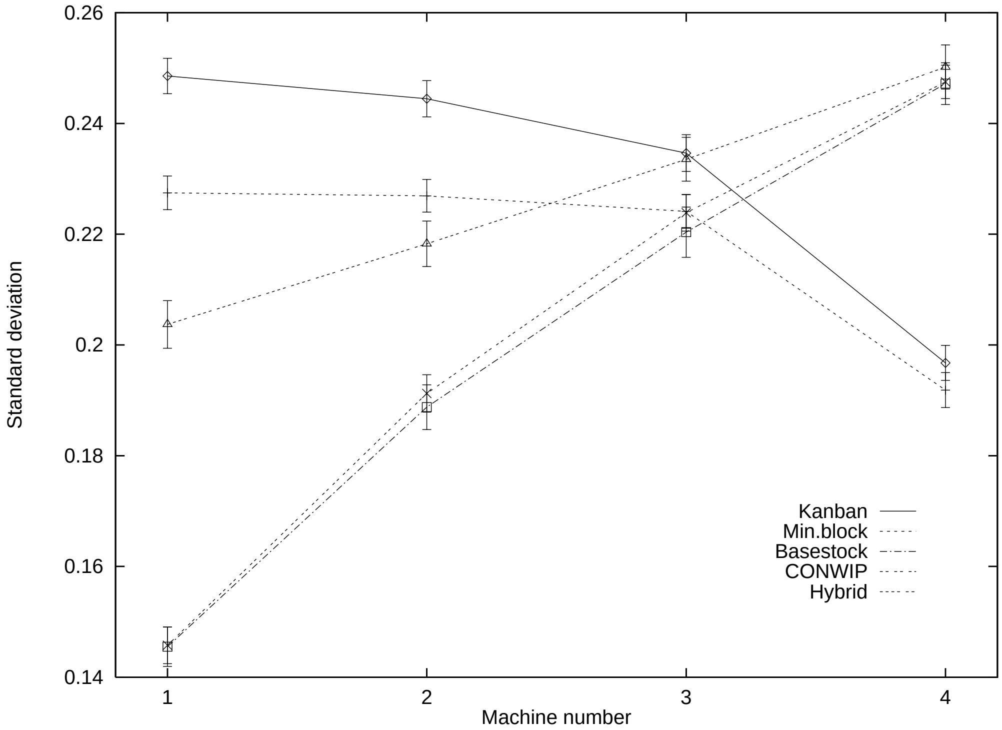

Figure 7: Variability of times between loading events

of experiments as in the previous section, and the configurations of table 1 were used.

Starting from machine 4, it is evident that kanban amplifies the variability as one proceeds upstream. The minimal blocking model has less variability, but about the same amplification as the tandem buffer model.

In contrast, the other policies appear to dampen the variability, although they start from a higher level of stage 4 variability. This is because a kanban controlled machine tries to track the immediate downstream demand process, while adding its own noise to the demand process seen by the next upstream machine. The other policies send demand information directly to the first machine, so the process of introducing material to the line will be much less noisy. The variability of the machines will then add noise as the parts move downstream. The basestock policy gives a slightly smoother flow than CONWIP, with the largest difference in the middle of the line, but the difference does not seem to be significant. In summary, our versions of basestock, CONWIP, and hybrid control all vastly outperform kanban by the same performance yardstick that was used by Tabe, Muramatsu, and Tanaka as well as Kimura and Terada to argue the merits of pull control over push.

3.3 Changing demand rate

Since this production line is quite reliable, a more important consideration than the response to machine failures is the response to adjustments in the demand rate. Such adjustments are done occasionally at Toyota to reflect changes in market conditions. Our experiment was to run the line for 2,400 minutes at a rate of 1 minute between jobs, then increase the time between jobs to 1.25 minutes for another period of 2,400 minutes, and go back to the 1 minute cycle for the last 2,400 minutes of the experiment. This run was prefixed by a warm-up period of 1,200 minutes, where the line was running at 1 job per minute. We extracted the buffer level trajectories for each buffer and for total inventory as a function of time for each run of the experiment. Then we averaged these trajectories over 50 runs, each one with a different seed to the pseudo-random number generator. The resulting data were aggregated by averaging over time windows of 60 minutes to remove high frequency fluctuations that were irrelevant to our study. We did this experiment with the best kanban, CONWIP, and hybrid configuration from the previous section without any adjustment to the parameters to reflect the changed conditions.

Figure 8 shows the results. The topmost three lines are the total inventory levels (WIP) for kanban (using the tandem buffer model), CONWIP, and hybrid as functions of time. Kanban has a higher total inventory at all times, but the difference grows much larger in the period of reduced demand. We note that the inventory levels are more variable in the periods of high demand than in the periods of low demand, due to less excess capacity to recover from random fluctuations. It also appears that kanban has more variable inventory levels than the other policies.

The next group of three lines shows the finished goods inventory (FG) over time for the three policies. For this group, the order is reversed from the total inventory case. Kanban has less finished goods inventory than the other policies, and its level increases less sharply when the demand decreases.

The last group of lines show the level of buffer 3 (B3). Note that kanban maintains a much higher inventory in this internal buffer than the other two policies. When the demand is decreased, CONWIP and hybrid control reduce

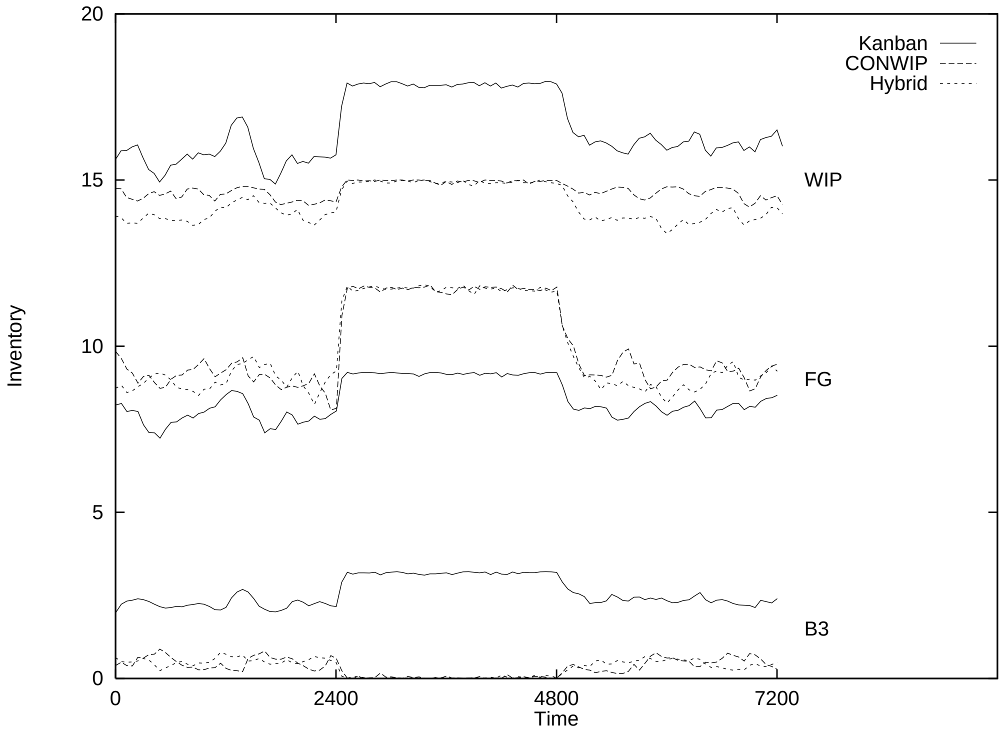

Figure 8: Inventory trajectories with changing demand

the inventory in the internal buffers almost to zero, but kanban *increases* its internal buffer levels. The behavior of buffers 1 and 2 are similar to that of buffer 3, but at lower levels, as we have already shown in figure 6.

4 Discussion

A kanban line ensures the overall production rate by making a few parts ahead of time and storing them in the buffers. These parts can then be used to let downstream machines continue working even if there is a disruption *upstream*. The kanban numbers regulate the size of these buffers, and the production rate is regulated by buffers filling up and blocking production. The hybrid policy, on the other hand, regulates the production rate by holding the release rate to the system equal to the demand rate. This keeps the internal buffers mostly empty. The empty space is then used to let upstream machines continue working, filling up this space, even if there is a disruption *downstream*. This achieves the same boost in production rate by partially decoupling the machines as if parts were stockpiled, but without the inventory cost.

Another important reason for the performance gap between these policies is the difference in information distribution. In kanban control, information flows only when material is moved. This is somewhat relaxed in the minimal blocking policy, but the information flow is still interrupted by the first empty buffer. The other policies distribute demand information directly to all machines whenever demands occur. This information passing is explicit in basestock control and implicit in CONWIP and hybrid control, where the demand information is sent to the first machine only. Decoupling the information flow from the part movements gives added stability against disruptions. This becomes increasingly important as the utilization approaches capacity. In our level schedule experiments, the effect of the hybrid policy as shown in figure 5 can be compared to entirely eliminating operation time variability or doubling machine reliability. In contrast to these potentially expensive ways of improving the system, changing the movement of kanban achieves the same benefit at essentially no cost.

In this study, there is only a small difference between the two most common models of a kanban system, represented by the tandem buffer kanban policy and the minimal blocking policy. The minimal blocking policy has higher throughput and higher inventories than the fixed buffer kanban policy for the same buffer sizes. For other systems, the difference could be larger, but the throughput advantage of the minimal blocking policy might not be enough to compensate for the increased inventories. For instance, if we set buffer sizes of 2, 2, 4, and 10 along the line, the tandem buffer kanban policy has an average inventory of 15.82 and the minimal blocking policy 16.53, a difference of 4.5% in favor of tandem buffers. At the same time, the tandem buffer policy has a throughput of .99916 and the minimal blocking policy has a throughput of .99925. This is only a .009% difference.

When the demand rate changes, lines controlled by kanban will fill or empty their buffers, and the change in demand rate will propagate upstream as blockages or starvations. The other policies change the material release rate to the system instantly as demand changes. Of course, no manager of a kanban line would allow extensive starvations or excessive inventories to occur as a consequence of a planned demand change. Instead she would change the numbers of kanban in circulation just before the demand rate is scheduled to change, so that the system response is improved. This would require much thought and adjustment to every kanban cell in the system.

In contrast, a line controlled by the hybrid policy will have the consequences of a demand rate change localized to the finished goods buffer, as demonstrated by the trajectories in figure 8. This can then be counteracted more easily by changing the single WIP limit for the system. If the demand change is severe, the intermediate buffer sizes may need to be adjusted, but the consequences of maladjustments are small. For example, if all buffer sizes in a hybrid line are allowed to be infinite, the resulting control policy will be CONWIP. This policy consistently comes in a close second best in our study.

5 Conclusion

We have studied the performance of kanban, minimal blocking, basestock, CONWIP, and hybrid control in a simulation of a short flow line making a single part type. The model was based on an actual system in a Toyota assembly factory. The cases we considered included both constant and time-varying demand rates. The hybrid control policy demonstrated superior performance in achieving a high service level target with minimal inventories, closely followed by CONWIP and basestock. The hybrid policy we studied can be implemented as a straightforward modification to a kanban policy, simply by routing kanban from the finished goods buffer to the first production stage instead of the last. The hybrid control policy will allow continuing productivity and service level improvements beyond what is attainable with kanban control alone, while making the production system more responsive to production rate adjustments.

Acknowledgements

We thank Michael H. Veatch and James E. Schor for valuable input and interesting conversations. We are also grateful to two anonymous reviewers for helpful suggestions that improved the presentation of the paper.

References

Arnold, S. F. (1990). *Mathematical Statistics.* Englewood Cliffs, New Jersey: Prentice Hall.

Berkley, B. J. (1991). Tandem queues and kanban-controlled lines. *International Journal of Production Research 29*(10), 2057–2081.

Berkley, B. J. (1992). A review of the kanban production control research literature. *Production and Operations Management 1*(4), 393–411.

Bielecki, T. and P. R. Kumar (1988, Jul-Aug). Optimality of zero-inventory policies for unreliable manufacturing systems. *Operations Research 36*(4), 532–541.

Buzacott, J. A. and L. E. Hanifin (1978). Models of automatic transfer lines with inventory banks — a review and comparison. *AIIE Transactions 10*(2), 197–207.

Buzacott, J. A. and J. G. Shantikumar (1992). A general approach for coordinating production in multiple-cell manufacturing systems. *Production and Operations Management 1*(1), 34–52.

Clark, A. J. and H. Scarf (1960, Jul). Optimal policies for the multi-echelon inventory problem. *Management Science 6*(4), 475–490.

Dallery, Y. and S. B. Gershwin (1992, Dec). Manufacturing flow line systems: A review of models and analytical results. *Queuing Systems Theory and Applications 12*(1-2), 3–94. Special issue on queuing models of manufacturing systems.

Di Mascolo, M., Y. Frein, and Y. Dallery (1994). An analytical method for performance evaluation of kanban controlled production systems. To appear in *Operations Research*, special issue on New Directions for Operations Management Research.

Frein, Y., M. Di Mascolo, and Y. Dallery (1994). On the design of generalized kanban control systems. To appear in *International Journal of Operations and Production Management*, special issue on Modelling and Analysis of Just-in-Time Manufacturing Systems.

Gershwin, S. B. (1994). *Manufacturing Systems Engineering*. Englewood Cliffs, New Jersey: PTR Prentice Hall.

Kimball, G. (1988). General principles of inventory control. *Journal of Manufacturing and Operations Management 1*(1), 119–130.

Kimemia, J. and S. B. Gershwin (1983, Dec). An algorithm for the computer control of a flexible manufacturing system. *IIE Transactions 15*(4), 353–362.

Kimura, O. and H. Terada (1981). Design and analysis of pull system, a method of multi-stage production control. *International Journal of Production Research 19*(3), 241–253.

Lee, Y.-J. and P. Zipkin (1992, Sep-Oct). Tandem queues with planned inventories. *Operations Research 40*(5), 936–947.

Mitra, D. and I. Mitrani (1990, Dec). Analysis of a kanban discipline for cell coordination in production lines, I. *Management Science 36*(12), 1548–1566.

Mitra, D. and I. Mitrani (1991, Sep-Oct). Analysis of a kanban discipline for cell coordination in production lines, II: Stochastic demands. *Operations Research 39*(5), 807–823.

Rubinstein, R. Y. (1981). *Simulation and the Monte Carlo method*. New York: John Wiley and Sons.

So, K. C. and S. C. Pinault (1988, Dec). Allocating buffer storages in a pull system. *International Journal of Production Research 15*(12), 1959–1980.

Spearman, M. L., D. L. Woodruff, and W. J. Hopp (1990). CONWIP : a pull alternative to kanban. *International Journal of Production Research 28*(5), 879–894.

Sugimori, Y., K. Kusunoki, F. Cho, and S. Uchikawa (1977). Toyota production system and kanban system materialization of just-in-time and respect-for-human system. *International Journal of Production Research 15*(6), 553–564.

Tabe, T., R. Muramatsu, and Y. Tanaka (1980). Analysis of production ordering quantities and inventory variations in a multi-stage production ordering system. *International Journal of Production Research 18*(2), 245–257.

Van Ryzin, G., S. X. C. Lou, and S. B. Gershwin (1993, Sep). Production control for a tandem two-machine system. *IIE Transactions 25*(5), 5–20.

Veatch, M. H. and L. M. Wein (1994, Mar-Apr). Optimal control of a two-station tandem production/inventory system. *Operations Research 42*(2), 337–350.

Wein, L. M. (1988, Aug). Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing 1*(3), 115–130.

Yan, H., S. Lou, S. Sethi, A. Gardel, and P. Deosthali (1994). Testing the robustness of various production control policies in semiconductor manufacturing. Forthcoming in *IEEE Transactions on Semiconductor Manufacturing.*