5-1997

Modeling and Simulation of Manufacturing Systems Under Signal Kanban Policies

Lawrence E. Holloway
University of Kentucky, larry.holloway@uky.edu

Presented at the 2nd International Symposium on Scale Modeling, Lexington, Kentucky, May 1997

MODELING AND SIMULATION OF MANUFACTURING SYSTEMS UNDER SIGNAL KANBAN POLICIES

L. E. Holloway
Center for Manufacturing Systems and
Dept. of Electrical Engineering
University of Kentucky
Lexington, KY 40506-0108
email: holloway@engr.uky.edu

ABSTRACT

A signal kanban system is used to order the production of various parts by processes with lengthy setup times, such as in metal stamping or injection molding. In this paper, we explore the behavior of signal kanban systems using simulations of systems producing two part types. Such simple models allows a visualization of the changes in the inventories of these parts under the signal kanban system. We consider the operation of the common fixed-batch signal kanban system, and a variation that we refer to as a fixed-fill policy. These policies are investigated under variations in demand, occasional disruptions in production, and modifications in policy parameters. Our investigations for this two-part-production model indicate that a fixed-fill signal kanban system is significantly more robust than the more common fixed-batch kanban system.

INTRODUCTION

The term *kanban* refers to a class of demand-driven production control techniques which are commonly used by companies practicing "Just-in-time" or "Lean Manufacturing" production [1][2]. Kanban is a classic example of a "pull" production technique, because orders for parts are generated by down-stream processes that are consuming them. One advantage of a kanban system is its ability to constrain inventory of completed parts, since additional production is only authorized by the consumption of parts at down-stream stations.

Our investigation focuses on a type of kanban system referred to as "signal-kanban." In a signal kanban system, there is only one authorization. Signal kanban systems are used in manufacturing systems with lengthy setups, where it may not be practical to make small batches [1]. In a signal kanban system, there is a single production authorization per part type. It is associated with a fixed level of stock which is a reorder point for that part --

thus whenever the inventory falls below the level of the signal, the signal is released, authorizing additional production of that part. This authorization enters a queue with signal kanbans for other parts that are awaiting production. In the most common implementation of a signal kanban system, when the producing machine gets to the authorization for a particular part, it will produce a fixed number of that part. We refer to this as a *fixed-batch* policy, since the production run for a part is always for the same number of parts. Upon the machine completing that production authorization, the authorization signal is then replaced at the reorder point for that part. This fixed-batch policy is an example of a fixed-quantity reorder system, which is a common method for inventory management [3].

Our investigation of signal kanban systems is motivated by discussions with personnel at the Toyota Motor Manufacturing (TMMK) plant in Georgetown, Kentucky. The stamping operations at TMMK had previously used a signal kanban system with fixed production batch sizes. The personnel had noted that some times, the subsequent operation (welding) would run out of a part. The problem arose when several parts had their signals released close in time, such that the machine was busy producing earlier orders while other parts ran out. However, at other times, there may be no signals authorizing production, so the machine sat idle. In this paper, we examine signal kanban systems and consider why these parts shortages occurred. We develop models for the fixed-batch signal kanban system and for a variation we refer to as fixed-fill kanban system. Each of these considers a single machine which produces two different parts. These models are then simulated under fixed and variable demand. By considering only two parts, we can plot the change in inventory of the parts over time and easily visualize the effect of the different policies.

In [4], it was shown that several variants of the signal kanban system, including the fixed-batch and fixed-fill policies, will be guaranteed to not have parts shortages if a capacity constraint is met and if the signal levels are set high enough. The sufficient signal levels to ensure this were also given, but they are not necessarily minimum. The inventory in signal kanban systems, as illustrated in this paper, can follow a periodic cycle. These cycles depend on several factors, including the initial state, the buffer level, and the relative ratios of consumption and production rates for the parts. By selecting the initial state, the buffer level, and the ratios of consumption and production rates, it may be possible to achieve a periodic cycle that needs only small signal levels. We explore this in this paper for fixed-batch and fixed-fill signal kanban systems, and we consider their operation under disruptions in production rates and variation in demand.

The models we consider were developed as functions in MATLAB, and ran on both a Pentium PC and a SparcStation 10. Each model considered a single machine responsible for production of two parts, denoted part x and part y. In the next section, we describe the model and the simulations for the fixed-batch signal kanban system. Section three presents the model and simulation for the modified policy we refer to as a fixed-fill policy. Finally, in section four we summarize the results and discuss future research directions.

FIXED-BATCH SIGNAL KANBAN MODELS

The signal kanban system that was in use in the Toyota stamping operation used fixed production batch sizes. Whenever the production of a part began, a fixed number of those parts were produced. The signal levels and the batch sizes for each are intended to have identical proportions with respect to daily demand for the part. Thus, for example, the signal level for each part might be set to correspond to the number of parts typically consumed in one third of a shift, and the batch size of all parts might be set to correspond to 1.5 shifts worth of consumption, for example.

The function written in Matlab to simulate a fixed-batch kanban system is *rfixbatch*. The arguments are:

```
rfixbatch(x0,y0,d0,[rx,sx,ry,sy],[trigx,trigy],][batchx,batchy],
      [rangex,rangey],[probfreq,probeffect],xsetup,n)
```

The function takes an initial inventory `x0`, `y0` of product x and product y, and considers the change in inventory (x and y) over `n` steps. The term step indicates a period when the machine is producing x, a period when it is producing y, or a period when it is idle. The machine is either initially idle (`d0` = 0), producing product x (`d0` = 1), or producing product y (`d0`=2). The nominal consumption rate of product x is `rx`, and the production rate of x is `sx`. Similarly, `ry` and `sy` are the nominal consumption rate and production rate of product y.

The range of variation in demand is specified by the vector [`rangex,rangey`]. The parameter `rangex` specifies the allowable +/- percentage of fluctuation permitted in the demand rate (consumption rate) for x, and similarly for `rangey` for the fluctuation in demand of y. For each step of the simulation, a random value chosen uniformly between the +/- range is chosen for each part, x and y. Drastic drops in production can be permitted using the vector [`probfreq,probeffect`]. The parameter `probeffect` indicates the percentage of drop in the production rate that occurs upon a disruption. The dissruptions occur randomly acccording to the frequency `probfreq` (expressed as a percentage) that is specified.

The signal levels for products x and y are given by the parameters `trigx` and `trigy`, respectively. When the inventory of a product drops below its trigger level, a request for production of that product is generated. If the machine is currently producing another product, then the new request does not interrupt the production, but waits until the current production has been completed. Once production of a part has begun, a fixed quantity of those parts are produced. The value of this batch size is `batchx` for product x and `batchy` for product y.

The machine goes through a setup whenever production of a part is to begin. The relative time delay associated with a setup is specified by the parameter `xsetup`. This parameter specifies how much of x is consumed during a setup, when no product is being produced.

Because the demand for product y is known relative to the demand for product x, the simulation can easily determine the amount of y that is also consumed during a setup.

Representative results of the simulations are shown in figure 1. Figure 1a plots the inventory of x versus the inventory of y over 100 steps. Both parts have identical ratios of consumption to production, as well as identical trigger level and batch size with respect to the consumption rates. Variation in demand is a low +/-1% for both products, and there will be no disruptions in production (`probfreq` = 0). As can be seen from the figure, the production cycles between idle (including setup for x), production of x, idle again (including setup for y), and finally production of y before the cycle repeats again. This cycle each time retraces almost the same path in the plane of x and y inventories.

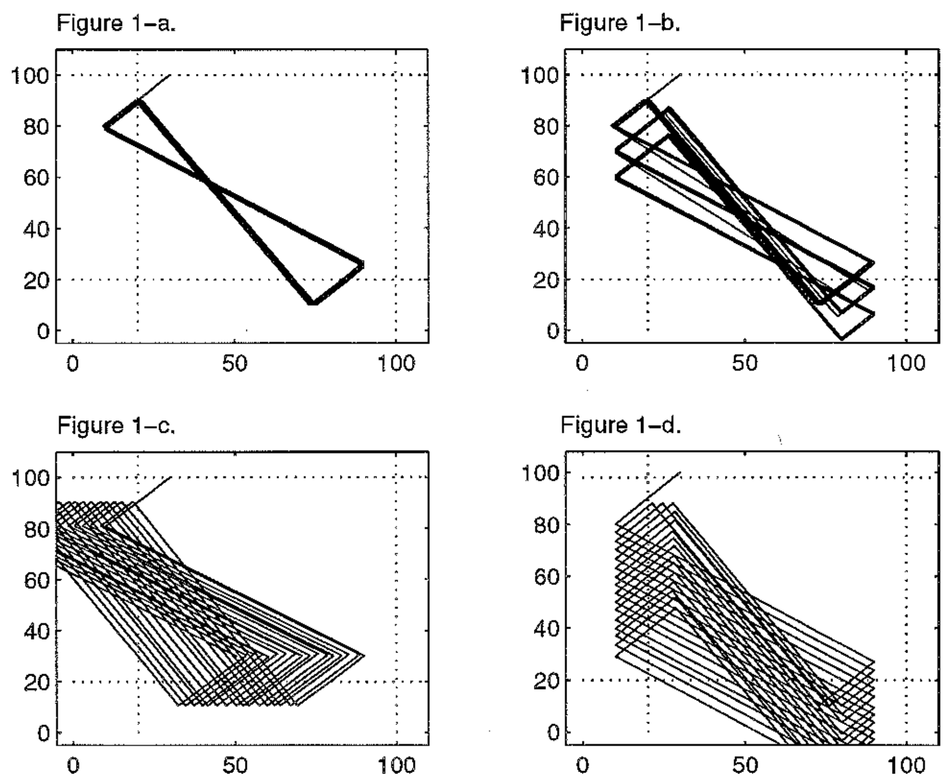

Figure 1 (a) rfixbatch(30,100,0,[4,10,4,10],[20,20],[80,80],[1,1],[0,0],10,100)
(b) rfixbatch(30,100,0,[4,10,4,10],[20,20],[80,80],[1,1],[3,10],10,100)
(c) rfixbatch(30,100,0,[4,10,3.8,10],[20,20],[80,80],[1,1],[0,0],10,50)
(d) rfixbatch(30,100,0,[4,10,4,10],[20,20],[80,78],[1,1],[0,0],10,50)

In figure 1b, we show almost the same situation, but where we permit a "problem" to occur in production. Three percent of the time a 10% drop in production could occur. In the run shown, three disruptions occur, each during the production of part x. As shown in the figure, each occurrence of such a problem drops the cycling of x and y inventory into a lower orbit until finally a parts shortage occurs in product y in the lowest orbit. It is important to note that inventory levels never recover to their original orbit.

Figures 1c and 1d also indicate how fragile this form of kanban production is. Our simulations seem to indicate that achieving a repeated orbit in the (x,y) plane requires that the ratios `rx/sx` and `ry/sy` must be integer multiples of each other. For example, vector `[rx,sx,ry,sy]` = [6,11,3,11] or [4,10,8,20] results in a repeated orbit. In figure 1c, however, a consistently lower demand of 3.8 of y per time unit compared to 4 of x per time unit leads to a long-term spiraling down of inventories. If the signal levels are not set high enough, then this spiraling down may lead to an eventual parts shortage. Note that this is less demand than in the previous cases, so this is not due to any increased utilization. Figure 1d shows that a slight imbalance in the batch sizes (80 of x, vs. 78 of y) with respect to the demand and production rates also leads to a similar spiraling down of the inventory towards a parts shortage.

It should be noted that the spiraling down of the inventories in the simulations of figure 1c and 1d will not continue indefinitely. If the inventories move low enough, then there will always be a production request waiting for service. This eliminates any idle time the system may have been exhibiting during the spiraling down period. The higher utilization of the machine results in a spiraling upward of the inventories, or at least prevents any further drop. If the ratios of demand rate to supply rate between the two products are related by a rational number, then the situation of figure 1c will eventually have the inventories return to close to their initial states, before starting to spiral down again. The result is a repeated pattern of spiraling down to a minimum, and then spiraling back up again. The time period for the repetition of this activity may be very lengthy.

Figure 2 shows the spiraling down and then spiraling up again for the case of mismatched demand proportions: x still has a demand/supply rate ratio of 4/10, but y has a ratio of 3.5/10. Initially, the spiraling similar to that seen in figure 1c occurs. In this situation, production of y is always followed immediately by the production of x (excepting setup). In each production of y, the signal level of x is crossed earlier and earlier during the production. The longer and longer delay between the crossing of the signal level of x and the commencement of production of x means that the level of x upon completion of a run of x becomes smaller and smaller. Eventually, the inventory level upon completion of production of x is very close to the signal level of x. In this case, the x signal level is crossed before the y signal level is crossed, leading to a second consecutive run of x. This flips the cycles from running large deficits of x to running large deficits of y. In this situation, the production of x is followed by the immediate setup and production of y. Since the ratio between demand and production rates of y is smaller than the ratio for x, then each time a production run of x is completed, the inventory of y is higher than after the previous completion. Eventually, the signal level for y is not crossed during the x production, and we eventually return to operation above the signal levels, at which time the inventory will again begin to spiral down.

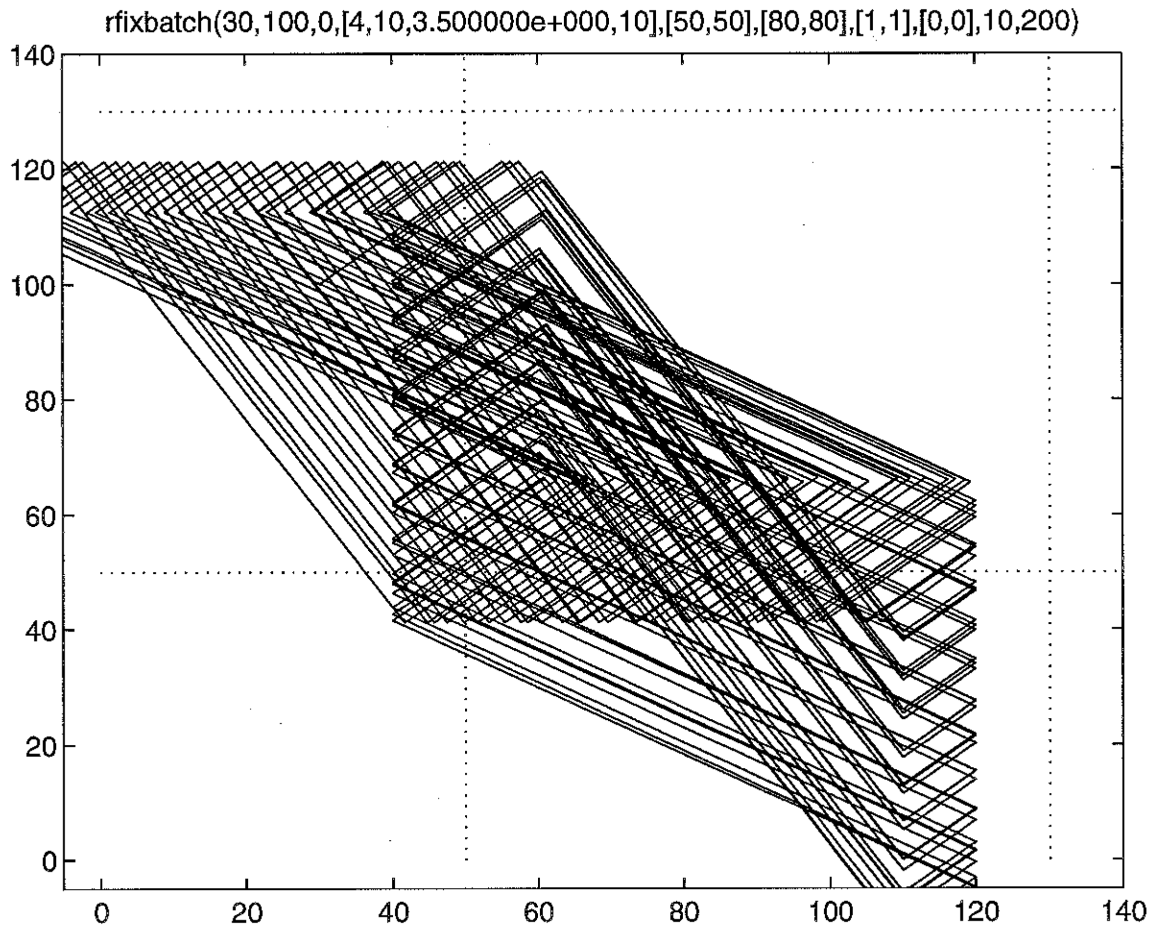

Figure 2 The long-term behavior due to mismatched relative demand rates. The inventories spiral down to a minimum and then spiral back up to operation above the signal levels.

The long-term recovery of the system illustrated in figure 2 was for the case of mismatched demand/production rate ratios. Similar long-term recovery behavior is also seen in the case of production disruptions (discussed for figure 1b) and in the case of mismatched batch-sizes (discussed for figure 1d). In each case, the spiraling down of inventories (as in mismatched batches or mismatched demand/production rate ratios) or the stepping down of inventories (as in the case of disruptions) may occur over a lengthy period of time. During this time, the system may appear to be operating well and the signal levels may appear to be unnecessarily high, but in fact the system may be moving towards lower cycling of parts inventories in the long-term. If the production operation is undergoing *continuous improvement* or *kaizen* activities to reduce inventories, a decision may be made to reduce the signal levels, which in the long-term may lead to a shortage of parts as the pattern of spiraling down continues. Such decisions might be made using observations of the system over a relatively short period (days or weeks), in which case the long-term spiraling down trend may not be evident.

FIXED-FILL KANBAN SYSTEMS

A simple variation of the fixed-batch policies are the fixed-fill policies. In these situations, the production authorizations come from inventory dropping below a given signal level, just as with the fixed-batch policies. However, once production is begun for a given part, it continues until a fixed buffer level is achieved. This means that the batch size may be smaller if the signal was acted upon immediately and production started shortly thereafter, and the batch size may be larger if the signal occurred while another part was in production when the signal was released. This larger batch size results from the continued lowering of the inventory of the part while waiting for the signal to be serviced. Note that the implementation of the fixed-fill requires more information flow than the fixed-batch policy. The fixed-fill policy requires the machine to have information on the level of inventory to know when to complete its production, whereas the fixed-batch makes a given amount of product when authorized and requires no additional information on current buffer levels while producing.

Our Matlab function written for simulating fixed-fill policies is `rfixfill`. Its command-line call is:

```
rfixfill(x0,y0,d0,[rx,sx,ry,sy],[trigx,trigy],[fullx,fully],
      [rangex,rangey],[probfreq,probeffect],xsetup,n)
```

The arguments for `rfixfill` are similar to those for `rfixbatch`, except that instead of giving batch sizes, the parameters `fullx` and `fully` indicate the fill levels for the two products. Whenever there is production of x, it will continue until the inventory of x reaches `fullx`, and production of y will always continue until its inventory reaches `fully`. Thus, the length of each production run depends on the amount of product consumed since the previous run. Figure 3a illustrates the typical operation of the model when there is very little variation in demand. As can be seen, after only one production of x and one production of y, the system settles into a repeated orbit and stays there. (The initial inventory level of x of 40, versus 30 under the fixed-batch policy, was chosen to illustrate that the eventual orbit of the inventories does not necessarily include the original point.)

Figure 3b shows that the fixed-fill policy seems very robust against occasional disruptions. Given a 3% chance of a disruption that temporarily drops the production rate of a batch by 10%, it can be seen from the simulations that the system recovers from the disruptions and returns to an acceptable repeating orbit. The resulting orbit may not be exactly the same as before a disruption, but it is above the signal levels (less the amount of consumption occuring during setups). The system has thus recovered to a point where it is ready for later disruptions where parts shortages should not occur. This is in contrast to the fixed-batch policies, where the system does not recover from disruptions until its cycle of operation falls so low as to always have a signal awaiting at the completion of each production run. As discussed in the preceding section, this will result in parts shortages for fixed-batch policies if the signal levels are not high enough.

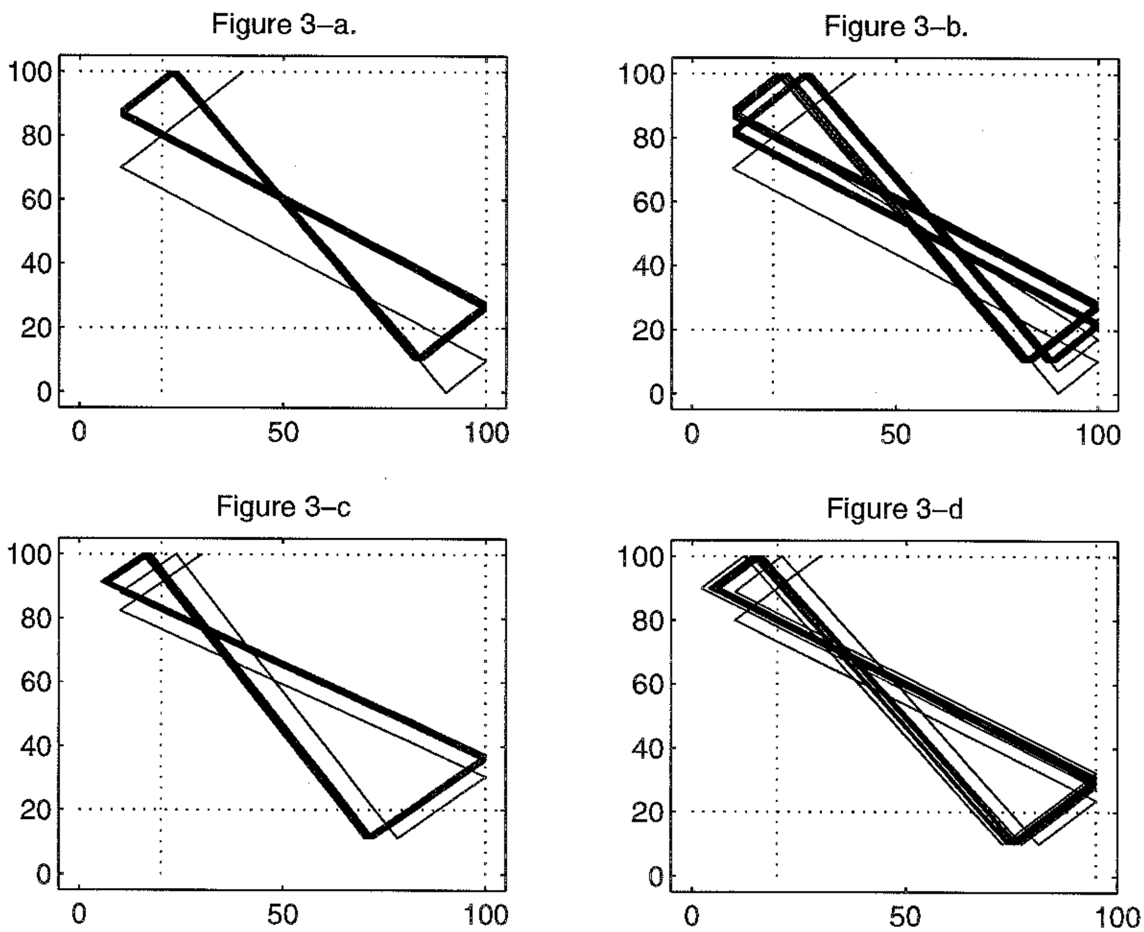

Figure 3 (a) rfixfill(40,100,0,[4,10,4,10],[20,20],[100,100],[1,1],[0,0],10,100)
(b) rfixfill(40,100,0,[4,10,4,10],[20,20],[100,100],[1,1],[3,10],10,100)
(c) rfixfill(30,100,0,[4,10,3.5,10],[20,20],[100,100],[1,1],[0,0],10,50)
(d)rfixfill(40,100,0,[4,10,4,10],[20,20],[95,100],[1,1],[0,0],10,50)

The fixed-fill policy also seems very robust to mismatches in the demand rate between x and y. Figure 3c shows that a consistently lower demand of 3.5 units of y per time period, versus 4.0 units of y per time period for x, will still lead to an acceptable repeated orbit. In the figure, it is seen that for the same initial inventory level, the mismatch results in several more steps before the repeated orbit is reached, but our simulations indicate that an acceptable repeated orbit is consistently reached for a wide range of demand mismatches. In figure 3d, it is seen that the system reaches a repeating orbit even with small mismatches in the fill sizes. In the figure, it is seen that the orbit is very close to having parts-shortages in x, but an increased safety margin can be achieved easily by raising the signal levels. Even with raising the signal levels, the signal levels required to prevent parts shortages in the fixed-fill policy are less than those that would be required in the fixed-batch policy.

SUMMARY

In this paper, we have considered the behavior of signal kanban systems. Simulations were carried out for the example of a single machine producing two part types.

The simulations indicate that the common fixed-batch signal kanban system is very sensitive to minor disruptions in production, choice of batch size, and mismatches of relative rates of consumption among the different parts produced. Such factors determine if the system operates in a repeated cycle of inventory levels with a short period that allows low signal levels (and thus low average inventories), or will operate in a repeated cycle with a long period, in which the inventory levels periodically spiral down toward a low-inventory situation. In this long-period behavior, either there will occur parts shortages, or else the signal levels will have to be set high, resulting in large average inventory levels.

The fixed-fill policy is a variation of the fixed-batch policy, but shows significantly improved robustness. The fixed-fill policy self-adjusts to small variations in rates of consumption, and appears to resynchronize to a short-period cycle following small disruptions in production. This self-adjustment and resynchronization means the system does not exhibit the long-term spiraling down of inventory levels seen under the fixed-batch policy. Thus, parts-shortages can be prevented under the fixed-fill policy using significantly lower signal levels (and thus lower average inventory) compared to the signal levels required to prevent parts-shortages under a fixed-batch policy.

In this paper, we explored only a single machine with two parts. Assuming two parts allowed the behavior of the inventories over time to be easily visualized. Future research will investigate the behavior of the policies when there are more than two parts, and the behavior of the fixed-fill policy when the amount-to-fill is based on delayed inventory information.

ACKNOWLEDGEMENTS

This work has been supported in part by the Center for Manufacturing Systems at the University of Kentucky, Toyota Motor Manufacturing, and ARO grant DAAH04-96-1-399.

REFERENCES

[1] Y. Monden, *Toyota Production System: Second Edition*, Industrial Engineering and Management Press, Norcross, Georgia, 1993

[2] J. P. Womack, D. T. Jones, D. Roos, *The Machine That Changed the World*, Harper Perennial, New York, 1991.

[3] J. Dilworth, *Production and Operations Management, 5th edition*, McGraw Hill, New York, 1993.

[4] T. I. Seidman and L. E. Holloway, "Stability of a 'signal kanban' manufacturing system," in *Proceedings of 1997 American Control Conference,* Albuquerque, New Mexico, USA, June 1997.