Operating high variability manufacturing systems*

Asbjoern M. Bonvik
Operations Research Center
MIT E40-149
77 Massachusetts Avenue
Cambridge, MA 02139-4307

Stanley B. Gershwin
Laboratory for Manufacturing and Productivity
MIT 35-331
77 Massachusetts Avenue
Cambridge, MA 02139-4307

Donald E. Troxel
Department of Electrical Engineering and Computer Science
MIT 36-287
77 Massachusetts Avenue
Cambridge, MA 02139-4307

Abstract— We present two factory control policies derived from optimal control considerations. Using sample factories distributed by Sematech, we compare the performance of these policies to other control strategies such as kanban and periodic lot release. We show that the policies designed for high variability environments have very attractive properties, including higher throughput than periodic release and much lower inventories than kanban.

I Background

Semiconductor manufacturing differs from many other manufacturing environments by the complexity and variability of its processes. A semiconductor process can be several hundred operations long, and the machines are relatively unreliable due to the leading edge technology involved. This makes semiconductor factories harder to control than other plants of comparable size.

We construct control policies by extending the structural properties found by optimal control formulations of smaller systems. This allows us to construct good, although not provably optimal, control schemes for large production systems.

In the following, we explain the structure of two such policies, and contrast them with control policies that are commonly found in industry. We then report simulation results from datasets distributed by Sematech, representing actual semiconductor factories.

A Unlimited inventory policies

An optimal control formulation consists of three parts: an objective function, the dynamics of the system, and constraints on the decision variables. Considering a single machine, the decision variables are the instantaneous production rates of each part type. The dynamics express how the production and demand rates interact to build or drain inventory, and how the machine failure and repair process behaves. The constraints state that the machine cannot be more than 100% utilized, that it cannot work when it is down, and that the buffer cannot exceed its limits. The objective function expresses the cost of holding inventory and of backlogging demand, and we assume that this cost is linear.

It can be shown [1] that the optimal control for this problem is a *hedging point control* where the machine works at maximal speed until it has exceeded demand by a certain amount, the hedging point. Thereafter, it will work at the demand rate. When the machine fails, it will again work at capacity after it is repaired, until it next reaches the hedging point.

Extensions of this control policy can be developed for other kinds of systems. These extensions work well, but have not been shown to be optimal. In a system with multiple part types, the machine will work on the part type that is most below its hedging point. Then, when the backlog of this part becomes equal to that of another part type, the machine will split its capacity between these two part types. This continues until all part types have reached their hedging points.

This policy can be applied as an input control to a large production system as well as a single machine. In this case, the hedging point for work release should take into account the expected lead time of the system, so that the output tracks demand correctly. If all hedging points are set to zero, this policy introduces new lots at the demand rate with constant intervals. The difference between the general hedging point control and periodic lot releases is illustrated in Figures 1 and 2 for a fab that starts empty. In both cases, the first lot is released to the fab at time zero, and finished product starts appearing one lead time later. The periodic release policy will thereafter have an average output rate about equal to the production target, while the

*This work was supported by the Advanced Research Projects Agency under contract N00174-93-C-0035.

hedging point policy will cause the system to work at its maximal rate until the hedging point is reached. Another difference is in the loading decisions that take place at individual machines. This is not specified in the periodic release policy, so it must be completed by some dispatch rule such as first come, first serve or shortest remaining processing time first. In contrast, a hedging point policy also makes loading decisions by comparing cumulative production and demand at each machine.

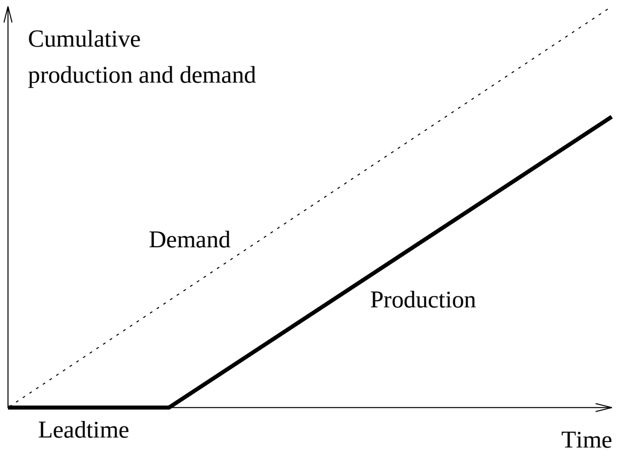

Fig. 1: Periodic lot release

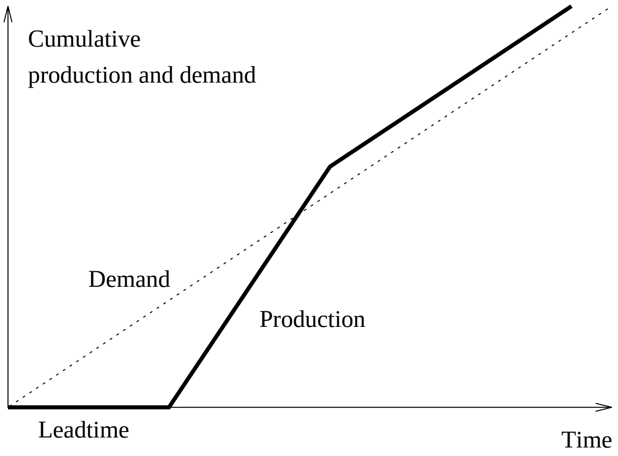

Fig. 2: Hedging point control

The value of the hedging point for the first machine in a system will control the amount of inventory in the entire system. The hedging points of the other machines control the distribution of that inventory. If all these hedging points are equal to the first one, the system will try to move material out of the line as quickly as possible, without building planned inventories anywhere in the system.

In a multiple part type system, there will be separate sets of hedging points for each process. Both lot release and dispatch decisions will therefore be based on the cumulative production and demand of each part type, and the system will be able to work around and recover from machine failures that affect only a subset of the products.

B Limited inventory policies

If some machine fails in a system controlled by hedging points, the other machines will keep working, and inventory will accumulate in front of the failed machine. This will help the production recover when the machine is repaired. However, if the machine is slower than the upstream stations, the inventory build-up may be larger than necessary for an efficient recovery. In these cases, it may be desirable to limit the inventory allowed in various locations.

This type of control policy, called a *two-boundary control*, arises from the optimal control formulation of a two-machine line where there is some cost associated with inventory in the buffer between the machines. This problem is analytically intractable, but numerical solutions [2] show that the optimal control can be closely approximated by a hedging point for each machine and a fixed limit to the inventory allowed in the buffer. The values of the hedging points and the buffer size depend on the reliability and production rates of the machines, and the costs of inventory and backlog.

Another control policy with fixed inventory limits is *kanban*. The main difference is that kanban only transmits demand information when pieces are loaded on downstream machines, while the hedging point and two-boundary control always distribute this information to all machines. This is an important distinction, since a failed machine will stop the information flow completely in kanban, and changes in demand rate will only be propagated as starvations and blockages. In the hedging point or two-boundary control, these changes take effect at all machines simultaneously, for a much faster system response.

II Methods

A The CAFE scheduling and simulation testbed

We have built a scheduling and simulation testbed integrated with the MIT CAFE computer aided fabrication environment [3]. This program can act as a factory floor dispatcher, where it recommends a daily schedule based on the factory state stored in the database. It can also run in a simulation mode, where it will project long-term consequences of following a certain scheduling policy under various assumptions on lot releases and machine reliabilities. The overall architecture of our testbed is shown in Figure 3. The main components are the data structures that represent the system state, the front ends that create the initial state, the system dynamics module that makes the state evolve in time, and the control modules that make release and dispatch decisions.

In its simulation mode, the testbed can read the Sematech datasets instead of the CAFE database. This allows us to test out our scheduling approaches in other manufactur-

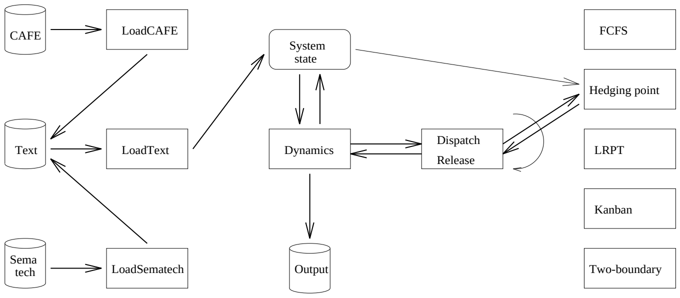

Fig. 3: CAFE testbed architecture

TABLE I: Summary of the Sematech sample fabs

Dataset no.	1	3
Number of products	2	11
Equipment groups	83	73
Number of machines	265	354
Raw process time	313 - 356 hrs	173 - 359 hrs
Process steps	210 - 245	270 - 473

ing systems than our own. The testbed does not currently support all features found in the Sematech datasets. For instance, we do not simulate operator availability, scrap, or rework. Setups are simulated, but the scheduling policies tested do not make any attempt to optimize the setup sequence.

This leads to different production rates from our testbed than from the Delphi simulator also distributed by Sematech. On the other hand, we allow the scheduling modules to control the lot release rates to the fab, not just manipulate the dispatch decisions as appears to be the case with Delphi. Previous work [4] has shown that the lot releases are a more important factor in fab performance than lot dispatch decisions on the shop floor.

B Experimental design

We have so far concentrated on Sematech datasets 1 and 3. These fabs are summarized in Table I. Since our testbed gives different results from the actual fab, we adjusted the lot release rates by initially introducing buffers of size 10 between all operations, and running the fabs by a kanban control, using the original lot release rates as finished goods demand rates. This gave maximal sustainable production rates for the model, and we reduced the release rates to 95% of these values for all products to create a heavily utilized, but not unstable environment.

We study four control policies for each fab. The base case is *periodic lot releases*, where lots are released individually at constant time intervals corresponding to the target production rate. This release policy is then combined with a dispatch policy that gives preference to lots with fewest remaining operations. Since we do not model scrap in these experiments, the average completion rate will be equal to the release rate in the long run, provided that this does not exceed the fab capacity.

Another candidate is *kanban control*, which means different things to different people. We introduce buffers of size 5 between all operations in each process. This is the same as circulating five cards beween each pair of successive operations. (The word "kanban" means "card" in Japanese.) We then create a deterministic demand process to pull lots out of finished goods inventory with constant time intervals corresponding to the target production rate. If no lots are present as finished goods inventory, we backlog the demand until it can be satisfied. Lots are released whenever the first machine used in a process is idle and its output buffer for that product is not full. Although other kanban models differ from ours, the essiential features are the fixed inventory limits between stations and the way demand information propagates from one production stage to the next upstream one.

The *hedging point control* tries to regulate each stage of the fab to track the demand rate as closely as possible. For this study, we compute a hedging point for the first machine as Z = (demand rate) $*$ (raw processing time), and set all other hedging points equal to this value. This choice of parameters does not allow for queueing time, so to the extent it can be realized, the lots will not spend any time in buffers, but go directly from one station to be worked on at the next one. It is possible to adjust the hedging points using the throughput times generated by the simulation instead of the raw processing time for even better backlog performance.

The *two-boundary control* combines the hedging point control with maximal inventory limits of 5 lots in all locations. This will cause an upstream station to avoid a particular product type if the downstream buffer contains five lots of that type, while applying hedging point control to all other product types.

For each sample fab and every control policy, we ran 30 replications, each for 52 weeks of 7-day, 24-hour operation, using an initial warm-up period of 8 weeks. The performance measures were averaged over all replications, and 95% confidence intervals estimated as $\overline{X} \pm 1.96S/\sqrt{N}$, where $\overline{X}$ is the sample mean, S the sample standard deviation, and N the sample size.

III Results

A Fab 1

In our simulations, fab 1 had a production target of 38.14 lots per week, consisting of 27.49 lots of product 1 and 10.65 lots of product 2. The raw processing times are 1.86 and 2.11 weeks, respectively. This gives a minimum WIP of 73.6 lots by Little's law WIP = (production rate)$*$ (delay). The variability in the fab will cause queuing delays, so the actual throughput times and WIP levels will be higher.

TABLE II: Overall performance measures for Fab 1

Control policy	Avg. WIP (lots)	Avg. lots/week	Std.dev. lots/week
Periodic rel.	101.6 ± 0.6	37.93 ± .05	2.16 ± .08
Kanban	337.5 ± 1.8	39.91 ± .04	1.75 ± .05
Hedging pt.	107.7 ± 0.4	38.92 ± .02	2.43 ± .07
Two-bound.	94.1 ± 0.9	39.28 ± .04	1.85 ± .05

TABLE III: Throughput times in weeks for Fab 1

Control policy	Product	Throughput time
Periodic rel.	1	2.529 ± .013
	2	3.016 ± .020
Kanban	1	6.956 ± .060
	2	10.582 ± .036
Hedging pt.	1	2.554 ± .008
	2	3.538 ± .010
Two-bound.	1	2.212 ± .010
	2	3.094 ± .067

Some overall performance measures of interest are the work in progress and lot completions per week, summed over all products. We are also interested in smoothness of output, which can be measured by the standard deviation of the number of lots completed per week. These performance measures are shown in Table II. We note that the kanban control has the highest WIP by far, while the two-boundary control has the least WIP.

We also note that the average output per week for the periodic release policy falls somewhat short of the target, while the other policies exceed the target over the measured period. This seems to indicate that the periodic release policy is still accumulating inventory, since the output will equal the input in the long run. The hedging point and two-boundary control work at capacity for some length of time (the steep part of the trajectory in Figure 2) before slowing down to the demand rate, so this drives its production rate up. Kanban will backlog demand during the initial phase, and will try to catch up when finished goods start appearing. This will lead to a similar behavior as the hedging point control, but it will also attempt to keep some finished goods in inventory, so its production rate will be even higher.

Another performance measure of great interest is the throughput time for each product type. We define this as the time from the lot is released to the fab until it leaves the last machine in its process. These results are shown in Table III. The results closely conform to the results for WIP, as one would expect. The two-boundary policy has the shortest throughput time for product 1 and the second shortest for product 2.

TABLE IV: Overall performance measures for Fab 3

Control policy	Avg. WIP (lots)	Avg. lots/week	Std.dev. lots/week
Periodic rel.	46.5 ± .01	27.01 ± .01	1.43 ± .04
Kanban	818.5 ± 1.2	28.67 ± .03	7.90 ± .32
Hedging pt.	73.6 ± .02	27.07 ± .01	1.50 ± .06
Two-bound.	73.6 ± .02	27.07 ± .01	1.50 ± .05

TABLE V: Fab 3, alternate buffer sizes

Control policy	Avg. WIP (lots)	Avg. lots/week	Std.dev. lots/week
Kanban 2	499.6 ± 1.2	27.63 ± .02	3.17 ± .08
Kanban 1	401.3 ± 1.1	27.25 ± .01	1.81 ± .05
Two-boundary 2	62.5 ± 0.6	27.07 ± .01	1.48 ± .05
Two-boundary 1	57.3 ± .01	27.08 ± .01	1.47 ± .07

B *Fab 3*

Fab 3 had a production target of 27.05 lots per week, divided among 11 product types. Table IV shows the overall performance measures for this fab.

As for fab 1, the periodic release policy falls slightly short of the production target, and the kanban policy builds excessive inventories. It is clear that the buffer sizes of 5 are too large for this system, since the two-boundary policy gives the same results as the hedging point policy. This indicates that the buffer sizes never become a constraint in the simulation. To test this hypothesis, we ran more simulations with the kanban and two-boundary policies, this time with buffer limits of 2 and 1 lots. We reduced the number of replications to 10 for these additional explorations. The results are shown in Table V. Even with buffer sizes of 1, kanban has much higher inventories than the other policies.

Kanban had the least standard deviation of lot completions per week in Fab 1, but has the highest output variability in this fab. Also, the two-boundary policy had a lower output variability than periodic release in Fab 1, but higher in this one. This makes it hard to draw conclusions based on this performance measure.

C *Workload sensitivity*

We tested the sensitivity to fab utilization rates by rerunning the experiments with production targets at 90% and 98% of the sustainable production rate instead of 95%, again using 10 replications per case. The results are in Tables VI and VII. The production targets are 36.14 and 39.35 lots per week for Fab 1 at 90% and 98% utilization, respectively. For Fab 3, the targets are 25.62 and 27.91 lots per week. We note a similar pattern as in the previous experiments, where the periodic release policy falls

TABLE VI: Fab 1, alternate demand rates

Control policy	Avg. WIP (lots)	Avg. lots/week	Std.dev. lots/week
Periodic rel. 90%	78 ± 4.5	36.10 ± .05	1.6 ± .5
Kanban 90%	383 ± 2.4	37.62 ± .05	2.2 ± .1
Hedging pt. 90%	90 ± 0.5	36.79 ± .03	2.6 ± .1
Two-bound. 90%	79 ± 0.3	36.91 ± .01	2.0 ± .1
Periodic rel. 98%	112 ± 0.7	39.09 ± .05	2.1 ± .2
Kanban 98%	335 ± 3.6	40.42 ± .04	1.6 ± .2
Hedging pt. 98%	126 ± 0.7	40.13 ± .04	2.4 ± .2
Two-bound. 98%	116 ± 2.2	40.51 ± .05	1.7 ± .2

TABLE VII: Fab 3, alternate demand rates

Control policy	Avg. WIP (lots)	Avg. lots/week	Std.dev. lots/week
Periodic rel. 90%	44 ± .01	25.58 ± .01	1.5 ± .1
Kanban 90%	860 ± 3.1	27.15 ± .06	8.5 ± .4
Hedging pt. 90%	71 ± .05	25.65 ± .01	1.4 ± .1
Two-bound. 90%	71 ± .04	25.64 ± .01	1.4 ± .1
Periodic rel. 98%	48 ± .01	27.92 ± .01	1.5 ± .1
Kanban 98%	794 ± 2.2	29.67 ± .06	7.6 ± .7
Hedging pt. 98%	75 ± .03	28.05 ± .02	1.9 ± .1
Two-bound. 98%	75 ± .02	28.06 ± .02	2.0 ± .1

slightly short of the production target (except in Fab 3 at 98%) and the kanban policy has very high inventories.

The WIP results from the periodic release policy in Fab 1 at 90% have a very high sample variance, reflected in the wide confidence intervals. This is because the individual simulation results were close to either 73 or 88 lots of WIP. No other case exhibited this bimodal behavior.

IV Discussion

Kanban is not a good choice for these production systems. Although kanban control imposes inventory limits at all stages in the production process, it also attempts to build inventory right up to these limits. In a production system characterized by many long processes, this will lead to a large number of small inventories, adding up to excessive work-in-progress. It is worth noting that the inventories decrease as the demand rate increases in a system controlled by kanban, while they increase in the other control policies we tested.

Another problem, not shown in these results, is that the production stages in a kanban line make loading decisions by tracking the loading process of the next downstream machine, while adding its own variability to the actual loading times. The next upstream machine will therefore see a noisier signal, and the variability will amplify upstream. In very long production processes, like semiconductor processes, production stages far upstream will alternate between periods of intense production pressure and periods of blockage. In contrast, the other policies regulate the lot release process, creating a more even workload at the upstream end of the process.

We have not tried to optimize the parameter choices of the different policies. It is probably the case that somewhat better results can be obtained by other distributions of buffer sizes in the kanban policy. Finding the optimal choice of all buffer sizes will be an enormous undertaking and will require very expensive computational resources, due to the size of the manufacturing systems involved. The hedging point and two-boundary controls are mainly controlled by the single hedging point of the first machine, with some minor adjustments possible by changing the other parameters of the system. This will be much easier to optimize and adjust in practice. Still, efficient tools for setting the parameters of these policies would be very useful.

The hedging point and two-boundary policies also allow for the just-in-time practice of forced improvements, where management reduces the inventory limits of the system gradually to uncover and solve production problems and to benefit from the learning that has already occurred in the organization. This can be done by adjusting the overall inventory limit, which is set by the hedging point of the first stage in each process, rather than adjusting every single buffer size as in a kanban system.

The hedging point and two-boundary policies are slightly more complex to implement than the periodic lot release policy. It is necessary to keep track of the production surplus for each product at each station. This quantity is the cumulative production plus the hedging point for the station, minus cumulative demand. In addition, the two-boundary control requires information about the queue length at the next downstream station for each process.

V Conclusion

The most important feature of the hedging point and two-boundary controls is that they distribute information about the production target directly to all machines. Local decisions are made on the basis of this information, the local production history, and the queue lengths of machines immediately downstream.

We have shown that the hedging point and two-boundary policies behave very well in the environments we studied. They attain the production rate target with low inventories. The two-boundary policy has inventory levels similar to the periodic lot release policy, which does not reach the production targets. Kanban behaves poorly in our experiments, building very large inventories in all cases tested.

References

[1] S. B. Gershwin, *Manufacturing Systems Engineering.*

Englewood Cliffs, New Jersey: PTR Prentice Hall, 1994.

[2] G. van Ryzin, S. X. C. Lou, and S. B. Gershwin, "Production control for a tandem two-machine system," *IIE Transactions*, vol. 25, pp. 5–20, Sep 1993.

[3] A. M. Bonvik, "Introduction to the CAFE scheduling and simulation testbed," CIDM Memo 94-11, Massachusetts Institute of Technology, Sep 1994.

[4] L. M. Wein, "Scheduling semiconductor wafer fabrication," *IEEE Transactions on Semiconductor Manufacturing*, vol. 1, pp. 115–130, Aug 1988.