Universität Würzburg
Institut für Informatik
Research Report Series

Performance Modelling of Pull Manufacturing Systems with Batch Servers and Assembly–like Structure

Alexander K. Schömig and **Matthias Kahnt**

Report No. 108 March 1995

Bayerische Julius–Maximilians Universität Würzburg,
Institut für Informatik, Lehrstuhl für Informatik III
Am Hubland, D-97074 Würzburg, Federal Republic of Germany
Tel.: +49-931-8885511, Fax: +49-931-8884601,
e-mail: schoemig@informatik.uni-wuerzburg.de

Abstract: *The manufacturing industries' need to focus on reducing work-in-process (WIP) as well as product cycle times has led to a widespread employment of the just–in–time (JIT) production philosophy and in particular kanban systems. Queueing models of several different kanban systems have been previously presented focusing on single job service machines. We extend the model and approximation method presented by Mitra and Mitrani, to investigate a singlecard kanban system with batch servers. Batch servers are machines that can process several jobs or lots at a time. We use the same technique to analyze assembly–like manufacturing systems and manufacturing systems with more than only one machine in each kanban cell.*

We compare numerical approximations with simulations of the configurations under consideration. We also discuss how given parameters influence performance measures such as production rate (throughput), cycle time, and WIP. This study was motivated by the need of semiconductor manufacturers for kanban–like pull production control systems.

Keywords: JIT, Kanban, Stochastic Manufacturing Models, Batch Server.

1 Introduction

During the past years much attention has been devoted to just–in–time (JIT) manufacturing systems as an alternative to push systems, traditionally used by European and U.S.–american manufacturers. In the philosophy of JIT, safety stocks are considered as a unnecessary waste of resources, since they require higher carrying costs. One way to implement a JIT–system, also called *pull system*, is the *kanban system*. It limits the WIP in a production line in an efficient way: Kanbans circulate within a production stage and their occurence at specific positions indicate the status of this stage. In other words, they signal to adjacent stages the need for raw material or the availability of finished parts and regulate the flow of material between stages. By adjusting the number of kanbans in each stage fab managers and stage supervisors can control the in–process inventory. However, increasing the buffer space means also increasing the cycle time and the work in process (WIP). The goals of maximizing the production rate and minimizing cycle time and WIP contradict each other. Hence, buffer space allocation is a crucial parameter that influences the performance of a production line as well as the service time distribution and lot sequencing rules.

Scheduling decisions have to be made when there are batch processing machines within a production line. *Batch servers* are machines that can process several jobs or lots at a time. Examples include plating baths, drying facilities, heat–treating ovens, e.g. in semiconductor wafer fabrication diffusion and oxidation ovens. The latter operations are very time consuming (cf. [15]). Each of those ovens can process a certain number of lots of wafers simultaneously. Since it is inefficient to run a long operation without utilizing the full capacity of the machine it may seem desirable always to load full batches, i.e. the maximum number of lots. Furthermore, additional lots cannot be added once the process has been started. But then a batch server might have to wait a long time for lots to form a full batch, leaving the machine idle. Even worse, this waiting time adds to the overall cycle time of a lot and downstream stages can be starved. To find a fair trade–off between lot cycle time and server utilization a nontrivial decision must be made whether to start the partial batch once the machine becomes available or to wait for additional lots.

Assembly lines are designated to build a finished product from component parts, e.g. mounting IC chips and other electronic devices onto circuit cards or attaching fenders and wheels to a car. In unpaced or asynchronous assembly lines appropriate buffers have to be provided in front of the assembly operation to avoid starving effects or blocking one of the feeder lines.

Our study is motivated by the ongoing discussion on whether and how to implement kanban–like pull production control systems in semiconductor fabrication facilities. In this paper we investigate the impact of buffer allocation and batching strategy on the performance of a singlecard kanban system. Chapter 2 gives an overview of the relevant literature on batch service systems, assembly–like queues, and pull production systems. In chapter 3, we first introduce the general model, and then we extend the methodology presented by Mitra and Mitrani [19, 20] for the cases of a) batch servers, b) assembly–like kanban systems, and c) manufacturing systems with more than only one machine in each kanban cell. Finally, we compare numerical approximations using our new algorithm with simulations of the configurations under consideration. We also report some general guidelines for the choice of system parameters.

2 Literature Review

Chaudhry and Templeton [4] discuss *bulk queueing models* in which jobs arrive in groups or are served as batches. They present a variety of single stage models and analytical techniques reviewing numerous papers. Gold [10, 11, 23] investigates sophisticated batch service systems in push and pull manufacturing environments as single stage systems by using embedded Markov chain techniques. Glassey and Weng [9], and Fowler et al. [8] approach the problem of controling batch processing systems in semiconductor wafer fabrication from a operation manager's point of view. These studies utilize online data of the specific facilities under consideration for decision making. The first study shows, using simulation, that forecasting information can significantly reduce the queueing time of lots in front of a batch server, the second extends this work to a multiproduct environment.

In an early theoretical work, Harrison [13] considered an assembly–like queue, which arrival streams of customers are independent renewal processes. Queue lengths are assumed to be unlimited. Lipper and Sengupta [16] study a queueing model of assembly–like manufacturing operations. They assume that customers arrive from n different classes according to independent Poisson processes with the same parameter into a single server station with exponentially distributed service times. Customer who find the queue for their class full are lost. They derive bounds for blocking probabilities, throughput, mean queue length, and mean sojourn times. Hopp and Simon [14] extend this work for pull production – like systems and again find upper and lower bounds for the throughput of the system under consideration. They focus on a single stage system with unlimited supply and demand.

The amount of literature on JIT and kanban systems has grown vast during the past decade. Golhar and Stamm [12] list in their overview more than 200 references to research papers and books on JIT. Berkley [3] reviews the the kanban production control research literature. He classfies the kanban models according to certain system features such as blocking mechanism and material handling.

The kanban system used in this study belongs to the category of pull production systems as previously presented by So and Pinault [22], Wang [24], DiMascolo et al. [6, 7], and in particular Mitra and Mitrani [19, 20]. The kanban blocking mechanism of these models is given by a single constraint which limits the input and output buffers of a production stage in contrast to two–card kanban systems. The authors of these studies try to overcome dimensionality (i.e. state space explosion) problems associated with Markov chain models by decomposing the whole line into smaller queueing networks, analyzing them seperately, and finding a solution for the entire line by applying an overall approximation algorithm. For example Mitra and Mitrani use a fixpoint equation approach that takes into account the interaction of adjacent stages.

3 Models and Analysis

3.1 General Model

The production line consists of N production stages, each stage has one machine. Processing times are assumed to be exponentially distributed with mean $1/\mu_k$, $k = 1, 2, \ldots, N$. There is storage space for a limited number of lots awaiting processing or authorization to move to the next stage. C_i kanbans are allocated to stage i. These kanbans are stored in the *bulletin board (BB)*. A lot may only enter a stage when there is a kanban available from the BB. In this case, a kanban is taken from the BB and attached to the entering lot; this kanban is not removed until the lot leaves the stage. Thus, the total number of lots in stage i is limited to C_i. For this reason, the total number of lots in the line may not exceed $\sum_{i=1}^{N} C_i$.

Upon completion of processing in stage i, a lot and its attached kanban proceed to the *output hopper (OH)* of stage i. Two possible actions can now take place: If the BB of stage $i+1$ is empty this lot has to wait in the OH until a free kanban of stage $i+1$ becomes available. This only can occur, when a lot leaves stage $i+1$. If there is a free kanban in the BB of stage $i+1$, the lot may leave stage i and enter stage $i+1$, after returning its kanban to the BB of stage i. This kanban is replaced by one of stage $i+1$. The lot goes to the input hopper of stage $i+1$ and awaits processing.

For our analysis we use the approximate solution method presented by Mitra and Mitrani: The line is decomposed into isolated components, representing a single stage (cf. [20], p. 1555, Fig. 4). Arriving lots and demands for finished goods are modeled by independent Poisson processes with rates ρ_k and σ_k, respectively. There are external buffers for incoming lots or demands. Arrivals that find the buffer full are lost. The buffer sizes are exactly the same as the capacity of the OH or BB of the adjacent stages. The state of the isolated stage k at time t is completely given by a tuple $(I_{k,t}, J_{k,t})$, where $-C_{k-1} \leq I_{k,t} \leq C_k$, $-C_{k+1} \leq J_{k,t} \leq C_k$ and $I_{k,t} + J_{k,t} \leq C_k$. When $I_{k,t} \leq 0$, $-I_{k,t}$ lots are waiting in the *external parts buffer*, else there are $I_{k,t}$ kanbans in the BB. Analogous, $J_{k,t}$ denotes the number of demands in the external demands buffer or finished lots in the OH of stage k. The number of lots at the service station is equal to $C_k - J_{k,t} - I_{k,t}$, when $J_{k,t} > 0$ and $I_{k,t} > 0$. Mitra and Mitrani point out, that $\mathcal{M} = \{(I_{k,t}, J_{k,t}), t \geq 0\}$ is a finite Markov process with the state space S_k and the state space distribution $\{p_{ij}\}$, where $p_{ij} = \lim_{t\to\infty} P(I_{k,t} = i,\ J_{k,t} = j)$, $(i, j) \in S_k$.

For the derivation of the state space probabilities we refer the reader to the orginal literature and only recall the basic equations of the steady state throughput of lots in stage k. The throughput T_k equals

a) $T_k = \rho_k \left[1 - P(I_k = -C_{k-1})\right]$, $\quad k = 2, 3, \ldots, N$,
(average number of lots entering the stage),

b) $T_k = \sigma_k \left[1 - P(J_k = -C_{k+1})\right]$, $\quad k = 1, 2, \ldots, N-1$,
(average number of demands accepted), and

c) $T_k = \mu_k \{1 - [P(I_k = C_k) + P(J_k = C_k) + P(I_k + J_k = C_k)]\}$, $\quad k = 1, 2, \ldots, N$,
(average number of lots being served).

Returning to the entire line consisting of N stages in tandem, we now approximate this system by calculating the parameters of the streams of lots and demands such that the original kanban system is modeled as close as possible. (Cf. Fig. 1.) Thus, we have to calculate the vectors $\vec{\rho} = (\rho_2, \rho_3, \ldots, \rho_N)$ and $\vec{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_{N-1})$. We assume $\rho_1 = \sigma_N = \infty$. By carrying out straightforward algebra Mitra and Mitrani derive equations of the form

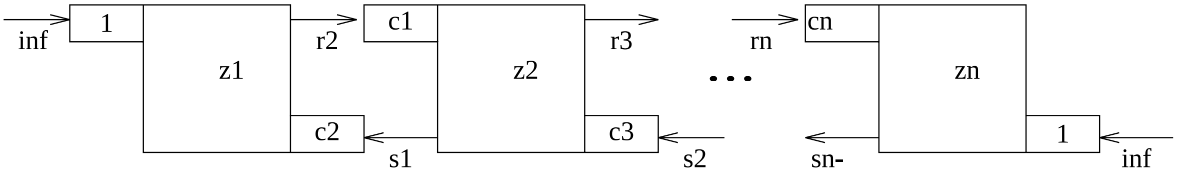

Figure 1: *Interaction of Stages*

$\vec{\rho} = f(\vec{\rho}, \vec{\sigma})$, $\vec{\sigma} = g(\vec{\rho}, \vec{\sigma})$ to calculate $\vec{\rho}$ and $\vec{\sigma}$, respectively. To solve this set of fix–point equations one has to ensure, that the throughput T_k through all stages is the same ($T = T_1 = T_2 = \ldots = T_N$) and the solution of the iterative procedure converges to a unique solution. Though Mitra and Mitrani deliver no formal proof they have strong arguments that this requirement is fulfilled for the given cases. Mitra and Mitrani produced best results using the following equations:

$$\rho_k = \frac{T_{k-1}}{1 - P(J_{k-1} = C_{k-1})}, \qquad k = 2, 3, \ldots, N\,, \tag{1}$$

$$\sigma_k = \frac{T_{k+1}}{1 - P(J_k = -C_{k+1})}, \qquad k = 1, 2, \ldots, N-1\,. \tag{2}$$

After one has calculated $\vec{\rho}$ and $\vec{\sigma}$ all other performance measures can be derived easily.

3.2 Systems with Batch Servers

Model and Analysis

We now extend the method described above to kanban systems with batch servers. We model a batch server with exponentially distributed service time (mean $1/\mu$), *threshold* A, and *batchsize* B. The queueing discipline is FCFS. An idle server waits until at least A lots have been accumulated in the queue. After serving one batch, the server proceeds according to the number of waiting lots: If there are more than A lots waiting, a maximum of B lots are loaded into the server. Remaining lots stay in queue. The state space of an isolated stage with a batch server is given by the triple $(I_{k,t}, R_{k,t}, J_{k,t})$. $R_{k,t}$ denotes the number of lots being served at time t. Thus, $R_{k,t} \in \{0, A_k, A_k + 1, \ldots, B_k\}$ and $I_{k,t} + R_{k,t} + J_{k,t} \le C_k$. Again, we have to deal with a finite, irreducible Markov process $\mathcal{M} = \{(I_{k,t}, R_{k,t}, J_{k,t}),\, t \ge 0\}$ with the steady state probability distribution $\{p_{irj}\}$, where $p_{irj} = \lim_{t\to\infty} P(I_{k,t} = i, R_{k,t} = r, J_{k,t} = j)$, $(i, r, j) \in S_k$.

Now we have to consider a three dimensional state space as depicted in Fig. 2. This state space consists of several planes, one of them for a certain number r of lots being served. The outline of these planes for $A_k \le r \le B_k$ is exactly the same as the state space of an usual stage. The plane for $r = 0$ can be considered as a special case, since within

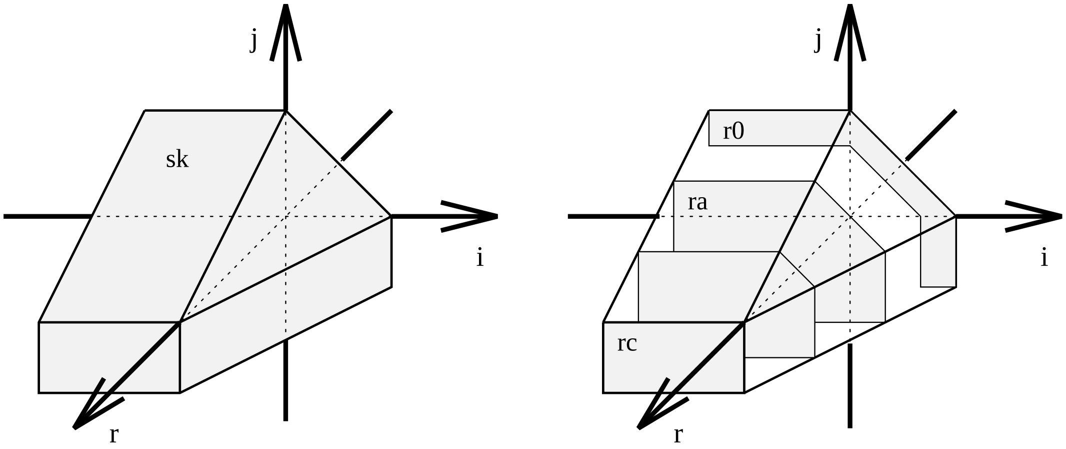

Figure 2: *Sketch of State Space*

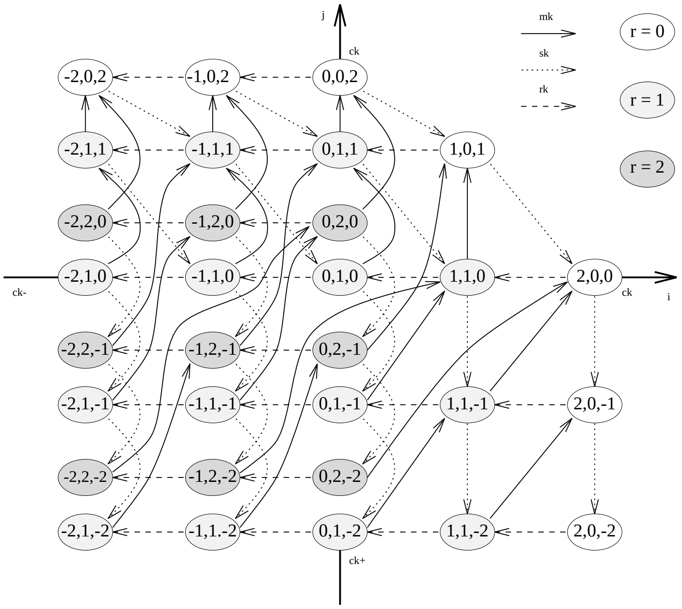

Figure 3: *Sample State Space*

this plane there are only states with a number of waiting lots less than A. State transitions with rates ρ_k take place within a plane $r = A_k$ when the number of lots being served does not change. A state transition between a state of plane $r = 0$ and of plane $r = A_k$ occurs every time a lot arrives at the server to complete a minimal batch of size A_k (cf. Fig. 3). Other state transitions can happen with rate μ_k depending on j, A_k, B_k and C_k. In other words, after completion of a service it is decisive how many new lots can enter the stage because of free kanbans and proceed to the server, where the number of waiting lots can exceed A_k. Given that stage k is in state $z = (i, r, j)$, then we define

$$Z^k_{i,r,j} = \begin{cases} \min(-i, \min(r, -j)) + C_k - r \ , & i, j < 0 \\ C_k - i\theta(i > 0) - j\theta(j > 0) - r, & \text{else} \end{cases}$$

as the number of queued lots at the server in state $\tilde{z}$. $\theta(a) = 1$, if a is true, $\theta(a) = 0$ else. State $\tilde{z}$ is reached from state z upon service completion. State probabilities are calculated using the following balance equations:

$\underline{j \geq 0}$:

$$\begin{aligned}
&\Big[\mu_k\,\theta\Big(r>0\Big)+\rho_k\,\theta\Big(i>-C_{k-1}\Big)+\sigma_k\,\theta\Big(j>-C_{k+1}\Big)\Big]\cdot p_{irj}\\
= \quad & \mu_k \sum_{b=A_k}^{B_k}\Big[p_{\ddot{u},b,j-b}\cdot\theta\Big(\ddot{u}\geq -C_{k-1}\wedge j-b\geq -C_{k+1}\Big)\Big]\cdot\\
&\quad\Big[\theta\left(r=0\wedge Z^k_{\ddot{u},b,j-b}<A_k\right)+\theta\left(B_k>r\geq A_k\wedge Z^k_{\ddot{u},b,j-b}=r\right)+\\
&\quad\ \theta\left(r=B_k\wedge Z^k_{\ddot{u},b,j-b}\geq B_k\right)\Big]\\
+ \quad & \rho_k\Big[p_{i+1,r,j}\cdot\theta\Big(i+r+j<C_k\Big)+p_{i+1,0,j}\,\theta\Big(i+r+j=C_k\wedge r=A_k\Big)\Big]\\
+ \quad & \sigma_k p_{i-1,r,j+1}\cdot\theta\Big(j+r<C_k\wedge i>-C_{k-1}\Big)\\
+ \quad & \sigma_k p_{i-1,0,j+1}\cdot\theta\Big(j+r=C_k\wedge i>-C_{k-1}\wedge r=A_k\Big) \qquad (3)
\end{aligned}$$

$\underline{j < 0}$:

$$\begin{aligned}
&\Big[\mu_k\,\theta\Big(r>0\Big)+\rho_k\,\theta\Big(i>-C_{k-1}\Big)+\sigma_k\,\theta\Big(j>-C_{k+1}\Big)\Big]\cdot p_{irj}\\
= \quad & \mu_k \sum_{b=A_k}^{B_k}\Big[p_{\ddot{u},b,j-b}\cdot\theta\Big(\ddot{u}\geq -C_{k-1}\wedge j-b\geq C_{k+1}\Big)\Big]\cdot\\
&\quad\Big[\theta\left(r=0\wedge Z^k_{\ddot{u},b,j-b}<A_k\right)+\theta\left(B_k>r\geq A_k\wedge Z^k_{\ddot{u},b,j-b}=r\right)+\\
&\quad\ \theta\left(r=B_k\wedge Z^k_{\ddot{u},b,j-b}\geq B_k\right)\Big]\\
+ \quad & \rho_k\Big[p_{i+1,r,j}\,\theta\Big(i+r<C_k\Big)+p_{i+1,0,j}\,\theta\Big(i+r=C_k\wedge r=A_k\Big)\Big)\\
+ \quad & \sigma_k p_{i,r,j+1} \qquad (4)
\end{aligned}$$

where $\ddot{u}=i+j-b$, if $j>0$, else $\ddot{u}=i-b$.

The equation for calculating the throughput has to be altered for a batch server stage: Since several lots are served at a time, and μ_k refers to an entire batch we introduce the mean lot service rate $\hat{\mu}_k$. It is given by

$$\hat{\mu}_k=\mu_k\sum_{r=A_k}^{B_k} rP(R_k=r)/P(A_k\leq R_k\leq B_k)\,.$$

Hence we obtain the throughput using

$$T_k=\hat{\mu}_k P(A_k\leq R_k\leq B_k)\,.$$

Cycle times are calculated using Little's Law.

Numerical Results

For all calculations presented here we used exponential service times with mean $\mu^{-1}=1.0$ time units and a model of a line that consists of five stages and may have up to two batch server stages. In the following tables 'A' refers to the analytical results, 'S' to simulation results, and 'BS' denotes the stage(s) where we find batch servers. To validate the analytical results 10 independent replications of a simulation run of a system were done. Each run had a total length of 10,000 time units and statistical data from the initial transient period were discarded. The columns containing the results from these simulation runs also show the lower and upper bound of the 95% confidence interval of the mean value. Tables 1, 3, and 4 list throughput, WIP, and cycle time.

	Throughput				WIP				Cycle Time			
BS	A	S			A	S			A	S		
–	0.857	0.858	0.860	0.862	23.77	23.91	24.06	24.21	27.72	27.79	27.98	28.18
1	0.866	0.869	0.872	0.875	24.81	25.21	25.32	25.43	28.65	28.91	29.03	29.16
2	0.874	0.878	0.881	0.884	24.49	24.98	25.10	25.23	28.02	28.33	28.48	28.63
3	0.882	0.881	0.883	0.886	23.56	23.65	23.84	24.04	26.73	26.73	26.99	27.25
4	0.880	0.876	0.879	0.883	22.05	22.24	22.39	22.54	25.07	25.30	25.47	25.64
5	0.870	0.870	0.873	0.876	20.96	20.91	20.99	21.08	24.10	23.90	24.05	24.19
1, 2	0.881	0.887	0.890	0.893	25.77	26.68	26.76	26.84	29.24	29.95	30.08	30.20
1, 3	0.894	0.894	0.898	0.901	25.19	26.19	26.30	26.40	28.19	29.17	29.30	29.42
1, 4	0.897	0.893	0.896	0.899	23.59	24.89	25.01	25.13	26.29	27.75	27.92	28.09
1, 5	0.884	0.888	0.891	0.893	22.06	22.31	22.41	22.50	24.95	25.05	25.16	25.27
2, 3	0.902	0.906	0.908	0.911	24.79	25.27	25.47	25.67	27.48	27.78	28.04	28.31
2, 4	0.911	0.913	0.915	0.917	23.09	23.46	23.69	23.93	25.34	25.63	25.89	26.16
2, 5	0.897	0.900	0.902	0.905	21.45	21.81	21.95	22.08	23.92	24.17	24.32	24.47
3, 4	0.912	0.907	0.910	0.913	21.16	20.89	21.08	21.26	23.19	22.98	23.16	23.33
3, 5	0.902	0.900	0.902	0.905	19.86	21.20	21.33	21.47	22.01	23.49	23.64	23.79
4, 5	0.889	0.885	0.887	0.890	18.22	20.89	21.02	21.14	20.49	23.55	23.69	23.83

Table 1: Comparison between analytical and simulation results

Table 1 summarizes the results for different locations of the batch service stage within the line. Each stage has six kanbans, and we choose $A = 2$ and $B = 3$. The relative error of the approximation varies in the range of $0\% - 13\%$. However, we conclude that the system behaviour is fairly modeled. Throughput has maxima for the cases when the batch server(s) is (are) located in the center of the line. Moving the batch server towards the end of the line decreases WIP and cycle time.

To demonstrate how the removal of kanbans, adding kanbans or an completely unbalanced kanban distribution may change the performance measures of the system, we assigned kanbans to the stages according to Tab. 2. All other parameters are the same as before. Unbalancing the line leads only to a slight decrease of the throughput of about 1%, but WIP and cycle time drop approximately 10% as shown in Tab. 3. If few kanbans from the first two stages are removed, we find that the throughput decreases 2%. In this case WIP and cycle time are much more reduced than before: 15–18%.

	Kanbans				
	C_1	C_2	C_3	C_4	C_5
Batch server at stage 3					
G 1	6	6	6	6	6
A 1	4	5	6	6	6
S 1	4	5	6	7	8
E 1	6	6	7	7	8
Batch server at stage 3 and 4					
G 2	6	6	6	6	6
A 2	4	5	6	6	6
S 2	4	5	6	7	8
E 2	6	6	7	7	8

Table 2: *Assignment of Kanbans*

We conclude that for the sake of a little loss in throughput one can gain benefits in terms

	Throughput				WIP				Cycle Time			
	A	S			A	S			A	S		
G 1	0.882	0.881	0.883	0.886	23.56	23.65	23.84	24.04	26.73	26.73	26.99	27.25
A 1	0.866	0.861	0.866	0.871	19.44	19.43	19.75	20.08	22.46	22.47	22.81	23.15
S 1	0.874	0.871	0.875	0.879	20.23	20.18	20.52	20.85	23.14	23.11	23.44	23.76
E 1	0.895	0.895	0.897	0.899	25.06	25.03	25.45	25.87	28.02	27.93	28.38	28.83
G 2	0.912	0.907	0.910	0.913	21.16	20.89	21.08	21.26	23.19	22.98	23.16	23.33
A 2	0.891	0.889	0.892	0.895	17.17	16.77	17.01	17.26	19.28	18.80	19.07	19.35
S 2	0.895	0.891	0.895	0.898	17.54	17.27	17.66	18.05	19.61	19.34	19.74	20.15
E 2	0.918	0.913	0.918	0.922	21.78	21.66	22.07	22.49	23.73	23.60	24.05	24.50

Table 3: Varying Assignment of Kanbans — Results

of cutting down WIP and cycle time. Adding kanbans increases throughput, WIP, and cycle time.

To study the impact of threshold A_2 we observed the behaviour of a three stage line with a batch server in the center stage. Batchsize is choosen as $B = 4$, the number of kanbans per stage is as denoted in Tab. 4. Though performance measures do not change more than 5%, one must recognize an overall trend: Increasing A_2 decreases the throughput and increases both WIP and cycle time. We explain this effect by the fact that the batch server has to wait longer before service starts, when a full batch rule is enforced. Consequently, lots spend more time queued, contributing significantly to the cycle time.

	Throughput				WIP				Cycle Time			
A_2	A	S			A	S			A	S		
4 Kanbans per stage												
1	0.915	0.892	0.898	0.904	9.89	9.81	9.90	10.00	10.81	10.93	11.03	11.12
2	0.891	0.897	0.903	0.909	9.78	9.94	10.00	10.07	10.97	10.97	11.08	11.18
3	0.874	0.890	0.895	0.901	9.84	10.14	10.18	10.23	11.26	11.31	11.37	11.43
4	0.863	0.881	0.884	0.887	9.96	10.35	10.40	10.45	11.55	11.70	11.76	11.82
8 Kanbans per stage												
1	0.968	0.947	0.954	0.960	19.51	19.25	19.56	19.87	20.15	20.14	20.51	20.89
2	0.960	0.950	0.955	0.960	19.23	19.29	19.61	19.93	20.03	20.20	20.54	20.8
3	0.950	0.951	0.956	0.961	19.45	19.73	20.04	20.35	20.47	20.70	20.96	21.21
4	0.941	0.948	0.954	0.961	19.65	19.92	20.19	20.46	20.87	20.82	21.16	21.50
Range of relative Error	1-2%				0-4%				1-2%			

Table 4: Variation of the threshold A_2

3.3 Assembly like Systems

Model and Analysis

We introduce the most simple assembly like system consisting of three kanban controlled production lines. There are two *feeder lines* E_1 and E_2 with N respectively M stages and one *assembly line* V with Z stages (cf. Fig. 4). Each of these stages fits the description of the general model, except for the *assembly stage* a, where the parts produced in the feeder lines are assembled. It is placed as the first stage of line V. A lot waiting in the

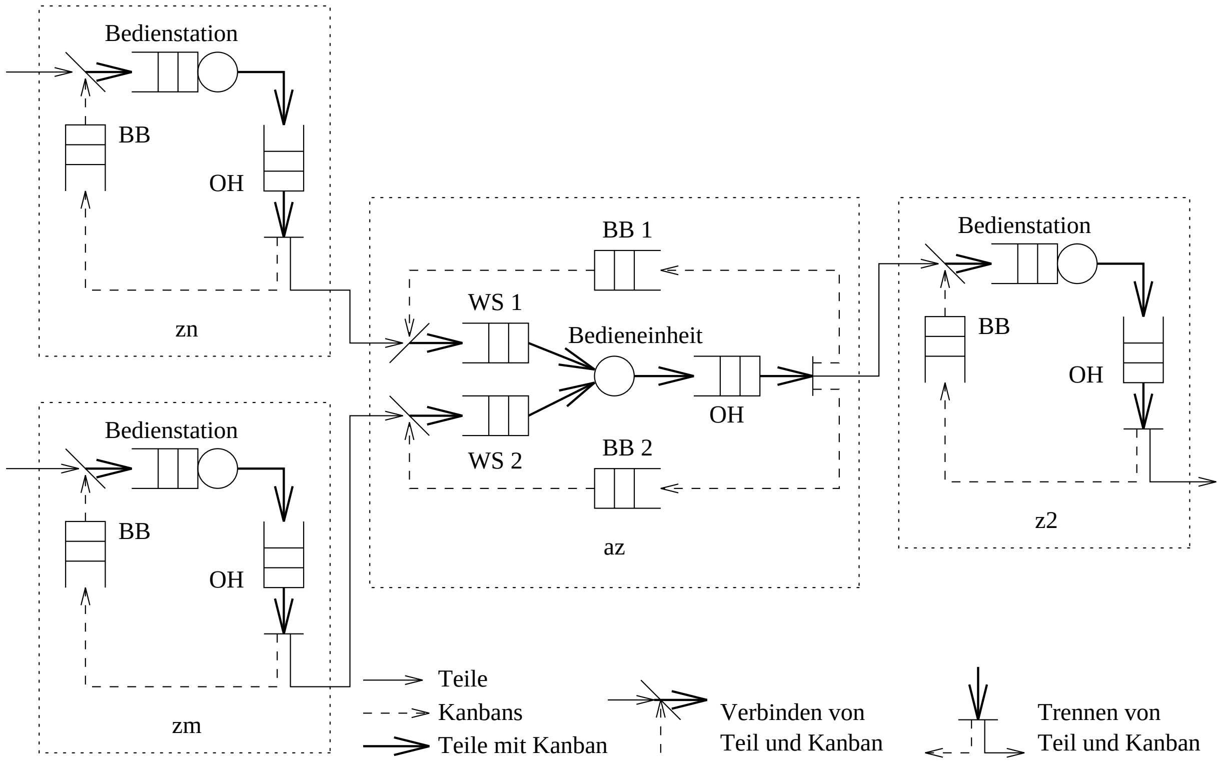

Figure 4: *Flow of lots and kanbans through assembly stage* a

output hopper of the last stage of E_k is only allowed to enter the assembly stage a if there is a free kanban available in the corresponding bulletin board BB k ($k = 1, 2$). If the lot can enter stage a, a stage a kanban is attached and the lot is stored in the input hopper WS k. The server starts processing if there is at least one part in WS 1 and WS 2. Upon completion of the assembly operation the lot with its two attached kanbans is moved to the output hopper where it waits for a kanban becoming gettable in stage 2 of line V. After the lot has received authorization to move, the kanbans are detached and returned to their bulletin boards. There are C_a kanbans allocated to the assembly stage for lots produced in each feeder line.

The formal state description of the isolated assembly stage is done by triples $(I_{a,1,t}, I_{a,2,t}, J_{a,t})$, where $I_{a,1,t}$ and $I_{a,2,t}$ denote the number of waiting lots in the external parts buffers respectively of free kanbans in the bulletin boards BB 1 and BB 2. Again, $J_{a,t}$ expresses how many lots/kanbans are waiting in OH/external demands buffer. We get a finite, irreducible Markov process $\mathcal{M} = \{(I_{a,1,t}, I_{a,2,t}, J_{a,t}); t \geq 0\}$ with steady state probability distribution $\{p_{i_1,i_2,j}\}$, where

$$p_{i_1,i_2,j} = \lim_{t\to\infty} P(I_{a,1,t} = i_1, I_{a,2,t} = i_2, J_{a,t} = j), \qquad (i_1, i_2, j) \in S_a. \tag{5}$$

Finally, the following balance equations can be derived:

$$
\begin{aligned}
& \Big[\mu_a^V\, \theta\left(i_1 < C_a^V \wedge i_2 < C_a^V \wedge j < C_a^V \wedge \max\left(i_1, i_2\right) + j < C_a^V\right) \\
& \quad + \;\; \rho_{a,1}^V\, \theta\left(i_1 > -C_N^1\right) + \rho_{a,2}^V\, \theta\left(i_2 > -C_M^2\right) + \sigma_a^V\, \theta\left(j > -C_2^V\right)\Big]\, p_{i_1,i_2,j} \\
= & \qquad \mu_a^V \left[p_{i_1,i_2,j-1}\, \theta\Big(j > 0\Big) + p_{i_1-1,i_2-1,j-1}\, \theta\left(j \leq 0 \wedge i_1 > -C_N^1 \wedge i_2 > -C_M^2\right)\right] \\
& \quad + \;\; \sigma_a^V\, p_{i_1-1,i_2-1,j+1}\, \theta\left(j \geq 0 \wedge j < C_a^V \wedge i_1 > -C_N^1 \wedge i_2 > -C_M^2\right) \\
& \quad + \;\; \sigma_a^V\, p_{i_1,i_2,j+1}\, \theta\Big(j < 0\Big) \\
& \quad + \;\; \rho_{a,1}^V p_{i_1+1,i_2,j}\, \theta\left(i_1 + j < C_a^V \wedge i_1 < C_a^V\right) \\
& \quad + \;\; \rho_{a,2}^V p_{i_1,i_2+1,j}\, \theta\left(i_2 + j < C_a^V \wedge i_2 < C_a^V\right), \qquad (i_1, i_2, j) \in S_a
\end{aligned}
\tag{6}
$$

To calculate the throughput of stage a as the average number of lots being served

$$
\begin{aligned}
T_a^V \;=\; & \mu_a^V \Big\{1 - \Big[P\left(\max\left(I_{a,1}, I_{a,2}\right) = C_a^V\right) + P\left(J_a = C_a^V\right) \\
& + \; P\left(\max\left(I_{a,1}, I_{a,2}\right) + J_a = C_a^V\right)\Big]\Big\}
\end{aligned}
\tag{7}
$$

has to be used.

To solve the fixpoint equations of the entire system the external arrival rates of assembly stage a and the last stages of E_1 and E_2 can be calculated using:

$$
\rho_{a,1}^V = \frac{T_N^1}{1 - P\left(J_N^1 = C_N^1\right)}, \qquad \rho_{a,2}^V = \frac{T_M^2}{1 - P\left(J_M^2 = C_M^2\right)}, \tag{8}
$$

$$
\sigma_N^1 = \frac{T_a^V}{1 - P\left(J_N^1 = -C_a^V\right)}, \qquad \sigma_M^2 = \frac{T_a^V}{1 - P\left(J_M^2 = -C_a^V\right)}. \tag{9}
$$

The other arrival rates are calculated according to equations (1) and (2).

Numerical Results

According to the studies of Baker, Powell and Pike [1, 2] conducted for non–kanban controlled lines, we did the same approach and analysis for a kanban controlled line.

Tables 5 and 6 show the results of unbalancing the mean processing times 'MPT' in an assembly like system consisting of four stages: Two feeder stages 'Feed.', one assembly stage a and one further stage. There are two kanbans allocated in each stage. The performance measures signed with number k refer to lots produced in feeder line k. The unbalancing is done keeping the sum of the mean processing times (i.e. the average amount of work) constant. The throughput has maxima and WIP and cycle time are decreased in case of the mean processing times 1.10 and 0.80. The so–called 1% principle is confirmed for kanban controlled lines. It says that throughput can be inreased by unbalancing the system, but only on the order of 1%.

MPT		Throughput				WIP 1				WIP 2			
Feed.	a	A	S			A	S			A	S		
0.90	1.2	0.658	0.661	0.664	0.666	4.80	4.87	4.89	4.90	4.80	4.87	4.89	4.90
0.95	1.1	0.673	0.676	0.679	0.682	4.75	4.86	4.87	4.88	4.75	4.85	4.86	4.88
1.00	1.0	0.683	0.686	0.689	0.692	4.70	4.80	4.81	4.82	4.70	4.80	4.81	4.82
1.05	0.9	0.689	0.691	0.694	0.697	4.61	4.74	4.77	4.79	4.61	4.74	4.76	4.77
1.10	0.8	0.690	0.690	0.694	0.697	4.51	4.65	4.67	4.69	4.51	4.65	4.68	4.70
1.15	0.7	0.686	0.684	0.687	0.690	4.39	4.57	4.58	4.60	4.39	4.54	4.56	4.58
1.20	0.6	0.677	0.674	0.677	0.679	4.28	4.45	4.47	4.49	4.28	4.42	4.44	4.46
1.25	0.5	0.665	0.661	0.665	0.669	4.16	4.35	4.38	4.40	4.16	4.32	4.35	4.38
1.30	0.4	0.651	0.646	0.649	0.653	4.04	4.22	4.25	4.27	4.04	4.21	4.25	4.28
1.35	0.3	0.635	0.628	0.633	0.638	3.94	4.11	4.14	4.17	3.94	4.11	4.16	4.20
		Range of relative Error											
		0–1%				2–5%				2–5%			

Table 5: *Throughput and WIP of the unbalanced assembly–like system*

MPT		Cycle Time 1				Cycle Time 2			
Feed.	a	A	S			A	S		
0.90	1.2	7.30	7.33	7.37	7.40	7.30	7.33	7.37	7.40
0.95	1.1	7.06	7.13	7.17	7.21	7.06	7.12	7.16	7.20
1.00	1.0	6.86	6.95	6.98	7.01	6.86	6.95	6.98	7.01
1.05	0.9	6.69	6.83	6.87	6.91	6.69	6.83	6.86	6.89
1.10	0.8	6.53	6.69	6.73	6.77	6.53	6.71	6.74	6.78
1.15	0.7	6.41	6.63	6.67	6.71	6.41	6.60	6.63	6.66
1.20	0.6	6.32	6.57	6.60	6.64	6.32	6.54	6.57	6.59
1.25	0.5	6.25	6.54	6.58	6.62	6.25	6.51	6.54	6.57
1.30	0.4	6.21	6.48	6.54	6.60	6.21	6.49	6.54	6.58
1.35	0.3	6.20	6.47	6.54	6.60	6.20	6.53	6.57	6.61
		Range of relative Error							
		1–5%				1–6%			

Table 6: *Cycle Times of the unbalanced assembly–like system*

Another possibility of increasing throughput is to unbalance the distribution of kanbans within the system. We studied the assembly–like system shown in Fig. 5. It consists of three feeder and two assembly lines. There are two kanbans in each stage and the processing times are exponentially distributed with mean one. The additional kanbans are distributed according to Table 7. The throughput is maximal for two different distributions of kanbans (V 3 and O 6) as shown in Tables 8 and 10. It is remarkable that O 6 gains high throughput and low values of WIP and cycle time (cf. Table 10). So it should be preferred to V 3. This result can be explained by the effects of blocking and starvation within the system.

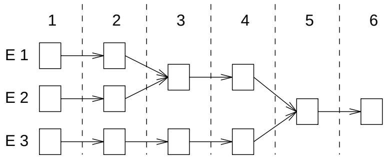

Figure 5: *Positions*

	Aditional Kanbans at					
	1	2	3	4	5	6
G	0	0	0	0	0	0
V 1	2	0	0	0	0	0
V 2	0	2	0	0	0	0
V 3	0	0	2	0	0	0
V 4	0	0	0	2	0	0
V 5	0	0	0	0	2	0
V 6	0	0	0	0	0	2
O 0	0	1*	0	0	0	0
O 1	1	1*	0	0	0	0
O 2	0	1	0	0	0	0
O 3	0	1*	1	0	0	0
O 4	0	1*	0	1	0	0
O 5	0	1*	0	0	1	0
O 6	0	1*	0	0	0	1

* refers only to E_1 and E_2

Table 7: *Assignment of kanbans*

	Throughput			
	A	S		
G	0.584	0.600	0.603	0.606
V 1	0.598	0.611	0.615	0.619
V 2	0.618	0.630	0.634	0.637
V 3	0.628	0.641	0.644	0.647
V 4	0.623	0.638	0.640	0.641
V 5	0.612	0.625	0.628	0.630
V 6	0.592	0.608	0.610	0.611
O 0	0.599	0.615	0.617	0.618
O 1	0.606	0.618	0.621	0.624
O 2	0.614	0.627	0.630	0.633
O 3	0.623	0.634	0.636	0.639
O 4	0.625	0.638	0.641	0.644
O 5	0.619	0.631	0.634	0.637
O 6	0.605	0.617	0.620	0.622
	relative Error 2–3%			

Table 8: *Throughput with three feeder lines*

	WIP 1				WIP 2				WIP 3			
	A	S			A	S			A	S		
G	8.98	9.41	9.44	9.47	8.98	9.42	9.45	9.47	9.47	9.72	9.76	9.81
V 1	11.36	11.74	11.78	11.81	11.36	11.76	11.79	11.83	11.73	11.91	11.95	11.99
V 2	11.42	11.80	11.85	11.89	11.42	11.81	11.86	11.91	11.60	11.88	11.92	11.97
V 3	11.07	11.46	11.51	11.57	11.07	11.43	11.46	11.49	11.30	11.58	11.64	11.69
V 4	8.68	10.26	10.36	10.45	8.68	10.27	10.37	10.46	10.64	11.13	11.22	11.32
V 5	10.21	9.85	9.95	10.04	10.21	9.84	9.93	10.03	11.00	10.69	10.78	10.87
V 6	9.13	9.44	9.50	9.56	9.13	9.43	9.49	9.55	9.68	9.79	9.85	9.90
O 0	10.35	10.73	10.78	10.84	10.35	10.74	10.79	10.84	9.34	9.57	9.62	9.67
O 1	11.58	11.90	11.95	12.01	11.58	11.92	11.97	12.02	10.55	10.83	10.87	10.90
O 2	11.45	11.84	11.89	11.94	11.45	11.85	11.90	11.94	10.50	10.79	10.83	10.86
O 3	11.40	11.85	11.88	11.91	11.40	11.82	11.85	11.88	10.30	10.57	10.60	10.63
O 4	10.60	11.40	11.47	11.54	10.60	11.40	11.47	11.53	9.78	10.23	10.29	10.36
O 5	11.00	11.18	11.25	11.31	11.00	11.16	11.24	11.32	10.00	9.96	10.03	10.10
O 6	10.47	10.79	10.86	10.94	10.47	10.81	10.88	10.95	9.46	9.64	9.69	9.74
	Range of relative Error											
	2–16%				2–16%				2–5%			

Table 9: *WIP with three feeder lines*

	Cycle Time 1				Cycle Time 2				Cycle Time 3			
	A	S			A	S			S	S		
G	15.37	15.57	15.67	15.76	15.37	15.58	15.67	15.76	16.21	16.09	16.20	16.30
V 1	19.00	19.03	19.16	19.28	18.60	19.06	19.18	19.31	19.61	19.29	19.44	19.59
V 2	18.48	18.56	18.69	18.82	18.48	18.57	18.71	18.85	18.77	18.65	18.81	18.98
V 3	17.63	17.77	17.88	17.99	17.63	17.71	17.80	17.89	17.99	17.97	18.08	18.19
V 4	13.94	16.04	16.19	16.35	13.94	16.06	16.21	16.37	17.07	17.38	17.55	17.71
V 5	16.68	15.69	15.85	16.00	16.68	15.68	15.83	15.98	17.97	16.99	17.18	17.37
V 6	15.42	15.46	15.59	15.71	15.42	15.45	15.57	15.69	16.34	16.07	16.16	16.24
O 0	17.29	17.42	17.49	17.56	17.29	17.44	17.50	17.57	15.60	15.54	15.61	15.67
O 1	19.11	19.11	19.24	19.36	19.11	19.14	19.27	19.39	17.41	17.41	17.49	17.57
O 2	18.66	18.71	18.87	19.02	18.66	18.73	18.88	19.03	17.11	17.09	17.18	17.26
O 3	18.30	18.59	18.66	18.74	18.30	18.52	18.62	18.72	16.53	16.58	16.66	16.73
O 4	16.95	17.85	17.90	17.95	16.95	17.85	17.89	17.94	15.65	15.90	16.07	16.23
O 5	17.76	17.61	17.73	17.84	17.76	17.58	17.72	17.86	16.16	15.65	15.81	15.97
O 6	17.30	17.41	17.53	17.66	17.30	17.42	17.56	17.70	15.64	15.53	15.64	15.75
	Range of relative Error											
	1–14%				1–14%				0–5%			

Table 10: *Cycle Time with three feeder lines*

3.4 Multiple Machine Stages

Model and Analysis

Mitra and Mitrani's model only copes kanban controlled systems having a single machine in each stage. This is not a very realistic assumption as most existing systems have several machines in each stage. One approach to extend the analysis of those systems is to use the *flow equivalent server* approximation technique.

Queueing networks that do not have a product form solution or suffer from state space explosion for a feasible numerical solution can be analyzed using *aggregation techniques*. The strategy is based on the reduction of the state space of the network replacing a subnetwork by a single queue with queue length dependent service rates. This composite queue is intended to behave as the entire subnetwork in its interaction with the remaining networks. In other words, this queue should be flow equivalent to the replaced network assuming equal load. This is done by setting the conditional throughputs of the isolated subnetwork equal to the load dependent service rates of the flow equivalent server. For more details see [21], pp. 165.

In the case of a kanban controlled stage, the machines/servers m_k^i (i.e. machine i of stage k) are considered as the subnetwork. To obtain the conditional throughputs of the subnetwork any appropriate technique can be used. We used Marie's Method (cf. [17, 18]) as suggested by Dallery [5].

As done before, we use the Markov process $\mathcal{M} = \{(I_{k,t}, J_{k,t}); t \geq 0\}$ to analyze the isolated stage and consequently derive the following balance equations:

$$\begin{aligned}
& p_{ij} \left[\mu_k\Big(G^k_{i,j}\Big)\ \theta\Big(i < C_k \wedge j < C_k \wedge i + j < C_k\Big)\right] \\
+ \quad & p_{ij} \left[\rho_k\, \theta\Big(i > -C_{k-1}\Big) + \sigma_k\, \theta\Big(j > -C_{k+1}\Big)\right] \\
= \quad & \mu_k\Big(G^k_{i,j-1}\Big)\, p_{i,j-1}\, \theta\Big(j > 0\Big) \\
+ \quad & \mu_k\Big(G^k_{i-1,j-1}\Big)\, p_{i-1,j-1}\, \theta\Big(j \leq 0 \wedge j > -C_{k+1} \wedge i > -C_{k-1}\Big) \\
+ \quad & \sigma_k \left[p_{i-1,j+1}\, \theta\Big(j \geq 0 \wedge j < C_k \wedge i > C_{k-1}\Big) + p_{i,j+1}\, \theta\Big(j < 0\Big)\right] \\
+ \quad & \rho_k p_{i+1,j}\, \theta\Big(i < C_k \wedge i + j < C_k\Big), \qquad (i,j) \in S_k
\end{aligned} \tag{10}$$

We define $G^k_{i,j} = C_k - i\,\theta(i > 0) - j\,\theta(j > 0)$ as the number of lots present in the service station, i.e. in the flow equivalent server in state (i,j). The throughput of isolated stage k can be calculated as the mean number of served lots using the equation:

$$T_k = \sum_{n_k=1}^{C_k} \mu_k(n_k) P(G^k = n_k)$$

Numerical Results

To analyze the effects of increasing the number of kanbans assigned to each stage we observe a kanban controlled system consisting of three stages. The network and parameters of servers located in stage two are shown in Figure 6. There is one single server in stage 1 and 3 with exponentially distributed service rates with mean one.

As expected throughput, WIP, and cycle time are increased by increasing the number of kanbans (cf. Figures 7, 8, 9). The gain of throughput by adding kanbans is the less the more kanbans are already assigned. The growth of WIP and cycle time is almost linear to the number of kanbans. These results are conform to Mitra and Mitrani's studies. There is also a good conformation between simulation and analysis. The relative errors of the performance measures lie between one and nine percent, decreasing with increasing number of kanbans.

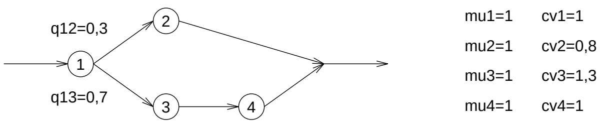

Figure 6: *Network*

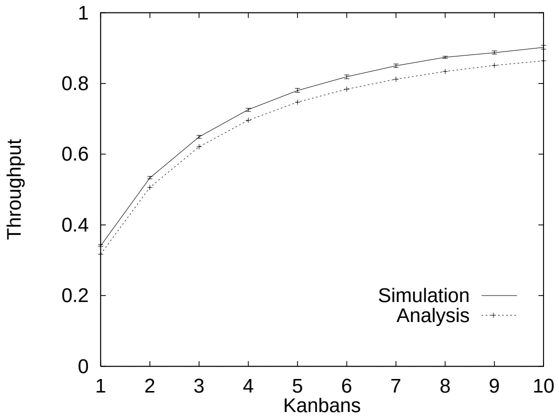

Figure 7: *Throughput*

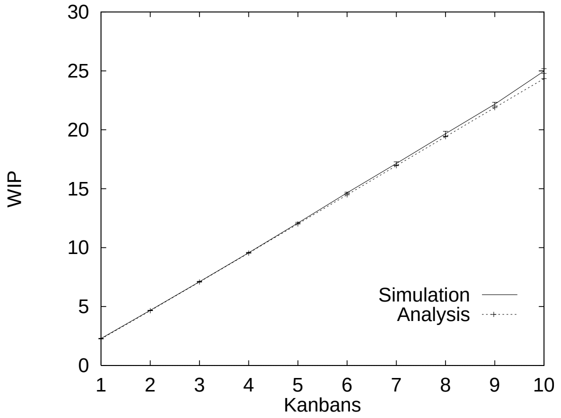

Figure 8: *WIP*

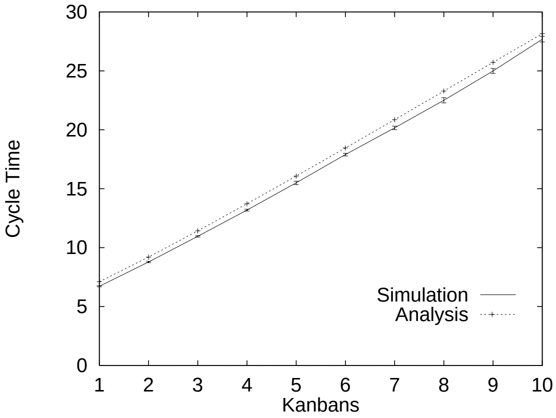

Figure 9: *Cycle Time*

4 Conclusion and Outlook

We showed how to extend the methodology of Mitra and Mitrani to model a kanban system with batch servers, assembly–like kanban systems, and systems with more than one machine within one stage. This was done by giving suitable balance equations for the three models and equations to calculate the throughput of the isolated cells. We illustrated the balance equations explaining the outlines of the state spaces and how transitions between different states can take place. We further introduced flow equivalent servers to reduce the complexity of a network's state space when we dealt with more than one machine within one stage.

Analytical results were validated by simulation, and we discussed how given parameters influence performance measures such as production rate (throughput), cycle time, and work in process. We found always a good conformation between analytical and simulative results with relative errors significant smaller than 10% for most parameter sets. Last but not least we showed that different kinds of unbalancing of the systems can reach a reduction of WIP and cycle time though throughput is almost held constant.

Further research has to be done to expand Di Mascolo's (et al.) model by batch service stations as well and to consider dualcard kanban systems.

Acknowledgement

The authors would like to thank Dr. O. Gihr from IBM Produktions GmbH, Sindelfingen, Germany and Dr. J. Fowler from SEMATECH, Austin, TX, U.S.A. for the motivation of this study and for stimulating discussions.

This work was supported by Deutsche Forschungsgemeinschaft (DFG), grant #Tr 257/2–1.

References

[1] Kenneth R. Baker, Stephen G. Powell, and David F. Pyke. Buffered and unbuffered assembly systems with variable processing times. *Journal of manufacturing and operations management*, 3:200–223, 1990.

[2] Kenneth R. Baker, Stephen G. Powell, and David F. Pyke. Optimal allocation of work in assembly systems. *Management Science*, 39(1):101–106, January 1993.

[3] Blair J. Berkley. A review of the kanban production control research literature. *Production and Operations Management*, 1(4):393–411, Fall 1992.

[4] M.L Chaudhry and J.G.C Templeton. *A First Course in Bulk Queues*. John Wiley & Sons, New York, NY, 1983.

[5] Yves Dallery. Approximate analysis of general open queueing networks. *Performance Evaluation*, 11:209–222, 1990.

[6] Maria Di Mascolo, Yannick Frein, and Yves Dallery. Queueing network modeling and analysis kanban systems. In *Third International Conference On Computer Intergrated Manufacturing*. IEEE / Rensselaer Polytechnic Institute, IEEE Computer Society Press, May 20–22 1992.

[7] Maria Di Mascolo, Yannick Frein, Yves Dallery, and René David. A unified modeling of kanban systems using petri nets. *The International Journal of Flexible Manufacturing Systems*, 3:275–307, 1991.

[8] John W. Fowler, Don T. Phillips, and Gary L. Hogg. Real-time control of multiproduct bulk-service semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, 5(2):158–163, May 1992.

[9] C. Roger Glassey and W. Willie Weng. Dynamic batching heuristics for simultaneous processing. *IEEE Transactions on Semiconductor Manufacturing*, 4(2):77–82, May 1991.

[10] Hermann Gold. *Performance Modelling of Batch Service Systems with Push and Pull manufacturing Management Policies*. PhD thesis, Universität Würzburg, 1992.

[11] Hermann Gold and Phuoc Tran-Gia. Performance analysis of a batch service queue arising out of manufacturing system modelling. *Queueing Systems*, 14:413–426, 1993.

[12] Damodar Y. Golhar and Carol Lee Stamm. The just-in-time philosophy. *The International Journal of Production Research*, 29(4):656–676, 1991.

[13] J. Michael Harrison. Assembly-like queues. *Journal of Applied Probability*, pages 354–367, 1973.

[14] Wallace J. Hopp and John T. Simon. Bounds and heuristics for assembly-like queues. *Queueing Systems*, 4:137–155, 1989.

[15] Pravin K. Johri. Practical issues in scheduling and dispatching in semiconductor wafer fabrication. *Journal of Manufacturing Systems*, 12(6):474–485, 1992.

[16] E. H. Lipper and B. Sengupta. Assembly-like queues with finite capacity: bounds, asymptotics, and approximations. *Queueing Systems*, 1:67–83, 1986.

[17] Raymond A. Marie. An approximate analytical method for general queueing networks. *IEEE Transactions on Software Engineering*, SE-5(5):530–538, September 1979.

[18] Raymond A. Marie. Calculating equilibrium probabilities for $\lambda(n)/c_k/1/n$ queues. *ACM Sigmetrics Performance Evaluation Review*, 9(2):117–125, 1980.

[19] Debasis Mitra and Isi Mitrani. Control and coordinaton policies for systems with buffers. *Performance Evaluation Review*, 17(1):156–164, May 1989.

[20] Debasis Mitra and Isi Mitrani. Analysis of a kanban discipline for cell coordination in production lines. *Management Science*, 36(12):1548–1566, December 1990.

[21] Charles H. Sauer and K. Mani Chandy. *Computer Systems Performance Modeling.* Prentice-Hall, Englewood Cliffs, NJ, 1981.

[22] Kut C. So and Steven C. Pinault. Allocating buffer storages in a pull system. *International Journal of Production Research*, 26(12):1959–1980, 1988.

[23] Phuoc Tran-Gia, Herman Gold, and Heidrun Grob. A batch service system operating in a pull production line. *Archiv für Elektronik und Übertragungstechnik*, 47(5/6):379–385, 1993.

[24] Hunglin M. Wang. *An analytical approach for planning and control of kanban systems for just-in-time manufacturing.* PhD thesis, University of Iowa, Iowa City, 1991.