Analysis of Loop Networks by Decomposition

Stanley B. Gershwin   Nicola Maggio   Andrea Matta   Tullio Tolio
Loren M. Werner

1 Introduction

A closed-loop production system or *loop* is a system in which a constant amount of material flows through a set of work stations and storage buffers alternately in a fixed sequence, and when it leaves the last buffer, it reenters the first machine. Figure 1 represents a K-machine loop. The purpose of this paper is to present a new analytical method for evaluating the production rate and the distribution of inventory.

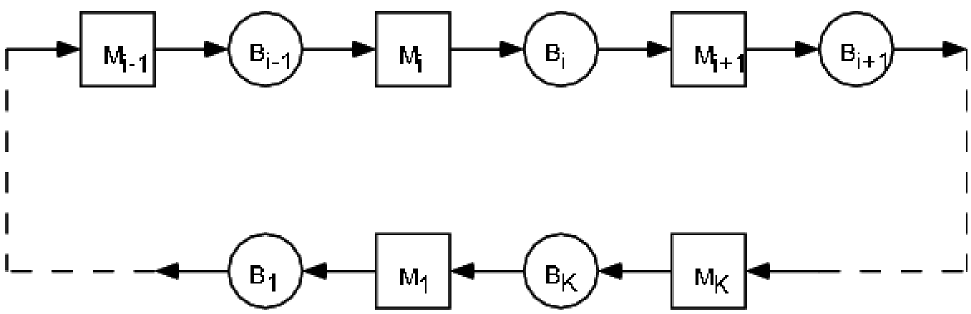

Figure 1: Illustration of a closed-loop production system

The first analytical method for evaluating the performance of closed-loop systems with finite buffers and unreliable machines was proposed in (Frein et al., 1996). This method is an extension of the decomposition method developed by (Gershwin, 1987). This method is only accurate for large loops because it does not account for the correlation between number of parts in the buffers. (Maggio, 2000) presents a decomposition method which does account for this correlation. However, the model is more complex and is not practical for loops with more than three machines. For a more detailed listing of previous work dealing with closed-loop systems, see (Maggio, 2000).

2 Closed-Loop Production Systems

2.1 Basic Model

We extend the deterministic processing time model presented in (Gershwin, 1994) to closed-loop systems. More specifically, we use the (Tolio and Matta, 1998) version, which allows machines to fail in more than one mode. Processing times for all machines are assumed to be deterministic and identical, and the operation time is the time unit. Machine failure and repair times are geometrically distributed.

M_i refers to Machine i. B_i is its downstream buffer and has capacity N_i. A machine is *blocked* if its downstream buffer is full and *starved* if its upstream buffer is empty. When M_i is *working* (operational and neither blocked nor starved) it has a probability p_{ij} of failing in mode j in one time unit. If M_i is down in mode j, it is repaired in a given time unit with probability r_{ij}. By convention, machine failures and repairs take place at the beginnings of time units and changes in buffer levels occur at the ends of time units.

2.2 Transfer Line Decomposition Techniques

Accurate decomposition methods have been developed for evaluating long transfer lines (Gershwin, 1994). These methods decompose a K-machine transfer line into $K-1$ two-machine lines or *building blocks*. In each building block $L(i)$, the buffer $B(i)$ corresponds to B_i in the original transfer line. The upstream machine $M^u(i)$ represents the collective behavior of the line upstream of B_i and the downstream machine $M^d(i)$ represents the behavior downstream.

To an observer sitting in $B(i)$, $M^u(i)$ appears to be down when M_i is either down or starved by some upstream machine. $M^u(i)$ is said to have *real* failure modes corresponding to those of M_i and *virtual* failure modes corresponding to each of the upstream machines (Tolio and Matta, 1998). Likewise, $M^d(i)$ has *real* failure modes corresponding to those of M_{i+1} and *virtual* failure modes corresponding to each of the downstream machines.

In order to estimate the system performance, we must find values of the virtual failure probabilities, $p^u_{k,j}(i)$ and $p^d_{k,j}(i)$, parameters of $M^u(i)$ and $M^d(i)$. $p^u_{k,j}(i)$ is the observer's estimate of the probability of machine $M^u(i)$ failing in mode (k,j). The goal of the decomposition equations is to determine the parameters of $M^u(i)$ and $M^d(i)$ such that the flow of parts through $B(i)$ mimics that through B_i.

Although the observer does not know this, mode (k,j) corresponds to mode j of machine M_k. If M_k is upstream of M_i, this is a virtual mode. If $k=i$ this is a real mode. (Similarly for $M^d(i)$.) The concepts of *range of starvation* and *range of blocking* (Section 2.6) eliminate the ambiguity of "upstream" and "downstream" in a loop.

2.3 Special Characteristics of Closed-Loop Systems

In a transfer line, blocking and starvation can propagate throughout the entire system. If the first machine fails, it is possible for all of the downstream machines to become starved. Similarly, if the last machine fails, all upstream machines can become blocked.

This is not the case in loops. Whether or not a machine can be starved or blocked by the failure of another machine depends on the number of parts in the system and the total buffer space between the two machines. For ease of notation, we define all subscripts to be modulo K. In particular, we define the set of integers (i,j) as:

$$(i,j) = \begin{cases} (i, i+1, ..., j) & \text{if } i < j \\ (i, i+1, ..., K, 1, ..., j) & \text{if } i > j \end{cases} \tag{1}$$

We define N^p to be the total number of parts in the system and $\Psi(v,w)$ as the total buffer capacity between M_v and M_w in the direction of flow (Maggio et al., 2000). More formally,

$$\Psi(v,w) = \begin{cases} \displaystyle\sum_{z \in (v,w-1)} N_z & \text{if } v \neq w \\ 0 & \text{if } v = w \end{cases} \tag{2}$$

Note that if $v \neq w$, the total buffer space is given by

$$\Psi(v,w) + \Psi(w,v) = \sum_{z\in(v,w-1)} N_z + \sum_{z\in(w,v-1)} N_z = \sum_{z\in(v,v-1)} N_z = N^{total}$$

If $N^p < \Psi(v,w)$, then the failure of M_w can never cause M_v to become blocked because there are not enough parts in the system to fill all buffers between M_v and M_w simultaneously. Conversely, if $N^p > \Psi(v,w)$, M_w cannot starve M_v.

2.4 Thresholds

Whether or not a machine can ever be starved or blocked by the failure of a specific other machine may depend on the number of parts in an adjacent buffer (Maggio et al., 2000). We define the *threshold* $l_k(i)$ to be the maximum level of B_i such that all buffers between M_{i+1} and M_k can become full at the same time. Alternately, we can think of $l_k(i)$ as the maximum level of B_i such that the failure of M_k can cause M_{i+1} to become blocked.

Consider building block i, shown in Figure 2, and assume that machine M_{i+1} can be blocked by machine M_k. This means that $M^d(i)$ has a virtual failure of type (k,j) (where j indicates one of the failure modes of machine M_k, $j = 1, \dots, F_k$) with probability $p^d_{kj}(i)$. Because of the population constraint, if buffer B_i has too many parts, the remaining parts cannot fill all the buffers between M_{i+1} and M_k. Therefore, if the level of the buffer is greater than threshold $l^d_k(i)$, then a failure of M_k cannot cause blocking of machine M_{i+1} so $p^d_{kj}(i)$ is 0. Generalizing, machine M_k could produce blocking on machine M_{i+1} and therefore it could affect $M^d(i)$ only if the level of buffer $B(i)$ is less than or equal to $l^d_k(i)$. A similar argument holds for the starvation of M_i. Machine M_z could produce starvation at machine M_i (and therefore it could affect $M^u(i)$) only if the level of buffer $B(i)$ is greater than or equal to $l^u_z(i)$.

The threshold $l^d_k(i)$ is the largest number of parts in buffer B_i that allows all the buffers between M_{i+1} and M_k to be full at the same time. In that case all the buffers between M_k and M_i would be empty. Similarly, threshold $l^u_k(i)$ represents the smallest number of parts that allows all the buffers between M_k and M_i to be empty. Therefore, we can write $l^u_k(i) = l^d_k(i) = l_k(i)$. The value of the threshold does not depend on the failure type but only on the buffer capacities between machines M_{i+1} and M_k and on the number of parts in the system N^p.

Let $n(t)$ indicate the number of parts in buffer $B(i)$ at time t. Then,

- if $n(t) < l_k(i)$ then $M^u(i)$ cannot be down in virtual mode due to a failure j of machine M_k. Therefore $n(t) < l_k(i) \longrightarrow p^u_{kj}(i) = 0$.
- if $n(t) \geq l_k(i)$, $p^u_{kj}(i) \neq 0$ has to be determined.
- $n(t) > l_k(i) \longrightarrow p^d_{kj}(i) = 0$
- if $n(t) \leq l_k(i)$, $p^d_{kj}(i) \neq 0$ has to be determined.

Since $p^u_{kj}(i)$ and $p^d_{kj}(i)$ depend on $n(t)$, the two-machine line is much more complicated than earlier versions. Worse, there may be more than one threshold in buffers of loops having more than three machines.

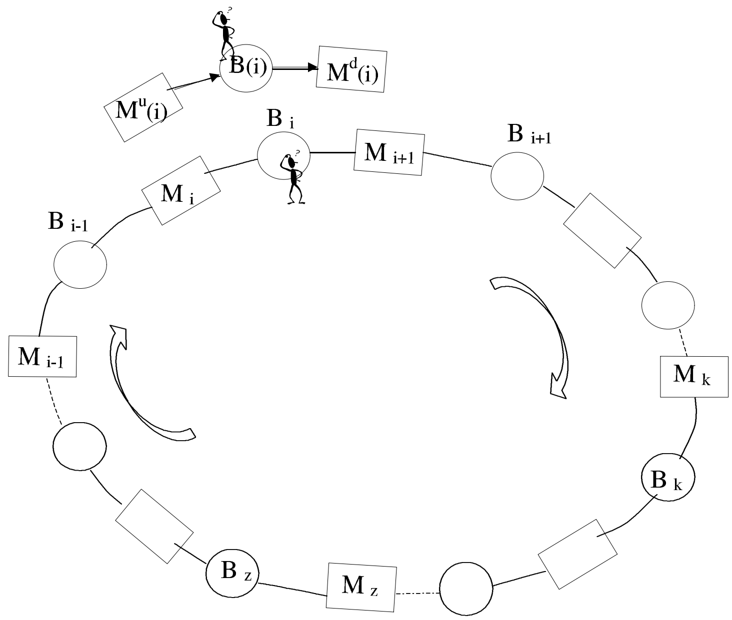

Figure 2: A loop system and one building block

2.5 Loop Transformation

It is possible to eliminate the complications in the two-machine building blocks due to thresholds by using a transformation procedure. The transformation allows us to evaluate much larger loops for a wider range of population levels and buffer sizes than is possible using the method presented in (Maggio et al., 2000).

For each threshold $0 < l_k(i) < N_i$, we insert a perfectly reliable machine M_{k^*} into buffer B_i such that $\Psi(k^*, k) = N^p$. B_i is now replaced by a buffer of size $N_i - l_k(i)$ followed by M_{k^*} followed by a buffer of size $l_k(i)$. Since each unreliable machine can cause at most one threshold between zero and N_i, the transformed loop will consist of at most $2K$ machines. Although the loop is larger, we can now use the same building block that is used in Tolio's transfer line decomposition — without the complication of thresholds.

2.6 Fixed Population Considerations

Once the loop is transformed, we account for the limited propagation of blocking and starvation due to a fixed population level by defining the *range of starvation* and *range of blocking*. The range of starvation of B_i is $\{M_{s(i)}, M_{s(i)+1}, ..., M_i\}$, where $M_{s(i)}$ is the machine farthest upstream which can cause B_i to become empty if it is failed for a long period of time. Similarly, the range of blocking of B_i is $\{M_{i+1}, M_{i+2}, ..., M_{b(i)}\}$, where $M_{b(i)}$ is the machine farthest downstream which can cause B_i to become full. We calculate $M_{s(i)}$ and $M_{b(i)}$ as follows:[1]

$$M_{s(i)} = \min_j M_{i+j} \qquad s.t. \quad \Psi(i, i+j) \quad > N^p \tag{3}$$

[1]Note that the inequalities are strict. We use this convention to deal with the situation of simultaneous blocking and starvation.

$$M_{b(i)} = \max_j M_{i+j+1} \quad s.t. \;\; \Psi(i+1, i+j+1) < N^p \tag{4}$$

The loop population is incorporated into the model by including in the building blocks only those virtual failure modes related to machines within the range of blocking and range of starvation. $M^u(i)$ has virtual failure modes corresponding only to the failure modes of $M_{s(i)}$ through M_{i-1}. Likewise, $M^d(i)$ has virtual failure modes corresponding to M_{i+2} through $M_{b(i)}$.

Simultaneous blocking and starvation If $\Psi(v, w) = N^p$ then machine M_v can become simultaneously blocked and starved when M_w is down for a long period of time. This is the case where the threshold $l_w(v-1) = 0$ and $l_w(v) = N_v$. In transformed loops, this situation can occur at each reliable machine M_{k^*} when M_k fails since $\Psi(k^*, k) = N^p$ by construction.

The two-machine building block developed in (Tolio et al., 2001) does not account for the states where both machines are down and the buffer level is either zero or N. Rather than modifying the building block, we associate the zero buffer level case with an upstream failure and the N buffer level case with a downstream failure.

3 Loop Decomposition

3.1 The Building Block Parameters

As shown in (Maggio et al., 2000), the failure and repair probabilities for the real failure modes are equal to the probabilities of the corresponding modes of the machines in the loop. Therefore, we have

$$p^u_{ij}(i) = p_{ij} \tag{5}$$

$$r^u_{ij}(i) = r_{ij} \tag{6}$$

$$p^d_{i+1,j}(i) = p_{i+1,j} \tag{7}$$

$$r^d_{i+1,j}(i) = r_{i+1,j} \tag{8}$$

In addition, we know that the probability of repair when a machine is down in virtual failure mode (k, j) is equal to the probability that machine M_k is repaired when it is down in failure mode j. This gives us

$$r^u_{kj}(i) = r_{kj} \tag{9}$$

$$r^d_{kj}(i) = r_{kj} \tag{10}$$

To evaluate the performance measure of the loop, we must find the virtual failure probabilities $p^u_{kj}(i)$ and $p^d_{kj}(i)$ for each $L(i)$. This is the objective of solving the decomposition equations.

3.2 Decomposition Equations

The decomposition equations are nearly identical to the transfer line decomposition equations presented in (Tolio and Matta, 1998). In fact, we need only modify the indices to account for the range of blocking and starvation and the fact that loops contain as many buffers as machines.

We define $P_{kj}^{st}(i)$ as the probability that $B(i)$ is empty due to $M^u(i)$ being down in virtual failure mode (k, j). Likewise, $P_{kj}^{bl}(i)$ is the probability that $B(i)$ is full due to $M^d(i)$ being down in virtual failure mode (k, j). Finally, we define $E(i)$ to be the average throughput of building block $L(i)$. Using this notation, we write the decomposition equations. Recall from Section 2.6 that $s(i)$ is the number of the machine furthest upstream which falls within the range of starvation of buffer B_i. Similarly, $b(i)$ is the number of the machine furthest downstream which falls within the range of blocking of B_i. Then,

For $i = 1$ to K', $k = s(i)$ to $i - 1$, $\forall j$:

$$p_{kj}^u(i) = \frac{P_{kj}^{st}(i-1)}{E(i)} r_{kj} \tag{11}$$

For $i = 1$ to K', $k = i + 2$ to $b(i - 1)$, $\forall j$:

$$p_{kj}^d(i) = \frac{P_{kj}^{bl}(i+1)}{E(i)} r_{kj} \tag{12}$$

We know the values of r_{kj} from the parameters of the machines in the transformed loop. By solving the building block transition equations presented in (Tolio and Matta, 1998) for each $L(i)$, we can find the values of $P_{kj}^{st}(i)$, $P_{kj}^{bl}(i)$, and $E(i)$, which are functions of $p_{kj}^u(i)$ and $p_{kj}^d(i)$. The decomposition equations (11) and (12) represent a system of $2K'$ independent equations in $2K'$ unknowns.

4 Implementing the Loop Transformation and Decomposition

This section provides a step-by-step procedure for evaluating a loop. First, we transform an arbitrary loop into one without thresholds. Next, we introduce a set of decomposition equations and an algorithm which is a slight modification of that of (Tolio and Matta, 1998) for solving them.

4.1 The Transformation Algorithm

1. For all $N_i > N^p$, set $N_i = N^p$. For all $N_i > N^{total} - N^p$, set $N_i = N^{total} - N^p$ (In this case we add $N_i - N^{total} - N^p$ to the resulting average buffer level in order to recover the true average buffer level for B_i).

2. Insert a perfectly reliable machine M_{i*} for every unreliable machine M_i such that $\Psi(i*, i) = N^p$ (unless a machine M_j such that $\Psi(j, i) = N^p$ already exists). The new loop consists of K' machines separated by K' buffers. Note that the size of the buffers may be different from the original buffers, but the total buffer space in the transformed loop is equal to that of the original.

3. Re-number the machines and buffers from 1 to K'.

4. For all $N_i = 1$, set $N_i = 2$ (see Section 5.2).

4.2 The Decomposition Algorithm

1. Calculate the range of starvation and range of blocking for each $L(i)$ using (3) and (4) to determine all possible failure modes.

2. Initialize $p^u_{ij}(i)$, $r^u_{ij}(i)$, $p^d_{i+1,j}(i)$, $r^d_{i+1,j}(i)$, $r^u_{kj}(i)$, and $r^d_{kj}(i)$ for all valid failure modes using equations (5), (6), (7), (8), (9), and (10). Set $p^u_{kj}(i) = p_{kj}$ and $p^d_{kj}(i) = p_{kj}$.

3. For $i = 1$ to K':
   – Calculate $E(i)$ and $P^{st}_{kj}(i)$.
   – Update $p^u_{kj}(i+1)$ using (11).

4. For $i = K'$ to 1:
   – Calculate $E(i)$ and $P^{bl}_{kj}(i)$.
   – Update $p^d_{kj}(i-1)$ using (12).

5. Repeat 3 and 4 until the parameters converge to an acceptable tolerance.

5 Performance of the Method

5.1 Three-Machine Loops

Numerous experiments on three-machine loop systems have been carried out in order to test the accuracy, convergence and speed of the new solution technique (Maggio, 2000), (Maggio et al., 2000). We take an average of the throughputs of each building block and compare the results with those obtained running a discrete event simulation. To ensure statistical significance, the length of simulation was chosen to be 10,000,000 time units. The first set of experiments considered *symmetrical* systems, in which the three machines are identical and all the buffers have the same capacity. The second set of experiments considered systems with identical machines but different buffer sizes. The third set examined a loop with different machines and identical buffers and the last set dealt with a completely asymmetrical closed line.

From the comparison of the analytical results with those from simulation, the method seems to be accurate. The error for the throughput was always less than 1.24%. The error in the average buffer level was almost always less than 3%, but it increased when the machines were not balanced or the buffers were different. Furthermore, the decomposition procedure converged in every case.

5.2 Larger systems

The method was tested extensively on three- to ten-machine loops with machine parameters and buffer sizes generated randomly. For each loop, the decomposition and simulation were performed for all possible population levels. Here, we examine the accuracy, convergence reliability, and speed of the method.

Comparison with Simulation The method gives extremely accurate (error of less than 1 percent) approximations of average throughput when the number of parts (or holes) is greater than the number of machines and/or the size of the smallest buffer. Average buffer level errors were normally less than 1 percent in this range, but in some cases the errors were as high as 6 percent. Details will be found in (Werner, 2001).

Buffers of Size One It is possible that the original loop may contain a buffer of size one. Depending on the population of the loop and the size of the buffers, the transformation may also create a buffer of size one. However, the two-machine line model we use does not allow buffers to have size one. Therefore, we replaced all buffers of size one with buffers of size two. While this approach is somewhat arbitrary, it results in a better approximation of average throughput and has only a negligible impact on average buffer levels.

Convergence Reliability In all cases studied, the decomposition algorithm converged. The convergence criterion was that the difference in all $E(i)$s between successive iterations be less than 10^{-6}. The decomposition algorithm does not exactly satisfy conservation of flow even though Tolio's equations imply that it should. However, the differences between the throughputs of the building blocks are generally very small.

Speed Since the algorithm is nearly identical to Tolio's transfer line decomposition algorithm, the computing speed will be comparable.

6 Observations on Loop Behavior

6.1 Transfer Line Flatness

This special type of flatness occurs in loops where the capacity of the largest buffer is greater than the sum of the capacities of the other buffers. For all population levels N^p such that $N^{total} - N^{max} < N^p < N^{max}$, the throughput is constant.

To illustrate the concept of transfer line flatness, we consider a three-machine loop with buffers of size 10, 5, and 22 (see Figure 3). When there are 16 parts in the system, it is possible for buffers B_1 and B_2 to be both full and empty. However, B_3 can never become full or empty. This means that machine M_1 can never be starved and M_3 can never be blocked. If we ignore B_3, the system has the same production rate and average buffer levels as a transfer line consisting of M_1, B_1, M_2, B_2, and M_3. This behavior remains the same for populations up to 21 because in each of these cases M_1 is never starved and M_3 is never blocked. In this population range, the average throughput remains constant (Figure 4).

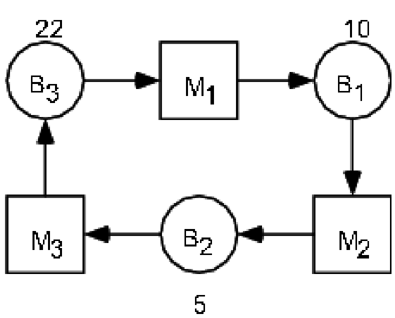

Figure 3: Example of loop with transfer line flatness

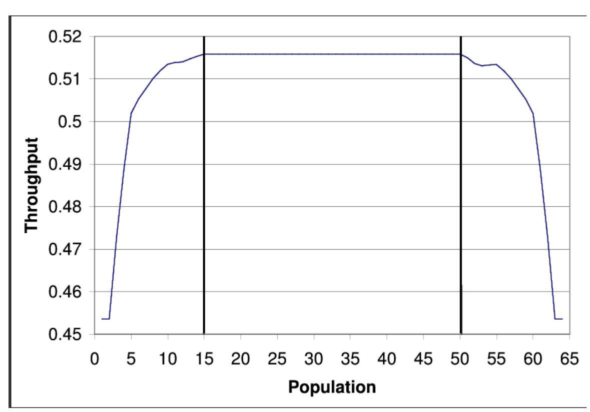

Figure 4: Analytical throughput as a function of population

Machine	1	2	3	4
r	—	.1	.1	.1
p	.01	.01	.01	.01

Buffer	1	2	3	4
Size	10	10	10	10

Population	15

Table 1: Parameters of four-machine loop

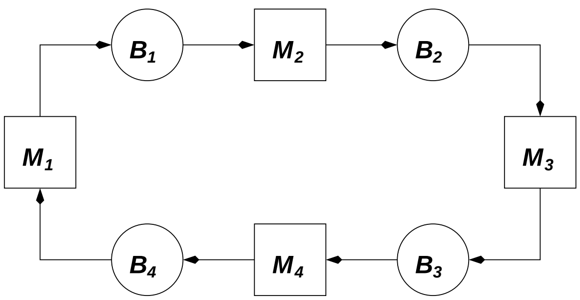

Figure 5: Four-machine loop

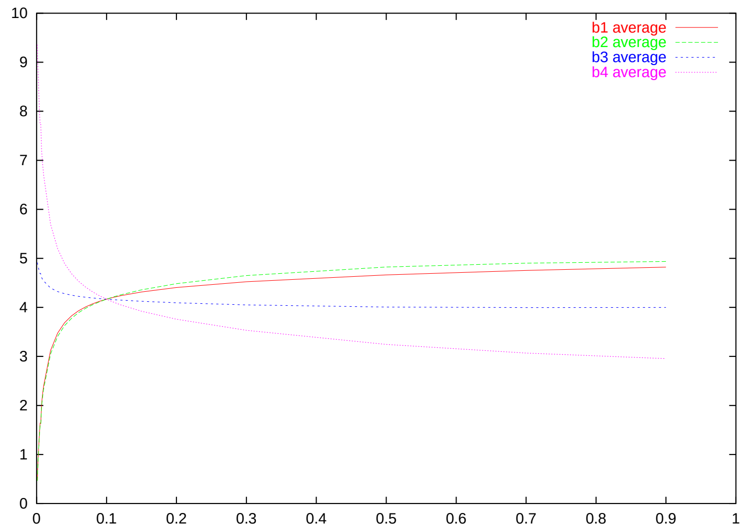

Figure 6: Average buffer levels

6.2 Near Flatness and Non-Flatness

We also observed a type of flatness we call near flatness. It occurs in loops that do not meet the requirements for transfer line flatness, but have population ranges where the throughput is nearly constant. In the cases we studied, loops which were very asymmetrical did exhibit near flatness. *Symmetrical* loops, in which the machines are identical and the buffer capacities are the same, did not seem to exhibit near flatness. The degree of flatness seemed to increase with the degree of asymmetry in the loop.

6.3 Balance and Imbalance

Figure 5 shows a four-machine loop whose parameters are listed in Table 1. All parameters are fixed except r_1, the repair probability of M_1. In the following, we examine the effect of varying r_1 on the production rate and the distribution of inventory. When r_1 is small, M_1 is the bottleneck of the system. Figure 6 shows how the average buffer levels vary with r_1. The four curves intersect when $r_1 = .1$ because the line is completely symmetric.

When $r_1 \to 0$, $\bar{b}_4 \to 10$, $\bar{b}_3 \to 5$, $\bar{b}_2 \to 0$, and $\bar{b}_1 \to 0$. This is because M_1 completely blocks the system. All the inventory accumulates upstream of M_1. Since there is only room for 10 parts in B_4, the rest of the inventory appears in B_3, and there is none elsewhere.

7 Conclusions and Future Work

The purpose of this paper was to summarize (Maggio, 2000) and (Maggio et al., 2000) and to describe a transformation that makes this work practical for larger systems.

There are several opportunities for future research:

1. Reduce discrepancies in conservation of flow.
2. Extend to very small and very large populations.
3. Reduce errors in average buffer levels.
4. Extend to multiple loop systems.
5. Extend to multiple part type loops.
6. Extend to multiple loop systems with multiple part types.

References

Frein, Y., Commault, C., and Dallery, Y. (1996). Modeling and analysis of closed-loop production lines with unreliable machines and finite buffers. *IIE Transactions*, 28:545–554.

Gershwin, S. B. (1987). An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations Research*, 35:291–305.

Gershwin, S. B. (1994). *Manufacturing Systems Engineering*. Prentice-Hall, Inc.

Maggio, N. (2000). An analytical method for evaluating the performance of closed loop production lines with unreliable machines and finite buffer. Master's thesis, Politecnico di Milano.

Maggio, N., Matta, A., Gershwin, S., and Tolio, T. (2000). An analytical method for the performance evaluation of closed loop production lines with multiple unreliable machines and finite buffers. Technical report, Politicnico di Milano - Dipartimento di Meccanica.

Tolio, T., Gershwin, S. B., and Matta, A. (2001). Analysis of two-machine lines with multiple failure modes. *IIE Transactions*. to appear.

Tolio, T. and Matta, A. (1998). A method for performance evaluation of automated flow lines. *Annals of the CIRP*, 47.

Werner, L. (2001). *in preparation*. Master's thesis, MIT OR Center.