A Model for Two-stage Manufacturing Systems

Wai Ki Ching

Department of Applied Mathematics
The Hong Kong Polytechnic University, Hung Hom, Hong Kong.
and
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong, Shatin, Hong Kong.
Research supported by HKRGC Grant 489/95E.
wkching@se.cuhk.edu.hk

Abstract

This paper studies a two-stage (Two machines in tandem) manufacturing systems under hedging point production policy. The manufacturing system produces one type of product and its demand is modeled as a Poisson process. Each product has to go through the manufacturing process by a reliable machine in each stage. Preconditioned Conjugate Gradient (PCG) method is employed to solve the steady state probability distribution of system. Preconditioner is constructed by taking circulant approximation of the generator matrix of the system. We prove that the preconditioned linear system has singular values clustered around one when the number of inventory levels tends to infinity. Hence conjugate gradient methods will converge very fast when apply to solving the preconditioned linear system. Numerical examples are given to verify our claim. The average running cost can be written in terms of this probability. By varying different possible values of hedging point h, the optimal value of h which minimizes the average running cost can be obtained. Extension to multiple parallel machines in each stage is also discussed.

Key Words: Manufacturing System, Hedging Point Policies, Steady State Distribution, Preconditioner, Conjugate Gradient Method.

1 Introduction

In this paper, we study a two-stage manufacturing systems. The system consists of two reliable machines (one in each stage) producing one type of product. Each product has to go through the manufacturing processes in the two stages before it is finished. We assume infinite supply of raw material for the manufacturing process of the first machine. The mean processing time for one unit of product in the first and second machine are exponentially distributed with parameters $1/\mu_1$ and $1/\mu_2$ respectively. A buffer B_1 of size b_1 is placed between the two machines to store the products which are finished by the first machine and waiting for further operations by the second machine. Finished products are put in a buffer B_2 of maximum size b_2. The inter-arrival time of a demand is assumed to be exponentially distributed with parameter $1/\lambda$. Finite backlog of finished product is allowed in the system. The maximum allowable backlog of product is m. When the inventory level of the finished product is $-m$, any arrival demand will be rejected. Hedging point policy is employed as the inventory control in both buffers B_1 and B_2. It is well known that the hedging point policy is optimal for one machine (one stage) manufacturing systems in some simple situations, see [1] for instance. When the optimal policy is a zero-inventory policy (i.e. the hedging point is zero), then the policy matches with the just-in-time (JIT) policy. The JIT policies have strongly been favored in real-life production systems for process discipline reasons even when they are not optimal. By using the JIT policy, the TOYOTA company can manage to reduce work-in-process and cycle time, see [10]. We focus ourselves in hedging point policies for our manufacturing systems. The hedging point policy is characterized by an integer number h. The machine keeps on producing products if the inventory level is less than h and the first buffer is not empty, otherwise the machine is shut down. For the first machine, the hedging point h is b_1 and inventory level is non-negative, i.e. $m = 0$. However, the inventory level of buffer B_2 can be negative, because we allow a maximum backlog of finished product of m. Very often the buffer size and the inventory levels of B_2 are much larger than that of B_1. Inventory cost and backlog cost can be written in terms of the steady state probability of the system, see [3, 4, 5, 2, 6] for instance.

The inventory levels are modeled as Markovian processes. It turns out that the process is an irreducible continuous time Markov chain. We give the generator matrix for the process, and Preconditioned Conjugate Gradient (PCG) method is employed to compute the steady state probability distribution. Preconditioners are constructed by taking circulant approximation of the generator matrix of the system. We prove that if the parameters μ_1, μ_2, λ and b_1 are fixed and independent of $n = m + h + 1$, then the preconditioned linear system has singular values clustered around one as n tends to infinite. Hence the Conjugate Gradient (CG) type methods will converge very fast when apply to solving the preconditioned linear system. Numerical examples are given in §5 to verify our claim.

The remainder of this paper is organized as follows. In §2, we formulate the manufacturing system and give the generator matrix of the corresponding continuous time Markov chain. In §3, preconditioner is constructed for the generator matrix. In §4, we prove that the preconditioned linear system has singular values clustered around one. In §5, we give a cost analysis for our method and numerical examples are given to demonstrate the fast convergence rate of our method. Concluding remarks are given to discuss the extension of multiple parallel machines in each stage in §6.

2 The Manufacturing System

In this section, we construct the generator matrix for the manufacturing system. Let us define the following system parameters.

(i) $1/\lambda$, the mean inter-arrival time of a demand,

(ii) $1/\mu_1$, the mean unit processing time of the first machine,

(iii) $1/\mu_2$, the mean unit processing time of the second machine,

(iv) b_1, buffer size for the first machine,

(v) b_2, maximum buffer size for the finished products,

(vi) h, the hedging point,

(vii) m, the maximum allowable backlog,

(viii) c_{I_1}, unit inventory cost for the first buffer B_1,

(ix) c_{I_2}, unit inventory cost for the second buffer B_2,

(x) c_B, unit backlog cost of the finished products.

We note that the inventory level of the first buffer cannot be negative or exceeds the buffer size b_1. Thus the total number of inventory levels in the first buffer is $b_1 + 1$. For the second buffer, under the hedging point policy, the maximum possible inventory level is $h(h \leq b_2)$. Since we allow a maximum backlog of m in the system, the total number of possible inventory levels in the second buffer is $n = m+h+1$. In practice the value of n can easily go up to thousands.

We let $z_1(t)$ and $z_2(t)$ be the inventory levels of the first and second buffer at time t respectively. Then $z_1(t)$ and $z_2(t)$ take integer values in $[0, b_1]$ and $[-m, h]$ respectively. Thus the joint inventory process

$$\{(z_1(t), z_2(t)), t \geq 0\}$$

is a continuous time Markov chain taking values in the state space

$$S = \{(z_1(t), z_2(t)) : z_1 = 0, \cdots, b_1,\ z_2 = -m, \cdots, h.\}.$$

Each time when visiting a state, the process stays there for a random period of time that has an exponential distribution and is independent of the past behavior of the process. The steady state probability of the system:

$$\lim_{t\to\infty} \text{Prob}\{(z_1(t), z_2(t)) = (i, j)\} = p(i, j),$$

$$\text{for} \quad i = 0, \cdots, b_1;\ j = -m, \cdots, h.$$

We order inventory states lexicographically, according to z_1 first and then z_2 and the tridiagonal block generator A for the joint inventory system can be obtained as follows: $A =$

$$\begin{bmatrix} \Lambda+\mu_1 I_n & \Sigma & & & \\ -\mu_1 I_n & \Lambda+D+\mu_1 I_n & \Sigma & & \\ & \ddots & \ddots & & \\ & -\mu_1 I_n & & \Lambda+D+\mu_1 I_n & \Sigma \\ & & & -\mu_1 I_n & \Lambda+D \end{bmatrix},$$

where

$$\Lambda = \begin{bmatrix} 0 & -\lambda & & 0 \\ & \lambda & \ddots & \\ & & \ddots & -\lambda \\ 0 & & & \lambda \end{bmatrix}, \tag{2}$$

$$\Sigma = \begin{bmatrix} 0 & & & 0 \\ -\mu_2 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -\mu_2 & 0 \end{bmatrix}, \tag{3}$$

I_n is the $n \times n$ identity matrix and D is the $n \times n$ diagonal matrix $D = \text{Diag}(\mu_2, \cdots, \mu_2, 0)$. The generator matrix A is similar to the one for the two-station queueing system discussed in [11]. We are interested in the solving the steady state probability distribution $\mathbf{p}$ of the generator matrix A. Many useful quantities such as the throughput of the system

$$(1 - \sum_{j=-m}^{h} p(0, j))\mu_2$$

and mean number of products in buffer B_1 and B_2 (work-in-process)

$$\sum_{i=1}^{b_1}[\sum_{j=-m}^{h} p(i,j)]i \quad \text{and} \quad \sum_{j=1}^{h}[\sum_{i=0}^{b_1} p(i,j)]j$$

can be written in terms of $\mathbf{p}$, see Siha [11]. Furthermore the average running cost and the average profit of the system can be also written in terms of $\mathbf{p}$ as follows:

$$C_{I_1}\sum_{i=1}^{b_1}[\sum_{j=-m}^{h} p(i,j)]i + C_{I_2}\sum_{j=1}^{h}[\sum_{i=0}^{b_1} p(i,j)]j +$$

$$C_B\sum_{j=1}^{m}[\sum_{i=0}^{b_1} p(i,-j)]j,$$

see Ching, Chan and Zhou [3, 4, 5, 6] for instance.

We note that the generator A is irreducible, has zero column sum, positive diagonal entries and non-positive off-diagonal entries, so $((I-A)\cdot(\text{Diag}(A))^{-1})$ is a stochastic matrix. Here $\text{Diag}(A)$ is the diagonal matrix containing the diagonal entries of the matrix A. From the Perron and Frobenius theory, we know that A has a one-dimensional null-space with a right positive null vector, see Varga [13, p. 30]. The steady state probability distribution $\mathbf{p}$ is then equal to the normalized form of the positive null vector. Since A is singular, we consider an equivalent linear system

$$G\mathbf{x} \equiv (A + \mathbf{f}\mathbf{f}^t)\mathbf{x} = \mathbf{f}, \tag{4}$$

where $\mathbf{f} = (0, \cdots, 0, 1)^t$ is the $(b_1+1)(m+h+1)$ unit vector. The following lemma shows that the linear system (4) is non-singular and hence the steady state probability distribution can be obtained by normalizing the solution of equation (4), see Chan and Ching [2, 6].

Lemma 1 *The matrix G is non-singular.*

Proof: Since the matrix G is an irreducible matrix, by applying the Gerschgorin circle theorem (see Horn and Johnson [9, p.346]) to the matrix G, we see that all the eigenvalues has non-negative real part. Moreover, by applying Theorem 1.7 in [13, p.20] to the matrix G, we know that zero cannot be an eigenvalue of G. Thus the matrix G is non-singular. □

However, the analytic solution of $\mathbf{p}$ is not generally available. Therefore, most of the techniques employed for the analysis are analytical approximations and numerical solutions. Yamazaki [15] gave an analytical approximation for the two-station queueing model under the assumption of infinite buffer size. Our manufacturing system deals with the realistic situation of finite buffer size. Usually, by making use of the block structure of the generator matrix A, classical iterative methods such as block Gauss-Seidel is applied in solving the steady state probability distribution to save computational cost, see [11]. However, in general the classical iterative methods have slow convergence rate, see the numerical results in §5. We employ Conjugate Gradient (CG) type methods to solve the steady state probability distribution. To speed up the convergence, a preconditioner C is introduced. We solve the following preconditioned system

$$GC^{-1}\mathbf{y} = \mathbf{f} \tag{5}$$

instead of $G\mathbf{x} = \mathbf{f}$. Obviously we have $\mathbf{x} = C^{-1}\mathbf{y}$. A good preconditioner C is a matrix such that it is easy to construct, the preconditioned matrix GC^{-1} has clustered singular values around one. The preconditioned system $C\mathbf{z} = \mathbf{r}$ is easy to solve for any right hand side vector $\mathbf{r}$.

3 Construction of Preconditioners

In this section, we construct preconditioner by taking the circulant approximation of blocks Λ, Σ and D of A. It is well known that any $n \times n$ circulant matrix C_n is characterized by its first column (or the first row) and can be diagonalized by the discrete Fourier matrix F_n, i.e. $C_n = F_n\Omega_n F_n^*$, where the entries of F_n are given by

$$[F_n]_{j,k} = \frac{1}{\sqrt{n}} e^{\frac{(2jk\pi)i}{n}}, \quad j,k = 0,1,\cdots,n-1,$$

F_n^* is the conjugate transpose of F_n and Ω_n is a diagonal matrix containing the eigenvalues of C_n. The matrix-vector multiplication of the forms $F_n\mathbf{y}$ and $F_n^*\mathbf{y}$ can be obtained in $O(n\log n)$ operations by the Fast Fourier Transform (FFT). By completing Λ, Σ and D to circulant matrices, we define the circulant approximation $c(\Lambda)$, $c(\Sigma)$ and $c(D)$ as follow:

$$c(\Lambda) = \begin{bmatrix} \lambda & -\lambda & & 0 \\ & \lambda & \ddots & \\ & & \ddots & -\lambda \\ -\lambda & & & \lambda \end{bmatrix}, \tag{6}$$

$$c(\Sigma) = \begin{bmatrix} 0 & & & -\mu_2 \\ -\mu_2 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -\mu_2 & 0 \end{bmatrix} \tag{7}$$

and

$$c(D) = \text{Diag}(\mu_2, \cdots, \mu_2, \mu_2).$$

From (6) we define the circulant approximation $c(A)$ of the generator matrix A as follows:

$$\begin{bmatrix} c(\Lambda)+\mu_1 I_n & c(\Sigma) & & \\ -\mu_1 I_n & c(\Lambda)+c(D)+\mu_1 I_n & c(\Sigma) & \\ & \ddots & \ddots & \\ & & -\mu_1 I_n & c(\Lambda)+c(D) \end{bmatrix}. \quad (8)$$

From (6) and Davis [7], we have the following lemmas.

Lemma 2 $\mathrm{Rank}(c(\Lambda)-\Lambda) = \mathrm{Rank}(c(\Sigma)-\Sigma) = \mathrm{Rank}(c(D)-D) = 1$.

Lemma 3 *The matrices $c(\Lambda)$ and $c(\Sigma)$ can be diagonalized by the discrete Fourier transform F_n. The eigenvalues of $c(\Lambda)$ and $c(\Sigma)$ are given by*

$$F_n^* c(\Lambda) F_n = \mathrm{Diag}(\nu_1, \nu_2, \cdots, \nu_n)^t$$

and

$$F_n^* c(\Sigma) F_n = \mathrm{Diag}(\xi_1, \xi_2, \cdots, \xi_n)^t,$$

where

$$\begin{cases} \nu_j = \lambda(1-e^{\frac{2\pi(j-1)}{n}i}), & j=1,2,\cdots,n, \\ \xi_j = -\mu_2 e^{\frac{2\pi(j-1)}{n}i}, & j=1,2,\cdots,n. \end{cases}$$

Moreover, there exists a permutation P such that

$$P^t \cdot (I_{b_1+1} \otimes F_n^*) \cdot c(A) \cdot (I_{b_1+1} \otimes F_n) \cdot P = \mathrm{Diag}(C_1, C_2, \cdots, C_n),$$

where $C_i =$

$$\begin{bmatrix} \mu_1+\nu_i & \xi_i & & 0 \\ -\mu_1 & \mu_1+\mu_2+\nu_i & \xi_i & \\ & \ddots & \ddots & \ddots \\ 0 & & -\mu_1 & \mu_2+\nu_i \end{bmatrix}. \quad (9)$$

We note that all C_i except C_1 are strictly diagonal dominant and therefore they are non-singular. By similar argument in the proof of Lemma 1,

$$\tilde{C}_1 = (C_1 + \mathbf{ff}^t)$$

is non-singular. We define the preconditioner C as

$$(I_{b_1+1} \otimes F_n) \cdot P \cdot \mathrm{Diag}(\tilde{C}_1, C_2, \cdots, C_n) \cdot P^t \cdot (I_{b_1+1} \otimes F_n^*). \quad (10)$$

4 Convergence Analysis

In this section, we study the convergence rate of the PCG method when $n = m+h+1$ is large. In practice, the number of possible inventory states n is much larger than b_1 in the manufacturing systems and can easily go up to thousands. We prove that if all parameters μ_1, μ_2, λ and b_1 are fixed independent of n, then the preconditioned system GC^{-1} has singular values clustered around 1 as n tends to infinity. Hence when CG type methods are applied to solving the preconditioned system (5), we expect fast convergence. Numerical examples are given in §5 to verify our claim. We begin the proof by the following lemma.

Lemma 4 *We have* $\mathrm{rank}(G-C) \le 2(b_1+2)$.

Proof: From (4), we have $\mathrm{rank}(G-A)=1$, and from Lemma 2, we see that $\mathrm{rank}(A-c(A)) = 2(b_1+1)$. Using (9) and (10), we have $c(A)$ and C differ by a rank one matrix. Therefore, we have

$$\begin{aligned} \mathrm{rank}(G-C) &\le \mathrm{rank}(G-A) + \mathrm{rank}(A-c(A)) \\ &\quad + \mathrm{rank}(c(A)-C) \\ &\le 2(b_1+2). \end{aligned}$$

□

Proposition 1 *The preconditioned matrix GC^{-1} has at most $4(b_1+2)$ singular values not equal to 1. Hence GC^{-1} has singular values clustered around 1 when n tends to infinity.*

Proof: We first note that

$$GC^{-1} = I + (G-C)C^{-1} \equiv I + L_1,$$

where $\mathrm{rank}(L_1) \le 2(b_1+2)$ by Lemma 4. Therefore

$$C^{-*}G^*GC^{-1} - I = L_1^*(I+L_1) + L_1,$$

is a matrix of rank at most $4(b_1+2)$. Thus the number of singular values of GC^{-1} that are different from 1 is a constant independent of n. Hence the preconditioned matrix $C^{-1}G$ has singular values clustered around one by the Cauchy interlace theorem, see Wilkinson [14, p.103]. □

5 Cost Analysis and Numerical Examples

In this section, we give the computational cost of our PCG method and numerical examples are given to demonstrate its fast convergence rate. From (8) and (9), the construction of our preconditioner C need no cost. The main computational cost of our method comes from the matrix-vector multiplication of the form $G\mathbf{x}$, and solving the preconditioner system $C\mathbf{y} = \mathbf{r}$. By making use of the band structure of G, the matrix-vector multiplication $G\mathbf{x}$ can be done in $O((b_1+1)n)$ operations. The solution for $C\mathbf{y}=\mathbf{r}$ can be written as follows (c.f. 10):

$$\begin{aligned}\mathbf{y} &= (I_{b_1+1}\otimes F_n)\cdot P\cdot \\ &\quad \mathrm{Diag}(\tilde{C}_1^{-1}, C_2^{-1}, \cdots, C_n^{-1})\cdot \\ &\quad P^t\cdot(I_{b_1+1}\otimes F_n^*)\mathbf{r}.\end{aligned}$$

The matrix-vector multiplication of the forms $F_n\mathbf{x}$ and $F_n^*\mathbf{x}$ can be done in $O(n\log n)$ operations. The solution of the linear system $\mathrm{Diag}(\tilde{C}_1^{-1}, \cdots, C_n^{-1})\mathbf{y}=\mathbf{b}$ can be obtained in $O((b_1+1)n)$ operations. Hence the cost for solving (11) is

$$O((b_1+1)n\log n+(b_1+1)n).$$

We conclude that in each iteration of the PCG method, we need $O((b_1+1)n\log n)$ operations. The cost per iteration of the Block Gauss-Seidel (BGS) method is $O((b_1+1)n)$. This can be done by making use of the band structure of the diagonal blocks of the generator matrix A. Although the PCG method requires an extra $O(\log n)$ operations in each iteration, the fast convergence rate (roughly constant independent of n) of our method can more than compensate for this minor overhead (see the numerical examples below). In Proposition 1, we have proved the preconditioned linear system (5) has singular values clustered around one, so we expect the PCG method converges very fast. The convergence rate of BGS is roughly linear in n. Thus the total cost for the PCG method and the BGS method are

$$O((b_1+1)n\log n) \quad \text{and} \quad O((b_1+1)n^2)$$

respectively. Both PCG and BGS require $O((b_1+1)n)$ memory. Clearly at least $O((b_1+1)n)$ memory is required to store the approximated solution in each iteration.

We use a generalized conjugate gradient method, namely the Conjugate Gradient Squared (CGS) (See [12]), to solve the preconditioned system (5). The method does not require the transpose of the iteration matrix $G^{-1}C$. We compare our PCG method with the BGS method in the following numerical examples. In the examples, we let

$$\lambda=1, \mu_1=3/2 \text{ and } \mu_2=3.$$

The stopping criteria for both PCGS and BGS is

$$||G\mathbf{x}_k-\mathbf{f}||_2<10^{-10},$$

where $\mathbf{x}_k$ is the approximated solution obtained at the k-th iteration. The initial guess for both methods is the unit vector $\mathbf{f}=(0,\cdots,0,1)^t$. All the computations are done in a HP 712/80 workstation with MATLAB. We give the number of iterations for convergence of PCGS and BGS (Tables 1) for different values of b_1. The symbols I, C, BGS represent the methods used, namely, CGS without preconditioner, CGS with preconditioner C and the Block Gauss-Seidel method. The symbol ** signifies the number of iterations is greater than 200.

	$b_1=1$			$b_1=2$		
n	I	C	BGS	I	C	BGS
16	34	5	72	46	8	71
64	129	7	142	130	8	142
256	**	8	**	**	8	**
1024	**	8	**	**	8	**

	$b_1=4$			$b_1=8$		
n	I	C	BGS	I	C	BGS
16	54	11	71	64	19	72
64	130	11	142	139	19	142
256	196	11	**	**	19	**
1024	**	11	**	**	19	**

Table 1 : Number of Iterations for PCG and BGS

6 Concluding Remarks

We consider a two-stage manufacturing system. The inventory levels are modeled as an Markovian process. The generator matrix A for the inventory system is derived. Preconditioned conjugate gradient method is presented to solve the steady state probability distribution of the process. Preconditioner is constructed by taking circulant approximation of the blocks of A. We prove that the preconditioned matrix has singular values clustered around one. Numerical experiments are reported to illustrate the fast convergence rate of our method.

We consider the case when the manufacturing system has multiple identical machines in each stage (r_1 machines in stage one and r_2 machines in stage two). In this case, the generator matrix $\hat{A}$ will be given as follows: (see Siha [11]): $\hat{A}=$

$$\begin{bmatrix} \Lambda+\Gamma & \Sigma_1 & & & 0 \\ -\Gamma & \Lambda+D_1+\Gamma & \Sigma_2 & & \\ & \ddots & \ddots & & \\ & & -\Gamma & \Lambda+D_{b_1-1}+\Gamma & \Sigma_{b1} \\ 0 & & & -\Gamma & \Lambda+D_{b_1} \end{bmatrix}, \quad (11)$$

where

$$\Lambda=\begin{bmatrix} 0 & -\lambda & & 0 \\ & \lambda & \ddots & \\ & & \ddots & -\lambda \\ 0 & & & \lambda \end{bmatrix}, \qquad (12)$$

$$
\Sigma_i = \begin{bmatrix} 0 & & & 0 \\ -\min(i, r_2)\mu_2 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -\min(i, r_2)\mu_2 & 0 \end{bmatrix}, \quad (13)
$$

D and Γ are the $n \times n$ diagonal matrices

$$D_i = \min(i, r_2)\mathrm{Diag}(\mu_2, \mu_2, \cdots, \mu_2, 0),$$

$$\Gamma = \mathrm{Diag}(\mu_1, 2\mu_1, \cdots, r_1\mu_2, \cdots, r_1\mu_2)$$

respectively. The circulant approximation techniques in §3 can be applied to the construction of preconditioner $\hat{C}$ for the generator $\hat{A}$. The circulant approximation of Λ, Σ_i, D_i and Γ_i are then given as follow:

$$
c(\Lambda) = \begin{bmatrix} \lambda & -\lambda & & 0 \\ & \lambda & \ddots & \\ & & \ddots & -\lambda \\ -\lambda & & & \lambda \end{bmatrix}, \quad (14)
$$

$$
c(\Sigma_i) = \begin{bmatrix} 0 & & & -r_2\mu_2 \\ -r_2\mu_2 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -r_2\mu_2 & 0 \end{bmatrix}, \quad (15)
$$

$$c(D) = \mathrm{Diag}(r_2\mu_2, \cdots, r_2\mu_2)$$

and

$$c(\Gamma) = \mathrm{Diag}(r_1\mu_1, \cdots, r_1\mu_2).$$

Following similar proof in §4, we can also prove that the the following proposition.

Proposition 2 *The preconditioned matrix $\hat{C}^{-1}\hat{A}$ has singular values clustered around one as n tends to infinity.*

Thus the PCG method will converge very fast when apply to solving the preconditioned matrix $\hat{C}^{-1}\hat{A}$ system.

It is interesting to generalize our method for the following cases: the machines are unreliable and the manufacturing system has more than two stages.

References

[1] R. Akella and P. Kumar, *Optimal Control of Production Rate in a Failure Prone Manufacturing Systems*, IEEE Trans. Automat. Contr., 31, (1986) pp. 116–126.

[2] R. Chan and W. Ching, *Toeplitz-circulant Preconditioners for Toeplitz Matrix and Its Application in Queueing Networks with Batch Arrivals*, SIAM J. Sci. Comput. 17, (1996) pp. 162–172.

[3] W. Ching, R. Chan and X. Zhou, *Circulant Preconditioners for Markov Modulated Poisson Processes and Their Applications to Manufacturing Systems*, SIAM J. Matrix Anal. Appl. 18, (1997) pp. 464–481.

[4] W. Ching and X. Zhou, *Hedging Point Production Planning for Failure Prone Manufacturing Systems*, Proceedings of the 5th Conference of the Operational Research Society of Hong Kong (1995), pp. 163–172.

[5] ———, *Matrix Method for Production Planning in Failure Prone Manufacturing Systems*, Lecture Notes in Control and Information Sciences (214) Springer-Verlag, London (1996), pp. 3–30.

[6] W. Ching, *Circulant Preconditioners for Failure Prone Manufacturing Systems*, To Appear in Linear Algebra and Its Applications (1997).

[7] P. Davis, *Circulant Matrices*, John Wiley and Sons, New York (1979).

[8] G. Golub and Van Loan, *Matrix Computations*, Johns Hopkins University Press (1989).

[9] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Sydney (1991).

[10] Y. Monden, *Toyota Production System*, Industrial Eng. Manufacturing Press, Atlanta, GA, (1983).

[11] S. Siha, *Modeling the Blocking Phenomenon in JIT Environment: An Alternative Scenario*, Computers and Engng., 30, (1996) pp. 61–75.

[12] P. Sonneveld, *A Fast Lanczos-type Solver for Non-symmetric Linear Systems,* SIAM J. Sci. Comput., 10, (1989) pp. 36–52.

[13] R. Varga, *Matrix Iterative Analysis*, Prentice-Hall, New Jersey (1963).

[14] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford (1965).

[15] G. Yamazaki, T. Kawashima and H. Sakasegawa, *Reversibility of Tandem Blocking Queueing Systems,* Mgmt. Sci., 31, (1985) pp. 78–83.