

자동 생산라인 모형에서의  
Transition Probability Matrix에 관한 연구  
(A Study on the Transition Probability Matrix  
set from a Transfer Line Model)

노 형 민 \*

Abstract

In this study, two stage transfer line with limited repair capability is modeled to formulate optimal dynamic repair priority policy. The method of Markov Chains is used to analyze the analytical model of this line. An efficient algorithm is developed, utilizing the block tridiagonal structure of the transition probability matrix, to obtain the steady state probabilities and system performance measures, such as the steady state production rate of the line and the average in-process inventory in the interstage buffer.

1. 서 론

생산라인 중에 있는 한 기계의 고장은 부품의 흐름을 중단시키게 되어 그 다음 공정에 있는 고장나지 않은 기계까지도 일을 할 수 없게 한다. 이에 대한 대책으로 기계와 기계 사이에 재공품을 갖는 버퍼(Buffer)를 설치하여 왔다. 최근에는 산업용 로봇과 자동화 시스템을 이러한 생산라인에 설치함으로써, 자동 생산라인에서는 작업자의 수가 감소하여 한 작업자가 여러 대의 기계를 관리하게 되었다. 이러한 자동 생산라인에서 기계적인 고장

이나 작업자의 잘못으로 인하여 한 대 이상의 기계나 로봇이 고장나면, 이 작업자는 생산라인의 최적 가동을 위해 보수 순서를 정할 필요가 있게 된다.

본 연구의 자동 생산라인에서는 그림 1에 보이는 것처럼 두 개의 작업장, 그 사이에 버퍼, 그리고 한 작업자가 있는 것으로 가정한다.

이 라인의 사이클타임, 버퍼 및 재공품, 기계와 로봇의 고장 및 수리에 대해서 아래와 같은 성질을 갖는다고 가정한다.

(a) 각 작업장에서 부품을 가공하기 위해 기계와

\* 한국과학기술원 CAD/CAM실

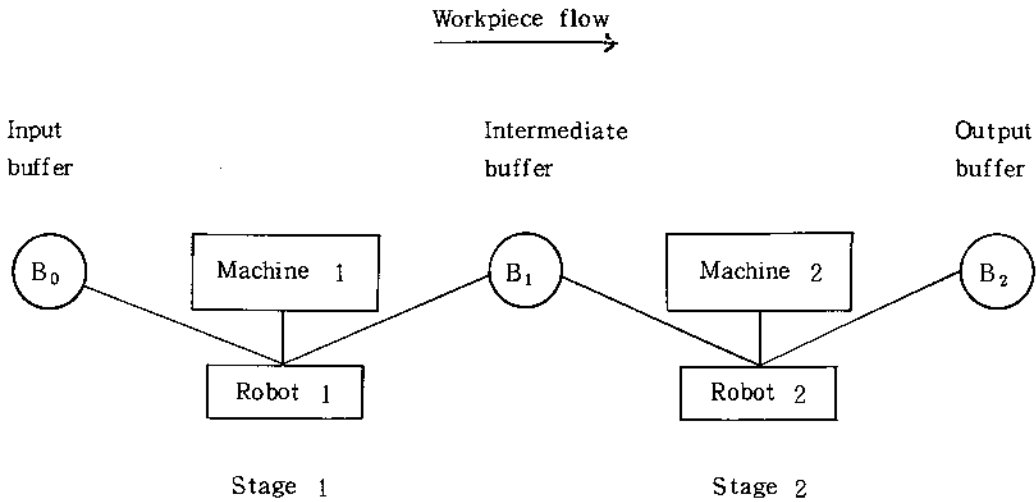


Figure 1. Two-stage transfer line with robots

로봇이 소요하는 시간(사이클 타임)은 일정하다.

(b) 이 라인에서 가공될 부품은 항상 input 버퍼에 있고 가공이 끝난 부품을 옮겨 놓을 장소도 항상 output 버퍼에 있다.

(c) 한 사이클타임 동안에 기계나 로봇이 고장날 확률은 일정하다. 단 부품이 없거나 버퍼 장소의 부족 그리고 관련기계나 로봇의 고장으로 가동하지 못한 로봇이나 기계는 고장나지 않는다.

(d) 한 사이클타임 동안에 고장난 기계나 로봇을 보수 완료할 확률은 일정하다.

(e) 기계나 로봇의 상태는 매 주기 로봇이 가공물을 기계에 놓은 후에 정해진다.

(f) 로봇의 고장은 부품을 기계에 놓을 때에만 고려된다.

가정 (c)와 (d)로부터 기계와 로봇의 상태는 가동상태와 고장상태로 나뉘어지며 이 두 상태의 관계는 Markov Chain을 이용하여  $2 \times 2$  변환확률(transition probability) 행렬식으로 나타낼 수 있다. 두 대의 기계와 두 대의 로봇의 상태를 각각  $M_1$ ,  $M_2$  및  $L_1$ ,  $L_2$ 로 표현하자. 버퍼에 있는 재공품 수량을  $X$ 로, 그리고 두 대 이상 고장났을 때 보수를 먼저 받는 작업장을  $R$ 로 표현하면,  $t$ 번째 사이클 타임에 이 자동 생산라인 시스템의 상태는 아래와 같은 시스템 요소 상태의 함수로 나타낼 수 있다.

$$S(t) = S(L_1, M_1, L_2, M_2, X, R) \quad (1)$$

여기에서

$$M_1, M_2, L_1, L_2 = 0 \text{ or } 1 \text{ (고장상태 or 가동상태)}$$

$$0 \leq X \leq N \quad (N = \text{버퍼 크기})$$

$$R = 1 \text{ or } 2$$

식(1)에서  $M_1, M_2, L_1, L_2$ 는 두 개의 가능한 상태를 갖으므로 주어진  $X$ 와  $R$  조건에서  $2^4 = 16$ 개의 다른 시스템 상태를 갖는다. 그리고 두 대 이상이 동시에 고장나서 보수 순서를 정해야 하는 경우는  ${}^4C_2 + {}^4C_3 + {}^4C_4 = 11$ 이다. 그러나 같은 작업장의 기계와 로봇이 모두 고장나는 경우에는, 모두 고쳐지기 전에는 보수 순서에 관계없이 부품이 가공되지 못하므로 이러한 두 경우는 제외한다. 따라서  $R$ 에 의해 고려되는 상태는 9이다. 이리하여 이 자동 생산라인 모델의 총 시스템 상태 수  $n$ 은

$$\begin{aligned} n &= (2^4 + 9)(N + 1) \\ &= 25(N + 1) \quad \dots \dots \dots (2) \end{aligned}$$

이다. 식(2)에서 버퍼 크기가 증가할 수록  $n$ 은 증가하여  $N = 30$ 일때 그 값은 775가 된다.  $n$ 값이 증가할 수록 이 시스템 모델은 그 변환 확률 행렬

식 크기의 증가로 해를 얻기가 어려워진다. [1,2]

본 연구의 목적은 앞에 제시된 모델의 수치 해를 얻기 위한 알고리즘 개발에 있다. 시스템 전체의 변환 확률 행렬식의 역을 구하기 위해서 Block tridiagonal 행렬식 구조와 Crout reduction 및 L-U decomposition 방법을 이용하였다.

## 2. 시스템 모델 형성

확률변수  $P_{ij}$ 를 한 사이클타임 동안에  $i$  시스템 상태에서  $j$  시스템 상태로 변화하는 변환 확률이라 하자.  $n$ 개의 총 시스템 상태를 갖는 irreducible ergodic Markov Chain에서,  $j$  시스템 상태의 정상상태 확률  $S_j$ 는 다음 두 식의 해가 된다.

$$S_j = \sum_{i=1}^n P_{ij} S_i \quad 1 \leq j \leq n \quad (3)$$

$$\sum_{j=1}^n S_j = 1 \quad (4)$$

$n$ 개의 정상상태 확률  $S_j$ 를 포함하는 벡터를  $\underline{S}$ 라고 하면 식(3)은

$$\underline{S} = \underline{T}\underline{S} \\ (\underline{T} - \underline{I})\underline{S} = 0 \quad (5)$$

으로 표현할 수 있는데 여기서  $\underline{T}$ 는 확률변수  $P_{ij}$ 로 이루어지는  $n \times n$  시스템 전체의 변환 확률 행렬식이고  $\underline{I}$ 는  $n \times n$  identity 행렬식이다.

$\underline{T}$  행렬식은 그림 2에서 보이는 바와 같이 횡렬을  $(t-1)$  사이클 타임에서의 시스템 상태로, 종렬을  $t$  사이클 타임에서의 시스템상태로 하여 재공품이 증가되는 순서로, 즉 0, 1, 2, ...,  $N$ 로, 배열하면  $(N+1) \times (N+1)$  block 크기를 갖는다. 식(2)로부터  $(N+1)$ 개의 재공품 각각에 대하여 25개의 시스템 상태가 횡렬과 종렬에 반복되므로 각  $P_i$  block은  $25 \times 25$  크기를 갖는다. 이때, 한 사이클타임 후에 재공품 수의 변화는 하나가 최대이므로 둘 이상의 증가 혹은 감소되는 상태로 변할 확률은 없다. 이러한 재공품 수의 변화 특성을 0에서  $N$ 까지 적용하면  $\underline{T}$  행렬식은 main, upper, lower의 diagonal-block 들만 0이 아닌 확률을

$X_{t-1}$	0	1	2	3	...	$N-1$	$N$	
$X_t$	25	25	25			25	25	
0	25 {	$\overline{P_1}$	$\overline{P_3}$	$\overline{0}$	0	...	$\overline{0}$	$\overline{0}$
1	25 {	$P_2$	$P_4$	$P_3$	0	...	0	0
2	25 {	0	$P_5$	$P_4$	$P_3$	...	0	0
.	.	.	.	.	...	.	.	.
$N-1$	25 {	0	0	0	0	...	$P_4$	$P_6$
$N$	25 {	0	0	0	0	...	$P_5$	$P_7$

Figure 2. Transition probability matrix  $\underline{T}$  from cycle  $(t-1)$  to  $t$  with  $P_i$  blocks

( $X_t$ : buffer level at cycle  $t$ )

갖는 block tridiagonal 구조를 갖는 행렬식이 된다.

식(3)과 식(4)의 연립방정식에서  $n$ 개의 변수에  $(n+1)$ 개의 식이 있으므로 한 식은 필요가 없다. 식(4)를 행렬식의 형태로 표현하면,

$$\underline{U}'\underline{S} = \underline{1}, \quad \underline{U} = (1, 1, \dots, 1)' \quad (6)$$

이다. 식(5)의  $(\underline{T} - \underline{I})$  행렬식에서 필요없는 식을  $i$ 번째 횡렬이라 하여 식(6)을 식(5)에 대체시킨 결과는 다음과 같이 된다.

$$\underline{T}_1 \underline{S} = \underline{b}, \quad \underline{b} = (0, \dots, 0, 1, 0, \dots, 0)' \quad (7)$$

따라서 정상상태 확률 벡터  $\underline{S}$ 는

$$\underline{S} = \underline{T}_1^{-1} \underline{b} \quad (8)$$

로 되므로  $\underline{T}_1$  행렬식의 역을 구해야 한다. 그런데 벡터  $\underline{U}$ 를  $\underline{T}$  행렬식의  $i$ 번째 횡렬에 대체시킴으로써  $\underline{T}_1$  행렬식은 block tridiagonal 구조를 갖지 못하므로,  $\underline{T}_1$  행렬식이 커질수록 그 역을 구하기 어려워진다. 왜냐하면  $\underline{T}_1$  행렬식 내의 상수들의 절대값이 1보다 작아서 truncation error가 증가

되어 그 해의 정확도가 감소되고, 컴퓨터의 기억용량을 많이 필요로 하기 때문이다. 따라서 해의 정확도를 증가시키고 컴퓨터 기억용량을 적게 쓰면서  $T_1$  행렬식의 역을 구할 수 있는 알고리즘이 개발되어야 한다.

### 3. 알고리즘 개발

#### ① Block tridiagonal 행렬식

Block tridiagonal 행렬식은 일반 행렬식의 역을 구하는데 따르는 문제점을 어느 정도 피할 수 있으므로  $T_1$  행렬식을 그러한 구조를 갖도록 변환하였다. 이 변환은  $T_1$  행렬식의 diagonal 과 bidagonal block 외에 있는  $i$  번째 횡렬의 각 상수로부터 1을 빼서 0으로 만듦으로써 이루어진다. 그 결과 행렬식을  $T'$ 라 하면  $T_1$  과  $T'$  의 관계는 다음과 같이 된다.

$$T' = T_1 - \underline{b} \underline{a}^t, \quad \underline{a} = (1, \dots, 0, \dots, 0, 1, \dots)^t$$

..... (9)

즉,

$$T_1^{-1} = (T' + \underline{b} \underline{a}^t)^{-1} \dots \dots \dots (10)$$

Householder lemma (3)를 이용하여 식(10)에 적용하면 (4), 식(8)은

$$\begin{aligned} \underline{S} &= T_1^{-1} \underline{b} \\ &= T'^{-1} \underline{b} - T'^{-1} \underline{b} (1 + \underline{a}^t T'^{-1} \underline{b})^{-1} \underline{a}^t T'^{-1} \underline{b} \\ &= (1 / (1 + \underline{a}^t T'^{-1} \underline{b})) T'^{-1} \underline{b} \dots \dots \dots (11) \end{aligned}$$

이 되는데  $(1 + \underline{a}^t T'^{-1} \underline{b})$  항은 스칼라 양이므로 양변에 이 값을 곱해 다음과 같이 나타낼 수 있다.

$$\underline{S}' = T'^{-1} \underline{b} \dots \dots \dots (12)$$

여기에서

$$\underline{S} = (1 + \underline{a}^t T'^{-1} \underline{b}) \underline{S}' \dots \dots \dots (13)$$

정상상태 벡터  $\underline{S}$ 는 식(12)에 의해 얻어지는  $\underline{S}'$  벡터에 스칼라 양  $(1 + \underline{a}^t T'^{-1} \underline{b})$ 를 곱한 값이 된다. 따라서 block tridiagonal 구조를 갖는  $T'$  행

렬식의 역을 구하여  $\underline{S}'$  벡터를 구하여야 한다.

#### ② L-U decomposition 방법

식(12)의 block tridiagonal 행렬식  $T'$ 과  $\underline{b}$ 는 아래와 같이 표현되는데  $T'$  행렬식은 L-U decomposition 방법으로 그 역을 구하였다.

$$T' = \begin{bmatrix} B_0 & -C_0 & 0 & 0 & & & & \\ -A_1 & B_1 & -C_1 & 0 & & & & \\ 0 & -A_2 & B_2 & -C_2 & & & & \\ 0 & 0 & 0 & 0 & \cdot & & & \\ & & & & & & & \\ & & 0 & & & & & \\ & & & & & & & \\ & & & & & & -C_{N-1} & \\ & & & & & -A_N & B_N & \end{bmatrix}$$

$$\underline{b} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_N \end{bmatrix} \dots \dots \dots (14)$$

여기서  $A_j, B_j, C_j$  행렬식은  $25 \times 25$  크기이고  $b_j$ 는  $25 \times 1$  크기이다. ( $j=0, \dots, N$ ) Boundary block인  $B_0, A_1, C_{N-1}, B_N$ 은 각각 그림 1의  $T_1$  행렬식의  $P_1, P_2, P_6, P_7$  으로부터 그리고 inner block인  $C_{j-1}, B_j, A_{j+1}$  은 각각  $P_3, P_4, P_5$  로부터 식(9)의 변형을 통해 얻어진다. Gaussian Elimination 방법의 변형인 Crout Reduction 방법(5)을 이용하여  $T'$  행렬식을 lower 와 upper diagonal 행렬식인 L 과 U로 나누었다.

$$T' = L U$$

$$L = \begin{bmatrix} B_0 & 0 & 0 & & \\ -A_1 & (B_1 - A_1 E_0) & 0 & & 0 \\ 0 & -A_2 & (B_2 - A_2 E_1) & & \\ & 0 & & \cdot & \cdot & 0 \\ & & 0 & -A_N & (B_N - A_N E_{N-1}) \end{bmatrix}$$

$$= \begin{bmatrix} X_0 & 0 & 0 & 0 & \\ -A_1 & X_1 & 0 & 0 & \\ 0 & -A_2 & X_2 & 0 & \\ 0 & 0 & \cdot & \cdot & 0 \\ & & 0 & -A_N & X_N \end{bmatrix}$$

$$U = \begin{bmatrix} I & -B_0^{-1} C_0 & & 0 & & 0 \\ 0 & I & -(B_1 - A_1 B_0^{-1} C_0) C_1 & & & 0 \\ 0 & 0 & I & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & \cdot \end{bmatrix}$$

$$= \begin{bmatrix} I & -E_0 & 0 & 0 & & \\ 0 & I & -E_1 & 0 & & \\ 0 & 0 & I & \cdot & & \\ 0 & 0 & 0 & \cdot & -E_{N-1} & \\ & & & 0 & & \end{bmatrix}$$

여기서  $E_j, X_j$  행렬식은 다음과 같이  $A_j, B_j, C_j$  행렬식의 함수이다.

$$\begin{aligned} X_0 &= B_0 \\ E_0 &= B_0^{-1} C_1 \\ X_j &= B_j - A_j E_{j-1} \quad (j = 1, \dots, N) \\ E_j &= X_j^{-1} C_j \end{aligned} \quad (15)$$

식(15)에서  $E_j$  행렬식의 각 종렬( $e_j, j = 1, \dots, 25$ )은 이에 대응하는  $C_j$  행렬식의 각 종렬( $c_j$ )과 다

음의 관계로 표현되므로, 25개의 종렬  $e_i$ 를 위해 오직 한 번의  $X_j$  역만 구하면 된다.

$$e_i = X_j^{-1} c_i \quad i = 1, \dots, 25 \quad (16)$$

이렇게 하여  $T'$ 의 L-U decomposition이 끝난다. 식(12)의 양변에  $T' (= LU)$ 를 곱하면 식(12)는

$$(LU) \underline{S}' = \underline{b} \quad (17)$$

이 되어 식(17)로부터 해  $\underline{S}'$ 을 아래의 두 단계를 통

해 얻는다. (4)

(a) 식(17)의 양변에  $L^{-1}$ 를 곱한다.(forward elimination)

(b) (a)의 결과 식 양변에  $U^{-1}$ 를 곱한다.(backw-

ard substitution)

식(17)의 해가 얻어지면  $T_1$  행렬식의 해는 식(13)을 통해 쉽게 얻어진다. 이 알고리즘의 흐름을 그림 3에 요약하였다.

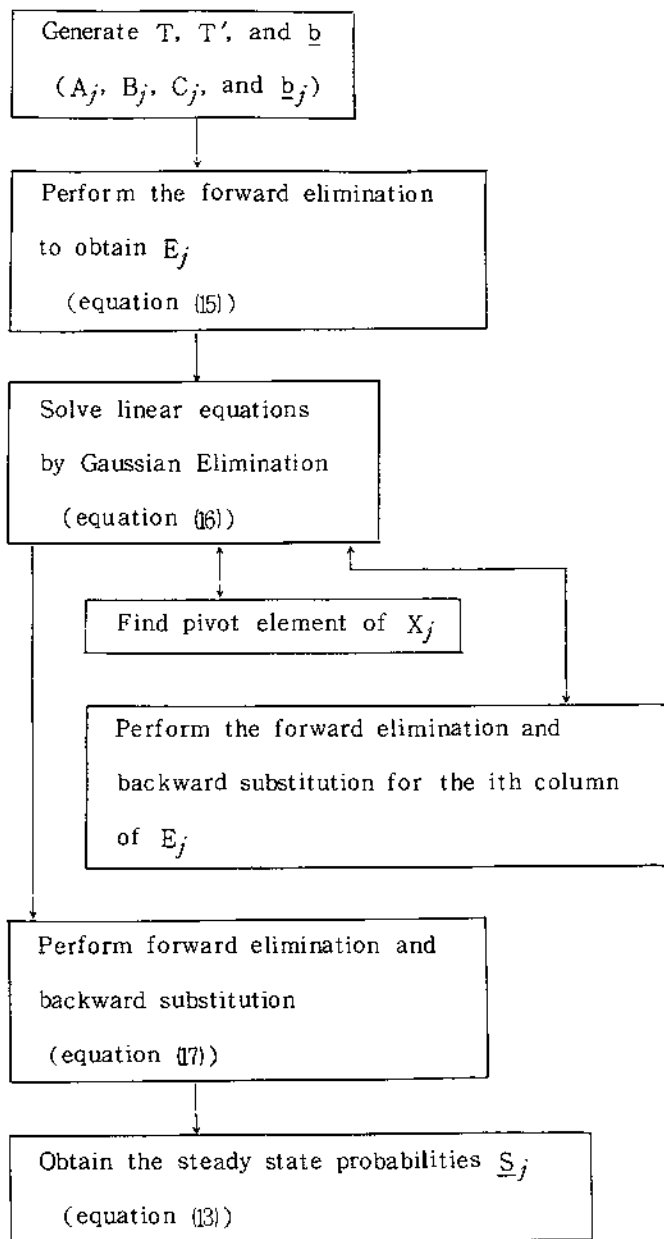


Figure 3. Flow diagram of the algorithm to obtain the steady state probabilities of  $T_1$  matrix

#### 4. 시스템 성능에 대한 수치 결과

자동 생산라인 모델의 생산량, 재공품 수량, 기계 이용도와 같은 시스템 성능에 대한 수치 결과는 앞에서 얻어진 정상상태 벡터  $\underline{S}$ 로부터 얻을 수 있다. 한 개의 부품을 생산하기 위해서는 버퍼에 재공품이 한 개 이상 있어야 하고 두번째 작업장의 로봇과 기계가 가동상태에 있어야 한다. 이러한 조건을 만족하는 시스템의 정상상태 확률들의 합이 한 사이클타임 동안에 예상되는 생산량  $E(N_p)$ 이다.

$$E(N_p) = \sum_{L_1=0}^1 \sum_{M_1=0}^1 \sum_{X=1}^N \sum_{R=1}^2 S(L_1, M_1, 1, 1, X, R)$$

예상 재공품 수량  $E(N_f)$ 는 재공품 수  $X$ 와 그 때의 정상상태 확률들의 곱들의 합으로 계산된다.

$$E(N_f) = \sum_{L_1=0}^1 \sum_{M_1=0}^1 \sum_{L_2=0}^1 \sum_{M_2=0}^1 \sum_{X=1}^N \sum_{R=1}^2 XS(L_1, M_1, L_2, M_2, X, R)$$

한 사이클타임 동안에 예상되는 기계 이용도는 1에서 기계가 가동되지 않는 시간을 뺀 값이다. 기계는 자기 자신의 고장으로 인해 가동되지 않을 뿐만 아니라 자신은 가동상태에 있음에도 불구하고 같은 작업장에 있는 관련 기계가 고장으로 혹은 두 대 이상의 기계가 고장나서 보수할 작업자가 부족함으로 인해서도 가동되지 못할 수가 있다. 또한 버퍼에 재공품이  $N$ 개 있는 경우에는 첫째 작업장이, 재공품이 없는 경우에는 둘째 작업장이 기계의 상태에 관계없이 가동할 수 없다. 예를들면, 첫번째 작업장에 있는 로봇(L1)의 이용도는 다음과 같이 표현된다.

$$E(T_u) = 1 - \left[ \sum_{M_1=0}^1 \sum_{L_2=0}^1 \sum_{M_2=0}^1 \sum_{X=1}^N \sum_{R=1}^2 S(0, M_1, L_2, M_2, X, R) \right]$$

$$+ \sum_{L_2=0}^1 \sum_{M_2=0}^1 \sum_{X=1}^N \sum_{R=1}^2 S(1, 0, L_2, M_2, X, R) + \sum_{L_2=0}^1 \sum_{M_2=0}^1 \sum_{R=1}^2 S(1, 1, L_2, M_2, N, R)$$

두 대 이상의 로봇이나 기계가 고장났을 때 작업자는 버퍼에 있는 재공품의 수량에 따라 보수 순서를 정하도록 하였다. 즉, 보수 순서 결정을 내릴 때 버퍼에 있는 재공품 수가 미리 정해진 수보다 적으면 첫번째 작업장에서 우선 순위를 갖고, 같거나 그 이상이면 두번째 작업장에서 그 우선 순위를 갖도록 하였다. 이러한 가정에 따라 기준 재공품 수를 1에서  $N$ 까지 변화시키며 정상상태에서의 예상 생산량과 평균 재공품 수량 등을 앞에 설명된 알고리즘을 통해 얻을 수 있다.  $N=30$  이고 주어진 로봇과 기계의 변환 확률 조건하에서, 예상 생산량과 재공품 수량을 기준 재공품 수(Buffer occupancy)의 변화에 따라 구한 결과가 표 1과 같다. 최대 생산을 위해서는 보수 순서를 정하는 기준 재공품 수가 7이며 이 때의 생산량은 한 사이클타임 동안에 0.3127개가 된다. 즉, 재공품 수가 7 이상이면 둘째 작업장을 먼저 보수하는 것이 최대 생산을 얻는 보수 기준임을 알 수 있다. 그러나 평균 재공품 수량은 기준 재공품 수가 증가함에 따라 증가하므로 기준 재공품 수는 시스템 성능을 최적화하는 값으로 정해져야 한다.

#### 5. 결 론

가정된 시스템 모델의 정상상태 변환 확률 벡터  $\underline{S}'$ 를 얻기 위해서  $25(N+1) \times 25(N+1)$  크기의 행렬식  $T_1$ 의 역을 구하는 대신  $T'$  행렬식의 Block tridiagonal 구조와 L-U decomposition 방법을 이용하여 식(5)에 보이는  $25 \times 25$  크기의  $X_j$ ,  $j=1, \dots, N$ , 행렬식만 그 역을 구하였다. 즉, 재공품 버퍼 크기가 30 이라면 역을 구하여야 할 행렬식의 크기는  $775 \times 775$  대신  $25 \times 25$  이 된다.

Table 1. Numerical results of the system performance measures for the various buffer occupancies

Buffer occupancy	Production rate	In-process inventory
1	0.3023	11.73
2	0.3026	11.79
3	0.3079	13.81
4	0.3106	14.97
5	0.3119	15.70
6	0.3125	16.17
7	0.3127	16.49
8	0.3125	16.71
9	0.3122	16.85
10	0.3117	16.94
11	0.3111	17.01
12	0.3105	17.05
13	0.3098	17.07
14	0.3090	17.09
15	0.3083	17.11
16	0.3074	17.13
17	0.3066	17.15
18	0.3057	17.18
19	0.3048	17.22
20	0.3039	17.28
21	0.3029	17.35
22	0.3020	17.43
23	0.3010	17.54
24	0.3000	17.66
25	0.2991	17.80
26	0.2982	17.96
27	0.2973	18.14
28	0.2965	18.33
29	0.2959	18.55
30	0.2941	18.80

when

L1	M1	L2	M2
$\begin{vmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{vmatrix}$	$\begin{vmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{vmatrix}$	$\begin{vmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{vmatrix}$	$\begin{vmatrix} 0.99 & 0.01 \\ 0.05 & 0.95 \end{vmatrix}$



역을 구할 행렬식의 크기 감소에 따라, 시스템 모델의 정상상태 확률들을 정확히 구할 수 있고 또한 이러한 확률들의 합으로 표현되는 시스템 성능에 대한 수치값을 정확히 예측할 수 있다.

그리고 개발된 이 알고리즘으로 계산이 필요한

컴퓨터의 기억용량 크기가 감소되었다.  $25(N+1) \times 25(N+1)$  크기의  $T_1$  행렬식을 위한 기억용량 대신에 식(5)의  $A_j, B_j, C_j$  행렬식들의 해당하는  $3 \times 25 \times 25$  크기의 기억용량만 할당하여  $(N+1)$  번 반복 사용함으로써  $E_j$  행렬식을 구할 수 있다.

## References

1. Okamura, K. and H. Yamashina, "Analysis of the Effect of the Buffer Storage Capacity in Transfer Line System," *AIIE Trans.* Vol. 9, No. 2, (1977).
2. Gershwin, S.B. and I.C. Schick, "Modeling and Analysis of Three Stage Transfer Line with Unreliable Machines and Infinite Buffers," *Op. Res.*, Vol. 31, No. 2 (1983).
3. Householder, A.S., *The Theory of Matrices in Numerical Analysis*, Chap. 5, Dover Publ. Inc. (1964).
4. Rho, H.M., "Dynamic Repair Priority Policy in Transfer Lines with Limited Repair Capabilities," unpublished Ph. D. Dissertation, The Penn. State Univ. 1985
5. Prather, D.W., "Pivoting and Block Tridiagonal Systems," Notes for AE/ME 526, Applied Research Lab, The Penn. State Univ.