

## Micro - Computer를 이용한 진동 시스템 Simulation에 관한 연구

주 해 호\* · 이 규 만\*\*

Development of Micro - Computer  
Simulation Programs for the Various Vibratory Systems

Haeho Joo · Gvumahn Lee

### Abstract

This paper presents a micro-computer simulation package for the various vibratory systems. The Package consists of 10 programs which describe the dynamical characteristics of the vibratory system. The programs have been written in BASIC (Applesoft) language and programmed on the 6502 CPU with 48 KRAM. This simulation package is stored in 5 1/4 inch floppy disk. The user requires no simulation expertise on the part of designer. Through a process of disk operation, the user can easily understand how to use this package.

### 1. 서 론

Computer Simulation이 공학 시스템을 해석하고 설계하는데 상당한 시간의 절약과 비용의 절감, 계산의 신속성과 정확성등의 장점 때문에 이공계 분야에서 중요한 설계 도구로서 널리 이용되고 있다. 최근 첨단기술의 발전에 따라 Computer는 점점 대형에서 부터 소형화 되어가고 있다. Micro - Computer의 탄생은 일반인에게까지 Computer를 이용할 수 있는 기회를 부여한 커다란 의미를 갖고있다.

그러나 Hardware의 개발만으로는 완전한 Computer의 구실을 다 할 수 없으며 Software의 개발이 뒤따라야 한다. 지금까지 개발된 Software는 대형 Computer에 사용되

는 FORTRAN, COBOL, PASCAL, PL/1, ALGOL 등등 많이 개발되었다.<sup>(1)</sup> 그러나 이러한 언어로써 공학문제를 해석하기에는 Programming 상의 어려움이 많아서 고급 언어인 CSMP, MIMIC, CSSL 같은 Simulation 언어가 개발되었으나 이것들은 대형 Computer에서만 사용 가능하므로 대형 Computer를 이용할 수 없는 일반인에게는 소용이 없다.

그래서 누구나 손쉽게 접할 수 있는 Micro - Computer로써 공학 문제를 해석하는 Software 개발이 절실히 요구되고 있다. 이러한 문제를 해소하기 위한 한가지 방법으로 본 연구는 공학용 문제중에서 진동시스템의 해석을 위한 Simulation Software를 개발하였다.

이 Simulation Software는 현재 국내에서 가장 널리 보급된 6502 CPU에 적합한 BASIC

\* 영남대학교 공과대학 교수

\*\* 영남대학교 대학원

언어로 만들었으며 48K 기억용량으로 사용자가 능하도록 만들었다. Program의 종류는 10가지이고 일자유도 진동 시스템에서 다자유도 진동 시스템에 이르기까지 광범위하게 취급하였다.

이 논문에서는 대표적인 예로써 Myklestad-Prohl의 전환행렬(Transfer Matrix)을 이용하여 균일 외팔보의 고유진동수를 구하는 Program을 소개한다.

진동 시스템의 수학적 모형으로 부터 Programming 기법을 설명하고, 개발한 Pro-

gram의 종류와 Disk의 운용방법을 설명한다.

## 2. 진동 시스템의 모형<sup>(2)</sup>

그림 1은 N개의 등구간으로 이산화(discrete)된 균일 단면 외팔보의 개략도이다. 길이 L인 외팔보를 N등분하고 임의의 새그먼트(Segment) n에 대한 그림 2와 그림 3에 표시한다.

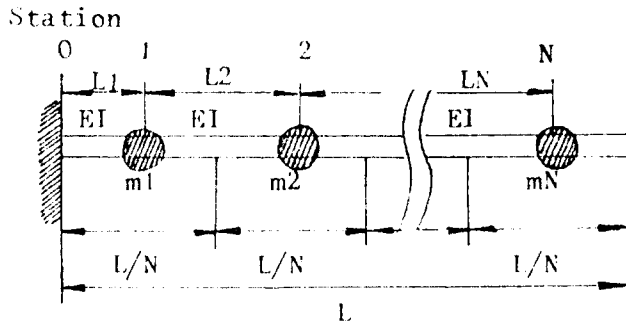


그림 1. N개의 등구간으로 근사화된 균일단면 외팔보

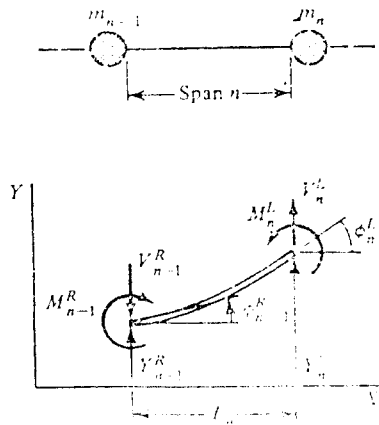


그림 2. 임의의 Span n에 대한 자유물체도

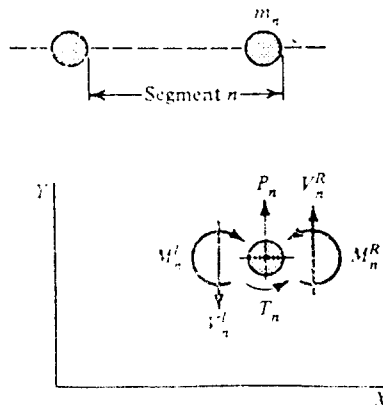


그림 3. 임의의 점질량 mn에 대한 자유물체도

여기서

- $\bar{N}$  : Station 수
- $L$  : 외팔보의 전체길이
- $EI$  : 보의 강성계수
- $M_n$  :  $n$  구간에서의 점질량
- $V_{n-1}^R, V_n^L$  :  $n-1$  Station 에서의 우단의 전단력,  $n$  Station 에서의 좌단의 전단력
- $M_{n-1}^R, M_n^L$  :  $n-1$  Station 에서의 우단의 모우멘트,  $n$  Station 에서의 좌단의 모우멘트
- $\phi_{n-1}^R, \phi_n^L$  :  $n-1$  Station 에서의 우단의 경사각,  $n$  Station 에서의 좌단의 경사각
- $Y_{n-1}^R, Y_n^L$  :  $n-1$  Station 에서의 우단의 처짐량,  $n$  Station 에서의 좌단의 처짐량
- $P_n$  : 점질량  $n$  에 작용하는 작용력
- $T_n$  : 점질량  $n$  에 작용하는 Torque

Myklestad-Prohl 의 일반 이론은 다음과 같이 전개된다.

$$\{Z\}_i^R = T_i \{Z\}_{i-1}^R \dots\dots\dots (1)$$

여기서

- $\{Z\}_i^R$  : 우단의 상태벡터
- $T_i$  : 전환행렬 (Transfer Matrix)
- $\{Z\}_{i-1}^R$  : 좌단의 상태벡터

그림 2 의 자유 물체도로 부터

$$V_n^L = V_{n-1}^R \dots\dots\dots (2)$$

$$M_n^L = M_{n-1}^R - L_n V_{n-1}^R \dots\dots\dots (3)$$

$$\phi_n^L = \phi_{n-1}^R + (L/EI)_n M_{n-1}^R - (L^2/2EI)_n V_{n-1}^R \dots\dots (4)$$

$$Y_n^L = Y_{n-1}^R + L_n \phi_{n-1}^R + (L^2/2EI)_n M_{n-1}^R - (L^3/6EI)_n V_{n-1}^R \dots\dots (5)$$

위식 (2),(3),(4),(5) 로 부터 필드 전환 과 같이 나타낸다.  
 행렬 (Field Transfer Matrix)  $F_n$  은 식(6)

$$\begin{pmatrix} Y \\ \phi \\ M \\ V \end{pmatrix}_n^L = F_n \begin{pmatrix} Y \\ \phi \\ M \\ V \end{pmatrix}_{n-1}^R \quad F_n = \begin{pmatrix} 1 & L & L^2/2EI & -L^3/6EI \\ 0 & 1 & L/EI & -L^2/2EI \\ 0 & 0 & 1 & -L \\ 0 & 0 & 0 & 1 \end{pmatrix}_n \dots (6)$$

그림 3의 자유 물체도로 부터

$$V_n^R = V_n^L - \omega^2 m_n Y_n^L \dots\dots\dots (7)$$

$$M_n^R = M_n^L - \omega^2 J_n \phi_n^L \dots\dots\dots (8)$$

$$\phi_n^R = \phi_n^L \dots\dots\dots (9)$$

$$Y_n^R = Y_n^L \dots\dots\dots (10)$$

여기서  $-\omega^2 m_n Y_n^L, -\omega^2 J_n \phi_n^L$ 은 D'Alembert의 관성 하중이며  $J_n$ 은  $m_n$ 의  $(x, y)$  평면에 수직인 축에 대한 질량 관성 모우먼트이다.

윗식 (7), (8), (9), (10)으로 부터 점전환행렬 (Point Transfer Matrix)  $P_n$ 은 식(11)과 같이 나타난다.

$$\begin{pmatrix} Y \\ \phi \\ M \\ V \end{pmatrix}_n^R = P_n \begin{pmatrix} Y \\ \phi \\ M \\ V \end{pmatrix}_n^L \quad P_n = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\omega^2 J & 1 & 0 \\ -\omega^2 m & 0 & 0 & 1 \end{pmatrix}_n \dots\dots (11)$$

여기서 전환행렬  $T_n$ 은 식 (6), (11)로부터 다음 식 (12)로 나타내어 진다.

$$T_n = P_n F_n$$

$$\{Z\}_n^R = T_n \{Z\}_{n-1}^R$$

$$T_n = \begin{pmatrix} 1 & L & L^2/2EI & -L^3/6EI \\ 0 & 1 & L/EI & -L^2/2EI \\ 0 & -\omega^2 J & 1 - \omega^2 J L/EI & -L + \omega^2 J L^2/2EI \\ -\omega^2 m & -\omega^2 mL & -\omega^2 m L^2/2EI & 1 + \omega^2 m L^3/6EI \end{pmatrix}_n \dots\dots\dots (12)$$

그러므로 식 (12)의 순환식을 적용하면 상태벡터  $\{Z\}_n^R$ 과 Station 0에서의 상태벡터

$\{Z\}_0^R$ 은 다음과 같이 된다.

$$\{Z\}_n^R = T_n [T_{n-1} \dots T_2 T_1] \{Z\}_0^R \dots\dots\dots (13)$$

식 (13)으로 부터 외팔보의 경계 조건인  $Y_0 = \phi_0 = 0, M_N = V_N = 0$ 을 식 (13)에 대입하

면 식 (14)를 얻으며 이식을 전개하면 고유진동수  $\omega$ 에 대한 방정식이 구하여 진다.

$$A(\omega) = \begin{vmatrix} T_{33} & T_{34} \\ T_{43} & T_{44} \end{vmatrix} = 0 \quad \dots\dots\dots (14)$$

### 3. Micro-Computer Simulation

3-1. 프로그램 CANTILEVE의 논리 흐름도  
균일 외팔보 진동시스템의 고유진동수를 구하는 프로그램 CANTILEVE를 작성하기 위한 논리 흐름도는 그림 4와 같다.

먼저 균일 외팔보의 형상을 화면에 그림으로 표시하고  $(EI) / ((\text{질량}/\text{길이})(\text{보의 단면적})(\text{전체 길이})) = 1$ 이 되도록 정규화(Normalize)하여 문제의 Data를 입력시킨다.

Station수 N에 대한 (N-1)개의 근사 고유진동수는 \$CAL Loop에 의하여 계산되

어진다. 진동수가 주어지면 식 (14)에 판별 식의  $\Delta(\omega)$ 의 부호를 검토한다. 진동수를 조금씩 증가시켜 가면서  $\Delta(\omega)$ 의 부호를 검토

한다. 즉, 진동수  $W1$ 을 가정할 때 다음 단계에서는 아래와 같다.

$$W = W1 + DW$$

$$DW(New) = 1.6 \times DW(Old) : \text{부호 변화가 없을때}$$

$$DW(New) = (W - W1) / 3.2 : \text{부호 변화가 있을때}$$

부호 변화가 없으면 증분  $DW$ 는 1.6배 만큼 증가한다. 부호 변화가 일어나서 나머지가 생기면  $DW$ 는 3.2배 만큼 감소시키고 부호 변화가 일어나기 전에 진동수  $W1$ 에서 다시 계산을 시작한다. 만약  $|DT| < 0.001$ 이 만족되면 부호 변화에 관계없이 근사 진동수  $FQ(J)$ 가 결정되어 진다.

진동수 근사값  $FQ(J)$ 는  $(DW - DL) \leq 0$  일때 얻어지며  $DL$ 은 입력 Data 로써 진동수 근사치를 정하기 위한 판정기준이다. 두번째 근사 고유진동수는 두번째 근을 구하기 위한 반복 계산에서  $DW$ 는 2배의 큰 값을 취하게 된다.

좀더 정확한 고유진동수를 얻기 위하여  $X(1)$ 과  $X(2)$ 를 계산된 진동수  $FQ(J)$ 의 두 값이라 하면 좀더 정확한 진동수  $FQ(J)$ 는 직선 보간법에 의하여 계산된다.

상태벡터는 위에서 얻어진 진동수에 대하여 고정단의 전단력이 1이 되도록 정규화하여 위와 동일한 방법으로 얻어진다.

### 3 - 2. 프로그램의 Coding

그림 4에 따라 BASIC(Apple Soft) 언어로 Coding 한 프로그램의 Listing은 부록 1에 수록하였다.

### 3 - 3. 프로그램의 입력 및 출력

프로그램을 실행하면 먼저 균일 외팔보의 형상을 화면에 그림으로 표시하고 입력 데이터를 요구한다. 요구하는 입력 데이터는 먼저 Station의 수, 최대 반복회수, 진동수의 증가량, 허용오차를 입력하게 되고 다음으로 각 Span의 길이 ( $L_n$ ), 각 Span의 강성계수 ( $EI$ ) 및 각질량 ( $M_n$ )의 값을 입력한다.

위의 입력 데이터가 확실한가를 확인한 다음 실행을 시키면 다음과 같은 출력이 나타난다.

다. 고유진동수의 근사값에 도달하는 실제 계산의 반복회수가 표시되고 (N-1)개의 고유진동수가 표시되며 각 고유진동수에 대응하는 변위 (Y), 기울기 (PI), 모우멘트 (M), 전단응력 (V)이 각 Station에 대하여 출력된다. 계산된 고유진동수중 처음의 두개의 연속계의 진동수에 매우 근접한 값이다. 더욱 정확한 진동수를 얻고자 하면 보의 분할을 더 많이 하면 된다. 실제의 Simulation의 출력을 부록 2에 표시하였다.

## 4. Simulation 프로그램의 종류<sup>2,3,4,5,6</sup>

진동 시스템을 Simulation하기 위한 다음의 10가지 프로그램을 개발하였다.

### (1) DFVSTEPFN

일정한 단계 기진력이 작용하는 일자유도 감쇄 강제 진동시스템의 시간반응을 Runge-Kutta 적분법에 의하여 해석하는 프로그램이다.

### (2) DFVARBTFN

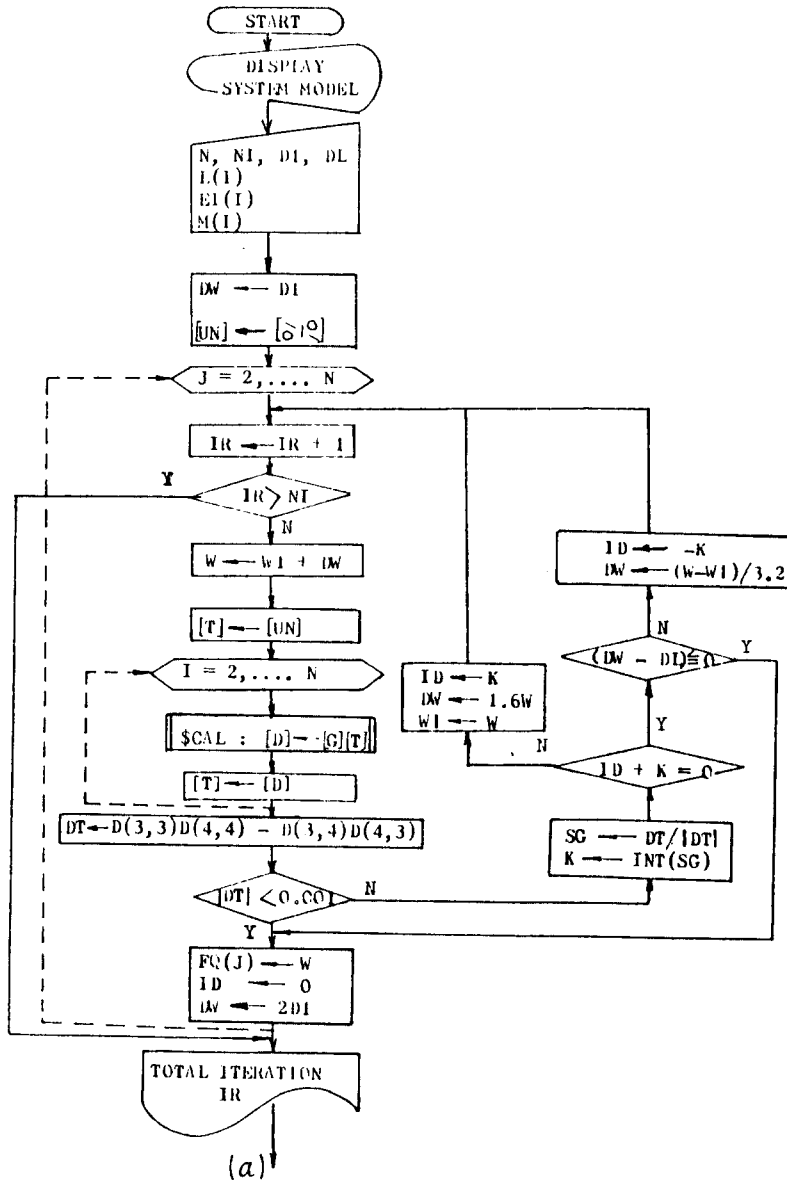
임의의 기진력이 작용하는 일자유도 감쇄 강제 진동시스템의 시간반응을 Runge-Kutta 적분법에 의하여 해석하는 프로그램이다.

### (3) SIMPLEPEN

비선형 단순 단진자 시스템의 시간반응을 Average-Acceleration 방법으로 해석하는 프로그램이다.

### (4) HARDENSPR

하중과 스프링의 변위가 3차 곡선의 관계를 가지는 비선형 일자유도 감쇄 진동시스템의 시간반응을 Average-Acceleration 방법으로 해석하는 프로그램이다.



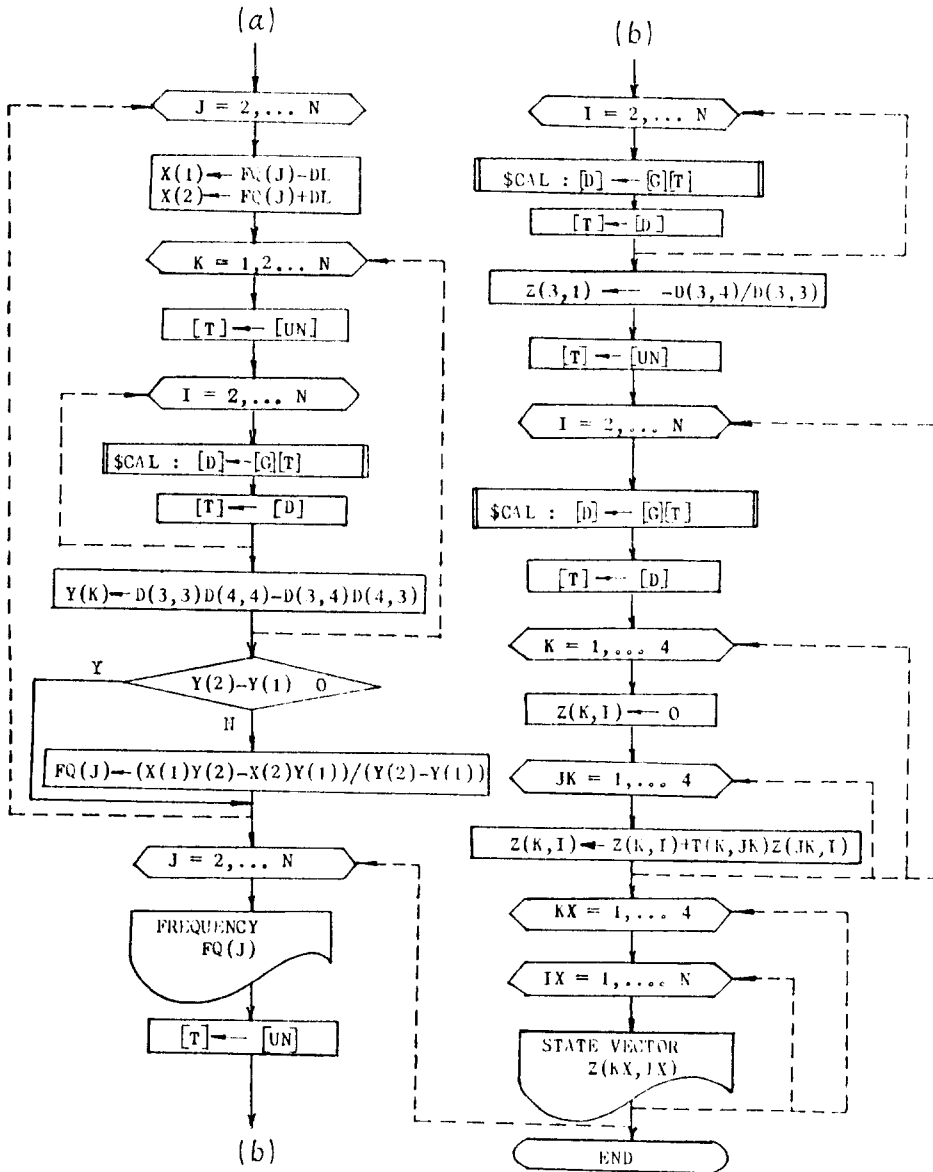


그림 4. 프로그램 CANTILEVE의 흐름도

(5) DFVIMPEDN

조화 기진력이 작용하는 일자유도 감쇄 진동 시스템의 진폭비와 위상각을 Impedance 방법으로 해석하는 프로그램이다.

(6) UDSARBT FN

임의의 기진력이 작용하는 비감쇄 불연속 시스템의 시간반응을 주좌표계(Principal Coordinate)를 이용하여 해석하는 프로그램이다.

(7) UDSMATRIX

비감쇄 불연속 시스템의 고유진동수 및 Eigen-Vector를 Duncan-Collar의 Matrix 반복법에 의하여 해석하는 프로그램이다.

(8) MULTMODAL

Modal 감쇄가 존재하는 3 자유도 진동 시스템에 임의의 기진력이 작용할때 이진동 시스템의 고유진동수 및 Eigen-Vector와 시간반응을 구하는 프로그램이다.

(9) MULTROTOR

다중 회전자 시스템의 고유진동수를 Rayleigh 방법에 의하여 해석하는 프로그램이다.

(10) CANTILEVE

균일 외팔보의 고유진동수 및 상태벡터

를 Myklesstad-Prohl 방법에 의하여 해석하는 프로그램이다.

5. Program Package의 구성 및 운용

위에서 설명한 10 가지의 프로그램을 5 1/4 Inch Floppy Disk에 수록하여 하나의 Package로 운용한다. Package의 구성도는 그림 5와 같다. Disk를 Driver Unit에 삽입하여 실행시키면 화면에 표 1과 같은 Package Menu가 나타난다. (A)는 각종 프로그램을 설명하는 안내문이고, (B)는 Disk에 수록된 모든 프로그램을 표 2과 같이 화면에 나타낸다. 표 2를 보면 한 문제를 해석하는 프로그램이 2-3개의 File로 분리되어 있는 것은 Micro-Computer의 기억용량(48K)이 부족하여 몇개의 File로 분리하여 Disk에 저장시킨 것이다.

실행할 때에는 Operating Command에 의하여 필요에 따라 각 File을 호출하여 실행한다. 프로그램 UDSMATRIX를 실행할 경우 각 File이 어떻게 접속되는가를 그림 6에 예시하였다.

각종 프로그램의 출력은 문제에 따라 여러 가지 형태로 표시된다. 대표적인 예를 부록 3에 첨부하였다.

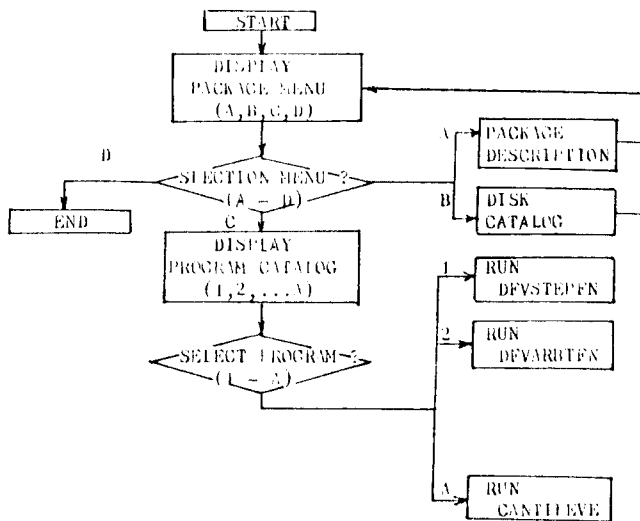


그림 5. Package의 구성도



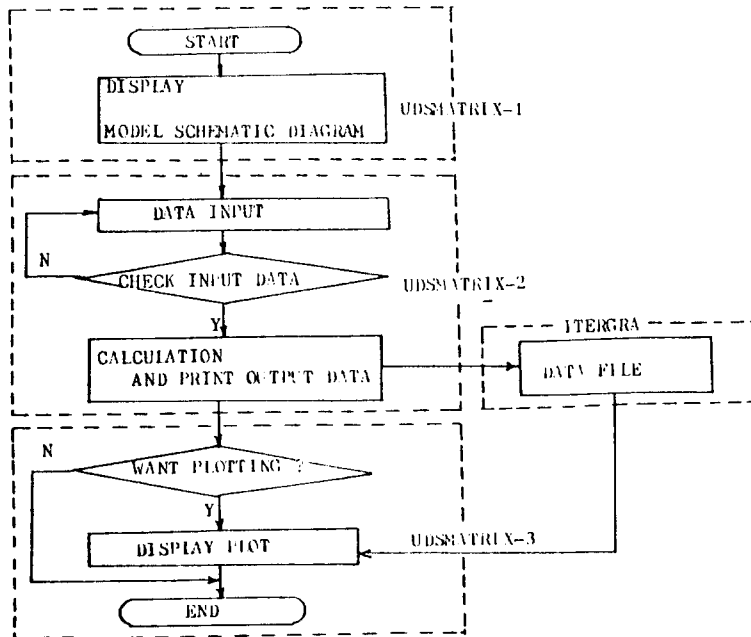


그림 6. 프로그램 UDSMATRIX의 구성도

=====

VIBRATION SIMULATION PROGRAM PACKAGE.

YEUNG NAM UNIV. MECHANICAL ENG. DEPT.

(CC 1984. SYSTEM ENG. LABORATORY

=====

- [ A ] PROGRAM DESCRIPTION
- [ B ] DISK CATALOG
- [ C ] RUN PROGRAM PACKAGE
- [ D ] QUIT

SELECT ONE ----->

표 1. Package의 Menu

DISK VOLUME 254

```

*A 010 HELLO
*B 008 CHARGEN.BIN
*A 008 DFVSTEPFN-1
*A 010 DFVSTEPFN-2
*A 008 DFVARETFN-1
*A 010 DFVARETFN-2
*A 007 SIMPLEPEN-1
*A 010 SIMPLEPEN-2
*A 008 HARDENSPR-1
*A 011 HARDENSPR-2
*A 008 DFVIMFEDN-1
*A 012 DFVIMFEDN-2
*A 011 UDSARETFN-1
*A 028 UDSARETFN-2
*A 010 UDSMATRIX-1
*A 018 UDSMATRIX-2
*A 008 UDSMATRIX-3
*A 012 MULTMODAL-1
*A 030 MULTMODAL-2
*A 007 MULTMODAL-3
*A 011 MULTROTOR-1
*A 011 MULTROTOR-2
*A 011 CANTILEVE-1
*A 018 CANTILEVE-2
*A 014 DEMO
T 002 DYNAS3
T 003 ITERGRA
    
```

표2. Disk Catalog

## 6. 결 론

Micro-Computer의 보급이 대중화 되면서 누구나 손쉽게 사용할 수 있는 BASIC 언어로써 여러가지 진동시스템의 동적 특성을 해석하는 Simulation 프로그램을 개발하였다. 그러나 8 Bit Micro-Computer 가 기억용량이 적고 연산속도가 늦어서 대형 프로그램을 사용할 수 없고 소수점 이하 수치의 정확성이 낮아 Matrix Iteration에 따른 누적 오차가 발생한다. 이러한 문제는 16 Bit 256K-500 K RAM 을 갖는 Micro-Computer 에서는 해결된다.

그러므로 Micro-Computer의 장점을 최대한으로 활용할 수 있는 Software 개발이 계속 연구되어야 한다.

본 연구에서 개발된 진동 시스템을 해석하

는 Simulation 프로그램들은 Micro-Computer의 기억용량 때문에 대형 문제를 처리할 수 없지만 일반 산업체에서 비교적 간단한 문제를 처리하는데 쉽게 사용할 수 있고 교육 목적으로 활용할 수 있다.

## 후 기

본 연구는 1984년도 한국 과학재단의 연구비 지원으로 이루어진 결과입니다. 한국 과학재단에 감사를 드립니다.

## 참고문헌

1. 주해호 "Simulation Language," 대한전기학회지 제 25 권 제 3 호, 1976년 5월
2. Tse, Morse, Hinkle: "Mechanical Vibration Theory and Application" 2nd Edition, Allyn and Bacon 1978.

3. Timoshenko, Young, Weaver : "Vibration Problems in Engineering" 4th Edition. John Wiley & Son, Inc. 1974.
4. Kuo : "Computer Applications of Numerical Method," Addison-Wesley 1972.
5. Roy E. Myer : "Micro-Computer Graphics," Addison-Wesley 1982.
6. "Apple II Basic Programing Manual," Apple Computer Inc. 1978.

## 부록 1. 프로그램 CANTILEVE의 Listing

```

100 TEXT : CLEAR
101 DIM D(4,4),EI(10),FQ(10),L(10),M(10),T(4,4),UN(4,4),X(2),Y
    (2),Z(4,11),G(4,4)
102 Z(1,1) = 0:Z(2,1) = 0:Z(4,1) = 1
103 HOME : VTAB 3: INVERSE
104 PRINT "  ]I E N T E R   D A T A ]I   ": NORMAL
105 VTAB 8
106 INPUT "NO. OF STATIONS ..... ";N
107 PRINT
108 INPUT "MAX. NO. OF ITERATIONS ..... ";NI
109 PRINT
110 PRINT "INITIAL INCREMENT IN - "
111 INPUT " - FREQUENCY ESTIMATE ..... ";D1
112 PRINT
113 INPUT "ALLOWABLE ERROR RATIO ..... ";DL
114 VTAB 20: INVERSE
115 PRINT "----> IS THIS CORRECT ? < Y/N > <---->"; GET A$
116 NORMAL : PRINT
117 IF A$ = "Y" THEN GOTO 119
118 GOTO 103
119 HOME : VTAB 3: INVERSE
120 PRINT "  ]I E N T E R   D A T A ]I   ": NORMAL
121 PRINT : PRINT
122 PRINT : PRINT "[ LENGTH OF SPAN BETWEEN STATIONS L(I) ]"
123 PRINT "[ FLEXURAL STIFFNESS OF EACH SPAN EI(I)]"
124 PRINT "[ VALUES OF POINT MASS M(I) ]"
125 NORMAL
126 PRINT : PRINT : FOR I1 = 2 TO N
127 PRINT " L ( "I1" ) = "; INPUT L(I1)
128 NEXT I1: PRINT
129 FOR I1 = 2 TO N
130 PRINT " EI ( "I1" ) = "; INPUT EI(I1)
131 NEXT I1: PRINT
132 FOR I1 = 2 TO N
133 PRINT " M ( "I1" ) = "; INPUT M(I1)
134 NEXT I1: PRINT : PRINT
135 INVERSE : PRINT "----> IS THIS CORRECT ? < Y/N > <---->"; GET
    A$
136 NORMAL : PRINT
137 IF A$ = "Y" THEN GOTO 139
138 GOTO 119
139 HOME : VTAB 5: HTAB 11: PRINT " ** SOLUTION **": PRINT : PRINT

140 FOR IO = 1 TO 4: FOR JO = 1 TO 4
141 UN(IO,JO) = 0
142 NEXT JO,IO
143 FOR IO = 1 TO 4

```

```

144 UN(I0,I0) = 1
145 NEXT I0
146 REM ***** II.APPROXIMATE FREQUENCIES.*****
147 DW = D1
148 FOR J = 2 TO N
149 IR = IR + 1
150 IF IR > NI THEN GOTO 174
151 W = W1 + DW
152 GOSUB 229
153 FOR I = 2 TO N
154 GOSUB 236
155 GOSUB 308
156 NEXT I
157 DT = D(3,3) * D(4,4) - D(3,4) * D(4,3)
158 IF ABS (DT) < 0.001 THEN GOTO 170
159 S6 = DT / ABS (DT)
160 K = INT (S6)
161 IF (ID + K) = 0 THEN GOTO 166
162 ID = K
163 DW = 1.6 * DW
164 W1 = W
165 GOTO 149
166 IF (DW - DL) < = 0 THEN GOTO 170
167 ID = - K
168 DW = (W - W1) / 3.2
169 GOTO 149
170 FQ(J) = W
171 ID = 0
172 DW = 2 * D1
173 NEXT J
174 PRINT "TOTAL ITERATIONS FOR APPROX.FREQ =" ; IR
175 REM ***** GET MORE EXACT FREQUENCY BY INTERPOLATION.*****
*
176 FOR J = 2 TO N
177 X(1) = FQ(J) - DL
178 X(2) = FQ(J) + DL
179 FOR K = 1 TO 2
180 GOSUB 229
181 FOR I = 2 TO N
182 GOSUB 260
183 GOSUB 308
184 NEXT I
185 Y(K) = D(3,3) * D(4,4) - D(3,4) * D(4,3)
186 NEXT K
187 IF ABS (Y(2) - Y(1)) < = 0 THEN GOTO 189
188 FQ(J) = (X(1) * Y(2) - X(2) * Y(1)) / (Y(2) - Y(1))
189 NEXT J
190 REM ***** IV.CALAUULATE STATE VECTORS FOR EACH FREQUENCY.*
****
191 PRINT : PRINT
192 PRINT "STATE VECTORS FO STATIONS (1) TO (N):-"
193 PRINT : PRINT "-----"
194 PRINT "[Y]:DEFLECTION, Y1,Y2,...YN"
195 PRINT "[P]:SLOPE, P11,P12,...PIN"
196 PRINT "[M]:MOMENT, M1,M2,M3,...MN"
197 PRINT "[V]:SHEAR STRESS, V1,...VN"
198 PRINT "-----"
199 PRINT : PRINT
200 FOR J = 2 TO N
201 PRINT "J[ FREQUENCY "FQ(J)" RAD/S. ]I " : PRINT

```

```

202 GOSUB 229
203 FOR I = 2 TO N
204 GOSUB 284
205 GOSUB 308: NEXT I
206 Z(3,1) = - D(3,4) / D(3,3)
207 GOSUB 229
208 FOR I = 2 TO N
209 GOSUB 284
210 GOSUB 308
211 FOR K = 1 TO 4
212 Z(K,1) = 0
213 FOR JK = 1 TO 4
214 Z(K,1) = Z(K,1) + T(K,JK) * Z(JK,1)
215 NEXT JK: NEXT K; NEXT J
216 FOR KX = 1 TO 4
217 ON KX GOTO 218,219,220,221
218 AA$ = "[Y] ": GOTO 222
219 AA$ = "[PI] ": GOTO 222
220 AA$ = "[M] ": GOTO 222
221 AA$ = "[V] i": GOTO 222
222 PRINT AA$;
223 FOR IX = 1 TO N
224 PRINT TAB( IX * 7); INT (Z(KX,IX) * 1000) / 1000;
225 NEXT IX: PRINT
226 NEXT KX: PRINT : PRINT
227 NEXT J
228 END
229 REM ***** $ SUBN *****
230 FOR I1 = 1 TO 4
231 FOR J1 = 1 TO 4
232 T(I1,J1) = UN(I1,J1)
233 NEXT J1,I1
234 RETURN
235 END
236 REM ***** $ CAL *****
237 FOR AI = 1 TO 4: FOR BI = 1 TO 4
238 G(AI,BI) = 0
239 NEXT BI,AI
240 FOR CI = 1 TO 3
241 G(CI,CI) = 1
242 NEXT CI
243 G(1,2) = L(I)
244 G(1,3) = L(I) ^ 2 / (2 * EI(I))
245 G(1,4) = - G(1,3) * L(I) / 3
246 G(2,3) = L(I) / EI(I)
247 G(2,4) = - G(1,3)
248 G(3,4) = - L(I)
249 G(4,1) = - M(I) * W ^ 2
250 G(4,2) = G(4,1) * L(I)
251 G(4,3) = G(4,1) * G(1,3)
252 G(4,4) = 1 + G(1,4) * G(4,1)
253 FOR I2 = 1 TO 4: FOR J2 = 1 TO 4
254 D(I2,J2) = 0
255 FOR K2 = 1 TO 4
256 D(I2,J2) = D(I2,J2) + G(I2,K2) * T(K2,J2)
257 NEXT K2,J2,I2
258 RETURN
259 END
260 REM ***** $ CAL *****
261 FOR AI = 1 TO 4: FOR BI = 1 TO 4
262 G(AI,BI) = 0

```

```

277 FOR I2 = 1 TO 4: FOR J2 = 1 TO 4
278 D(I2,J2) = 0
279 FOR K2 = 1 TO 4
280 D(I2,J2) = D(I2,J2) + G(I2,K2) * T(K2,J2)
281 NEXT K2,J2,I2
282 RETURN
283 END
284 REM ***** $ CAL *****
285 FOR AI = 1 TO 4: FOR BI = 1 TO 4
286 G(AI,BI) = 0
287 NEXT BI,AI
288 FOR CI = 1 TO 3
289 G(CI,CI) = 1
290 NEXT CI
291 G(1,2) = L(I)
292 G(1,3) = L(I) ^ 2 / (2 * EI(I))
293 G(1,4) = - G(1,3) * L(I) / 3
294 G(2,3) = L(I) / EI(I)
295 G(2,4) = - G(1,3)
296 G(3,4) = - L(I)
297 G(4,1) = - M(I) * FB(J) ^ 2
298 G(4,2) = G(4,1) * L(I)
299 G(4,3) = G(4,1) * G(1,3)
300 G(4,4) = 1 + G(1,4) * G(4,1)
301 FOR I2 = 1 TO 4: FOR J2 = 1 TO 4
302 D(I2,J2) = 0
303 FOR K2 = 1 TO 4
304 D(I2,J2) = D(I2,J2) + G(I2,K2) * T(K2,J2)
305 NEXT K2,J2,I2
306 RETURN
307 END
308 REM ***** $ SUBN *****
309 FOR I1 = 1 TO 4
310 FOR J1 = 1 TO 4
311 T(I1,J1) = D(I1,J1)
312 NEXT J1,I1
313 RETURN
314 END

```

]

부록 2. CANTILEVE의 실행 예

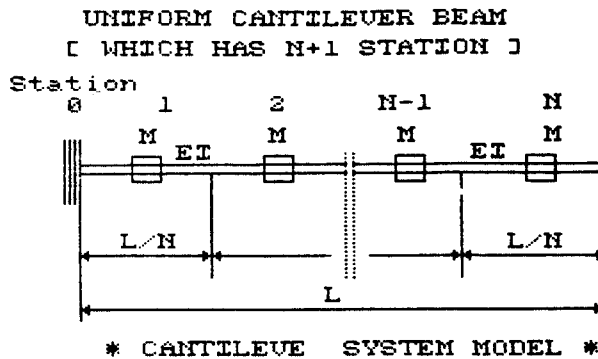


그림 A-1. CANTILEVE System Model

표 A-1. CANTILEVE의 데이터 입력 예

```

JC ENTER DATA JC

NO. OF STATIONS ..... 5
MAX. NO. OF ITERATIONS ..... 200
INITIAL INCREMENT IN -
- FREQUENCY ESTIMATE ..... 1
ALLOWABLE ERROR RATIO ..... 0.001
----> IS THIS CORRECT ? < Y/N > <----

JC ENTER DATA JC

[ LENGTH OF SPAN BETWEEN STATIONS L(I) ]
[ FLEXURAL STIFFNESS OF EACH SPAN EI(I) ]
[ VALUES OF POINT MASS M(I) ]

L ( 2 ) = ?0.125
L ( 3 ) = ?0.25
L ( 4 ) = ?0.25
L ( 5 ) = ?0.25

EI ( 2 ) = ?1
EI ( 3 ) = ?1
EI ( 4 ) = ?1
EI ( 5 ) = ?1

M ( 2 ) = ?0.25
M ( 3 ) = ?0.25
M ( 4 ) = ?0.25
M ( 5 ) = ?0.25

----> IS THIS CORRECT ? < Y/N > <----
    
```

표 A-2. 표 A-1의 입력에 대한 출력

```

** SOLUTION **

TOTAL ITERATIONS FOR APPROX.FREQ =102

STATE VECTORS FO STATIONS (1) TO (N):-
-----
[V]:DEFLECTION, Y1,Y2,...YN
[P]:SLOPE, P11,P12,...PIN
[M]:MOMENT, M1,M2,M3,...MM
[V]:SHEAR STRESS, V1,...VN
-----

JC FREQUENCY 3.5670918 RAD/S. JC

[V] : 0      5E-03 .041 .099 .167
[P] : 0      .081 .198 .258 .275
[M] : .716   .591 .345 .133 0
[V] : 1      .983 .85 .532 0

JC FREQUENCY 23.1824411 RAD/S. JC

[V] : 0      1E-03 5E-03 4E-03 -5E-03
[P] : 0      .017 .01 -.023 -.041
[M] : .202   .077 -.131 -.14 0
[V] : 1      .831 .039 -.56 -1E-03

JC FREQUENCY 66.3158108 RAD/S. JC

[V] : 0      0      1E-03 -2E-03 0
[P] : 0      6E-03 -7E-03 -4E-03 .011
[M] : .116   -9E-03 -.097 .12 0
[V] : 1      .354 -.868 .48 -1E-03

JC FREQUENCY 140.980398 RAD/S. JC

[V] : 0      0      -1E-03 0      -1E-03
[P] : 0      1E-03 -2E-03 0      -1E-03
[M] : .075   -.05 .026 -.012 0
[V] : 1      -.307 .151 -.045 0
    
```

부록 3. 프로그램의 출력 예

표 A-3. DFVIMPEDN 데이터 입력예

```

JC DATA INPUT JC
DAMPING FACTOR (Z>0) = ?0.1
STARTING FREQUENCY RATIO (RI=0) = ?0
ENDING FREQUENCY RATIO (RI<=3) = ?3
FREQUENCY RATIO INCRESEMENT(QT<=0.1) = ?0.1
----> IS THIS CORRECT ? ( Y/N ) (----)
    
```

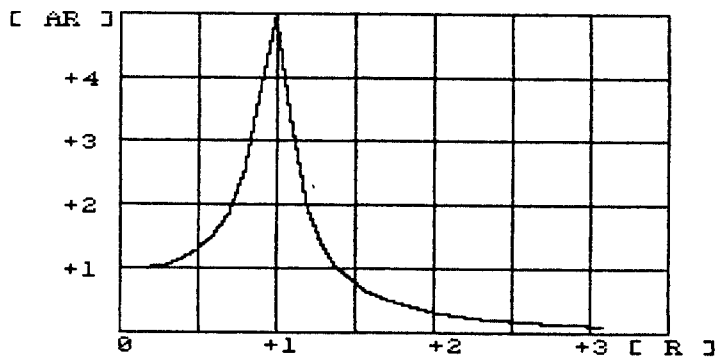
표 A-4. 표 A-3의 입력에 대한 DFVIMPEDN의 출력

```

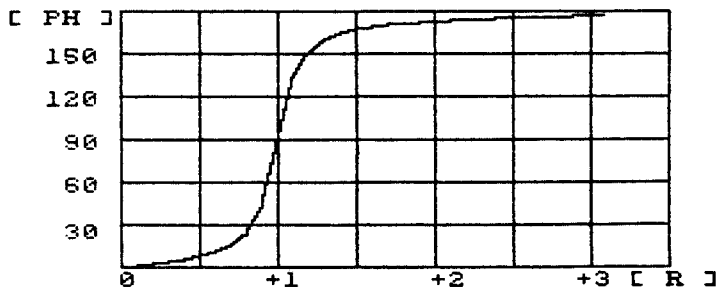
** SOLUTION **
FREQ. RATIO  AMPL.RATIO  PHASE ANG(DEG)
    
```

FREQ. RATIO	AMPL.RATIO	PHASE ANG(DEG)
0	1	0
.1	1.00989495	1.15733752
.2	1.04076362	2.38595321
.3	1.09652023	3.77229813
.4	1.18511366	5.44035297
.5	1.3216372	7.5946726
.6	1.53573779	10.6196962
.7	1.89083559	15.3501956
.8	2.53836542	23.9625812
.9	3.82080361	43.4520097
1	5	89.9996543
1.1	3.28797974	133.667602
1.2	1.9952172	151.38943
1.3	1.35618936	159.35293
1.4	.999999995	163.739733
1.5	.77790984	166.504215
1.6	.627950446	168.40778
1.7	.520741521	169.80186
1.8	.440772486	170.869789
1.9	.379144364	171.716258
2	.330409299	172.405327
2.1	.291055761	172.978379
2.2	.258723768	173.463341
2.3	.23177165	173.879757
2.4	.209023963	174.241718
2.5	.189618185	174.559647
2.6	.172907934	174.841428
2.7	.158399855	175.093138
2.8	.145711301	175.319538
2.9	.134541256	175.524418
3	.124649914	175.71083
3.1	.115844062	175.881265

WANT GRAPIC ? (Y/N)



AMPLITUDE RATIO [ AR ] - VERSUS - FREQUENCY RATIO [ R ]



PHASE ANGLE [ PH ] - VERSUS - FREQUENCY RATIO [ R ]

그림 A-4의 Graphic 실행 예



표 A-5. 프로그램 MULTIMODAL 데이터 입력 예

```

[ DATA INPUT ]

NUMBER OF DEGREE OF FREEDOM(<=3)...3
NUMBER OF TIME STEP ( <= 100 ) ...20
ALLOWABLE ERROR FOR EIGENVECTORS..0.001
MODAL DAMPING RATIO (GAMMA).....
      GAMMA (1)=0.05
      GAMMA (2)=0.05
      GAMMA (3)=0.05
-> TYPE OF INDICATOR<-----

[ INPUT MATRIX IS STIFF.OR FLEX.]
(0:STIFFNESS, 1:FLEXIBILITY).....0

[ EXCITATION FORCING FUNCTION ]
  1 : STEP FUNCTION
  2 : RAMP FUNCTION
  3 : ARBITRARLY FUNCTION

TYPE 1 , 2 , 3 ---->1

----> IS THIS CORRECT ? ( Y/N ) <----

\ ENTER THE DATA OF STIFFNESS OR
  FLEXIBILITY MATRIX : A(I,J) \

A ( 1,1 ) = ?2
A ( 1,2 ) = ?-1
A ( 1,3 ) = ?0

A ( 2,1 ) = ?-1
A ( 2,2 ) = ?2
A ( 2,3 ) = ?-1

A ( 3,1 ) = ?0
A ( 3,2 ) = ?-1
A ( 3,3 ) = ?1

----> IS THIS CORRECT ? ( Y/N ) <----

INPUT DATA [ M(I) ], [ B(I) ], [ C(I) ]

[ M(I) ] : MASS VECTOR
[ B(I) ] : INITIAL DISPLACEMENT
[ C(I) ] : INITIAL VELOCITY

M ( 1 ) = ?1
M ( 2 ) = ?1
M ( 3 ) = ?1

B ( 1 ) = ?0
B ( 2 ) = ?0
B ( 3 ) = ?0

C ( 1 ) = ?0
C ( 2 ) = ?0
C ( 3 ) = ?0

----> IS THIS CORRECT ? ( Y/N ) <----
INPUT DATA : STEP FORCING FUNCTION

[ P(I) ] : VECTOR OF LOAD FACTOR
[ DT ] : TIME INCRESEMENT
[ FD ] : FORCE MAGNITUDE

[ P(1) ] .....0
[ P(2) ] .....0
[ P(3) ] .....1
[ DT ] .....1
[ FD ] .....1

----> IS THIS CORRECT ? ( Y/N ) <----

```

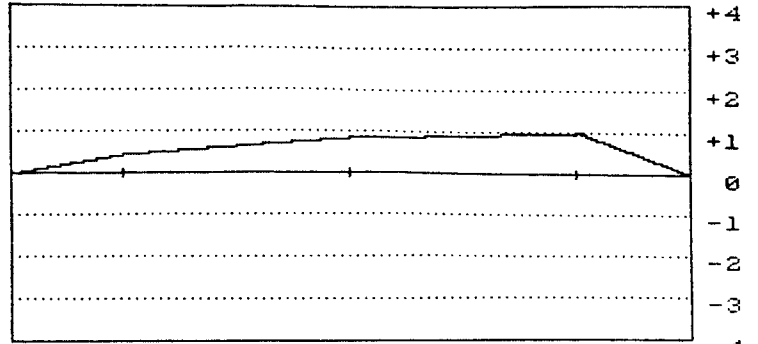
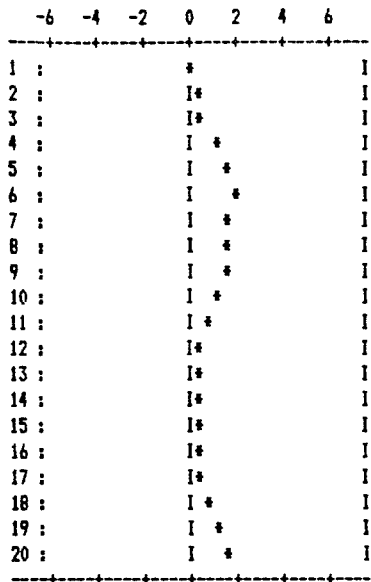
표 A - 6. 표 A - 5의 입력에 대한 MULTIMODAL 실행 예

				RESPONSE IN ORIGINAL COORDINATES (PRINTED COLUMN-WISE)			
MULTIMODAL: DYNAMIC RESPONSE OF FIRST THREE MODE OF DAMPED MULTI-DEGREE SYSTEM TO PIECEWISE-CONSTANT FORCING FUNCTION				TIME	X1	X2	X3
THREE EIGENVALUES AND VECTORS BY ITERATION				1	3.04584918E-03	.0425409327	.449311548
[ MODE 1. ]				2	.0753878392	.450670856	1.4233209
EIGENVALUE .....	5.05095542			3	.09051256	1.36132664	2.41768225
ANGULAR FREQUENCY.....	.444952071			4	1.19426633	2.37027383	3.34088616
NUMBER OF ITERATIONS...	4			5	1.85733253	3.16343263	4.24638319
[ MODE 2. ]				6	2.01675666	3.69304241	5.02031155
EIGENVALUE .....	.644060711			7	1.87980486	3.861289	5.43155943
ANGULAR FREQUENCY.....	1.24605323			8	1.81076519	3.64959377	5.27435373
NUMBER OF ITERATIONS...	10			9	1.74088399	3.22671759	4.54876717
[ MODE 3. ]				10	1.44056461	2.63674832	3.56268996
EIGENVALUE .....	.308802905			11	.93326203	1.83185093	2.66920165
ANGULAR FREQUENCY.....	1.79953091			12	.422546238	1.02711138	1.94871485
NUMBER OF ITERATIONS...	2			13	.113041699	.543254621	1.34874427
[ MODAL MATRIX ]				14	.104453843	.423582396	.984248905
.445145019	-1.24808955	1.78610357		15	.283211379	.552920148	1.05793954
.802017655	-.55264866	-2.2383115		16	.457560463	.908600462	1.59311428
1	1	1		17	.643897282	1.46670893	2.40264212
				18	.984782094	2.1121408	3.24076933
				19	1.42439539	2.73106063	3.92920209
				20	1.73232572	3.20731764	4.40634714

[[ GRAPH WANT ? ( Y/N ) ]]

\* SCALE FACTOR (--- 0.5

[[ TIME RESPONSE OF MODE 1 ]]



[[ TIME RESPONSE OF MODE 2 ]]

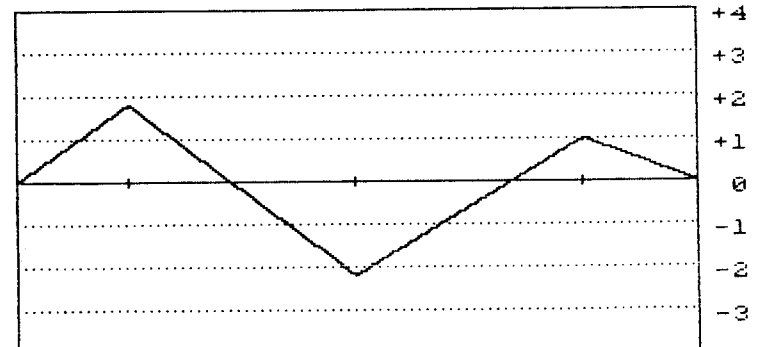
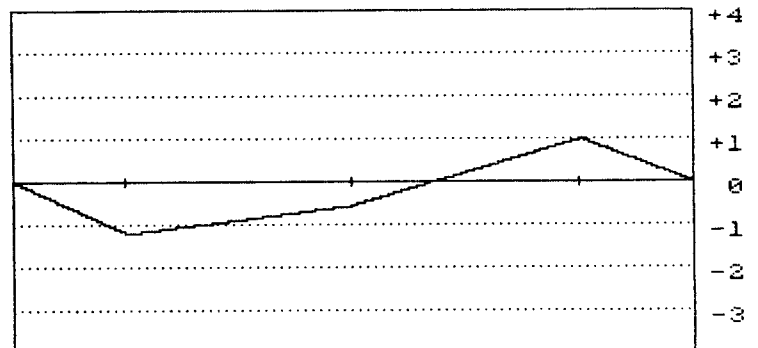
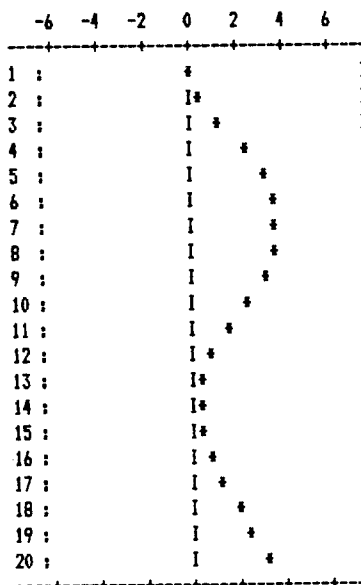


그림 A-3. 표 A-6의 출력에 대한 시간반응과 고유벡터의 Graphic