SIMNON - User's Guide

Elmqvist, Hilding

1973

Document Version:
Publisher's PDF, also known as Version of record

SIMNON

User's guide

Hilding Elmqvist

INTRODUCTION.

SIMNON is an interactive simulation package for nonlinear systems:

$$\frac{dx}{dt} = f(x,t)$$

$$y = h(x,t).$$

To describe the system, the equations are given in a special language, which are compiled into a pseudocode interpreted by an executionroutine.

SIMNON has ten commands in order to read system description from either teletype or disk, display values, change initial value of state variable, change parameter value, tell what identifiers to plot, draw axes, perform computations, simulate and return to monitor.

SYSTEM DESCRIPTION.

The syntax for system description is given in appendix 1. Used notations have the following meanings:

:=	constitute
/	or
$\{arg\}_i^j$	arg is repeated at least i and at most j times, if i and j are missing they are assumed to equal 1
---	except

<system description> is built up by:

1. <X0 definition>
   Used to give initial value of <state variable>. If no initial value is given it is assumed zero.

   Ex. X1:5

2. <parameter definition>
Used to give a <parameter> desired value.

Ex. ALPHA:0.5

3. <own function definition>
Own functions (called FCN) can be defined by giving a table of increasing argument values and corresponding function values. Linear interpolation will be done at execution time.

```
Ex.  *FCN5
     .5    2E1
     1.5   3.5E-3
     3     .2
     *FIN
```

4. <assignment statement>
<derivative>:s and <variable>:s are given values in <assignment statement>:s. The syntax is much like ALGOL:s.

<one argument function> and <two argument function> has exactly the same meanings as in FORTRAN.

The execution routine is made so that if a <boolean expression> is true the result is 1.0 else -1.0. A <boolean expression> is true if the corresponding value is greater than or equal zero. The last fact should be remembered when using <parameter>:s as switches.

All computations are performed in floating arithmetic, hence a relation ARG1 = ARG2 is considered true if |ARG1-ARG2| < 1E-6 .

```
Ex.  ABC=X1-5/X2+SIN(T)
     T59=8-ATAN2(P+EXP(X1),AINT(T))+FCN3(ABC)
     DX1=IF T59 > 2 THEN 1 ELSE 2
     DX2=IF DX1 < 5 THEN(IF B THEN X1 ELSE X2) ELSE 7
     DX3=IF DX1 < 5 THEN X1 ELSE IF B THEN X2 ELSE 7
```

5. <comment>

<comment> could be just a <line terminator> or

Ex. " THIS IS A COMMENT

<X0 definition>/<parameter definition>/<own function definition> is a matter of saving numerical values during compilation, hence they could be inserted in any order in <system description>.

Assignments are performed in the same order as the <assignment statement>:s are given.

A <name> is considered as a <parameter> either if its first appearence is in a <parameter definition> or in the right hand part of an <arithmetic expression>.

End of <system description> is signalled by *END.

Blanks are not treated by the syntax, but must be separating for example <boolean word> and <identifier>. Blanks could be inserted anywhere except in <name>/<reserved word>/<unsigned number>/FIN/END.

COMMANDS.

The syntax for commands is given in appendix 3. Further comments are given below.

SYSTT{<null>/<filename>}<comment>

Used when you want to type in <system description> from the teletype. If you want to save <system description> on the disk, give wanted <filename>.

Each line is immediately compiled and is then written on the display, maybe together with an error message (see appendix 2). Incorrect lines are ignored and are not written on the disk.

Ex. >SYSTT
SYSTT TEST

SYSDK<filename><comment>

The <system description> will be read from the file on disk with given <filename> and extension NON.

Ex. SYSDK TEST

DISP<comment>

Displays value of <identifier>:s and initial value of <state variable>:s.

Ex. DISP
DISP " DISPLAY VALUES

CX0<X0 definition>

Changes initial value of <state variable>.

Ex. CX0 X2:0.2

CPAR<parameter definition>

Changes value of <parameter>.

Ex. CPAR BETA:10

PLOT<identifier>←{<identifier>}$_{1}^{0}$<comment>

Used to tell what <identifier>:s to plot. The horizontal <identifier> stands to the left of ← and the vertical <identifier>:s (1-10) to the right.

Ex. PLOT T ←X2 VAR
PLOT X1←X2

AXIS{<null>/{H/V}<minimum value><maximum value>}<comment>

Draws axes on display. H (horizontal) and V (vertical) are used to indicate which axis that is concerned. To erase the display and draw axes with unaltered scaling, just use AXIS.

Ex. AXIS
AXIS H 0 20
AXIS V -10 10

COMPU{<start value>/,}$_i^i${<stop value>/,}$_j^j${<number of intervals>/,}$_k^k${{MARK}$_1^\infty$}$_1^1$<comment>

The equations will be executed <number of intervals> + 1 times while T varies linearly in the interval <start value> - <stop value>.
Concerning MARK: see below.

Ex. COMPU 0 10 100 MARK
COMPU , , 200
COMPU

SIMU{<start value>/,}$_i^i${<stop value>/,}$_j^j${<number of intervals>/,}$_k^k${{CONT/MARK/<upper error bound>}$_1^\infty$}$_1^1$<comment>

Used to simulate dynamical systems. The independent variable T will vary in the interval <start value> - <stop value> with initial increment
(<stop value> - <start value>)/<number of intervals>.

Before simulation the initial values of the state vector is transferred to the state vector. If a simulation should be continued the transfer should be inhibited by placing CONT in command.

<upper error bound> is a bound for the mean of the <state variable>:s absolute error.

Ex. SIMU 0 1 100 MARK 1E-5
SIMU ,,, MARK CONT
SIMU 0 -2
SIMU

STOP<comment>

Return to monitor

Ex. STOP

The parameters in command COMPU and SIMU have the following initial values:

<start value>	0
<stop value>	1
<number of intervals>	100
<upper error bound>	0.001

If the value of a parameter in command COMPU or SIMU should be unaltered it could be replaced by a , (comma) or possibly the command could be shortened (indicated in the syntax by $1 \geq i \geq j \geq k \geq l \geq 0$).

When using MARK in command COMPU or SIMU marks will be placed on the curves at every ten point. The marks are used in the following order:

square, octagon, triangle, plus, cross, asterisk, horizontal bar, vertical bar.

To control execution of command COMPU and SIMU bits 0 and 1 of the DATA-switches on the consol is used as shown below:

bit 0	bit 1	
0	0	continue
0	1	continue
1	0	halt
1	1	read next command

If the command word should be the same as in last command it could be replaced by a , (comma).

```
Ex.  AXIS H 0 20
     , V -10 10
```

EXAMPLES.

As a first example how to use SIMNON consider the Mathieu equation

$$\frac{d^2y}{dt^2} + (a - 2q\ \cos 2t)y = 0$$

Let $x_1 = y$ and $x_2 = \frac{dy}{dt}$ giving the system

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = -(a - 2q\ \cos 2t)x_1$$

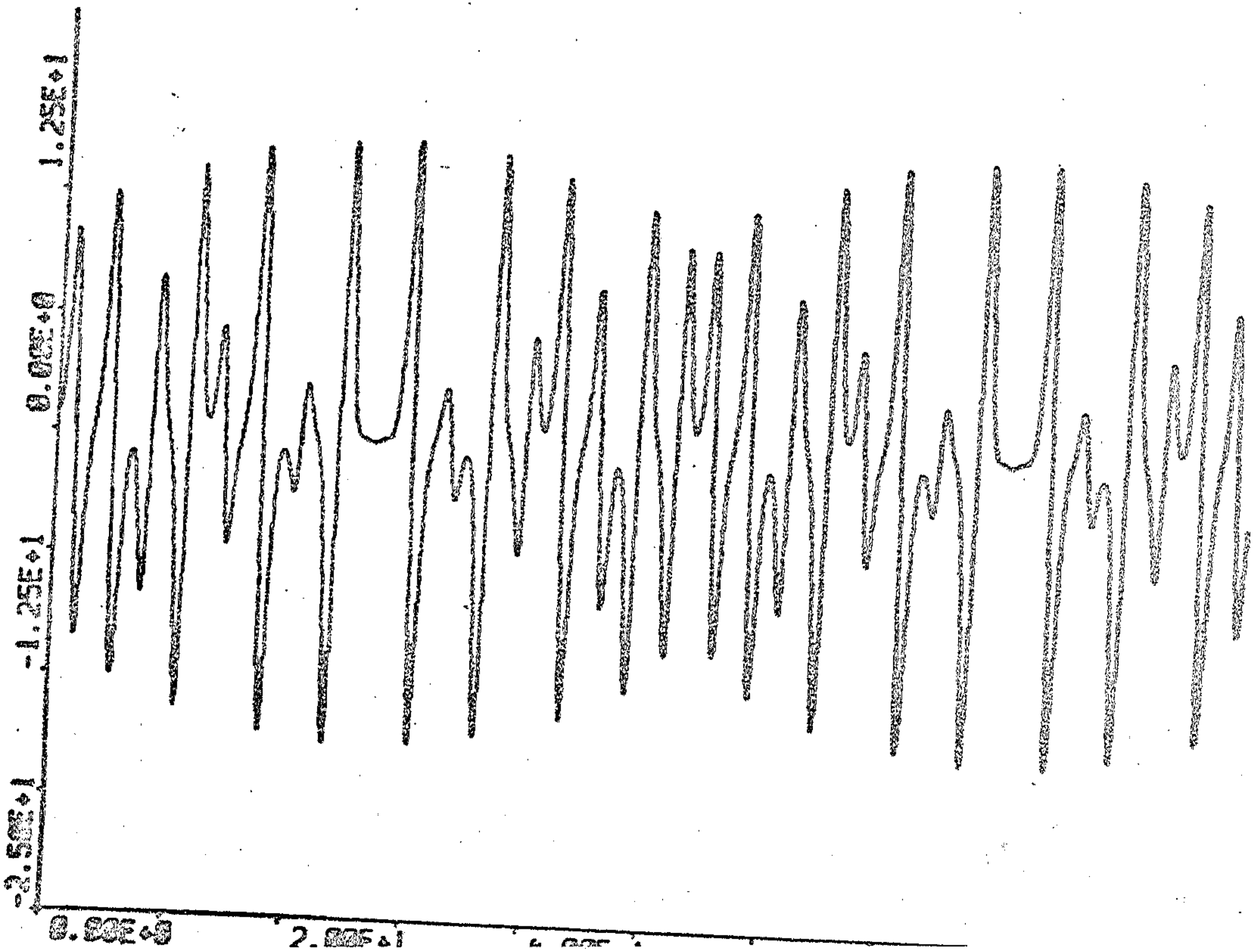
of first order differential equations.

The system has been simulated with different values of the parameters a and q. The used sequence of commands and the results on the display are shown below and on the following pages.

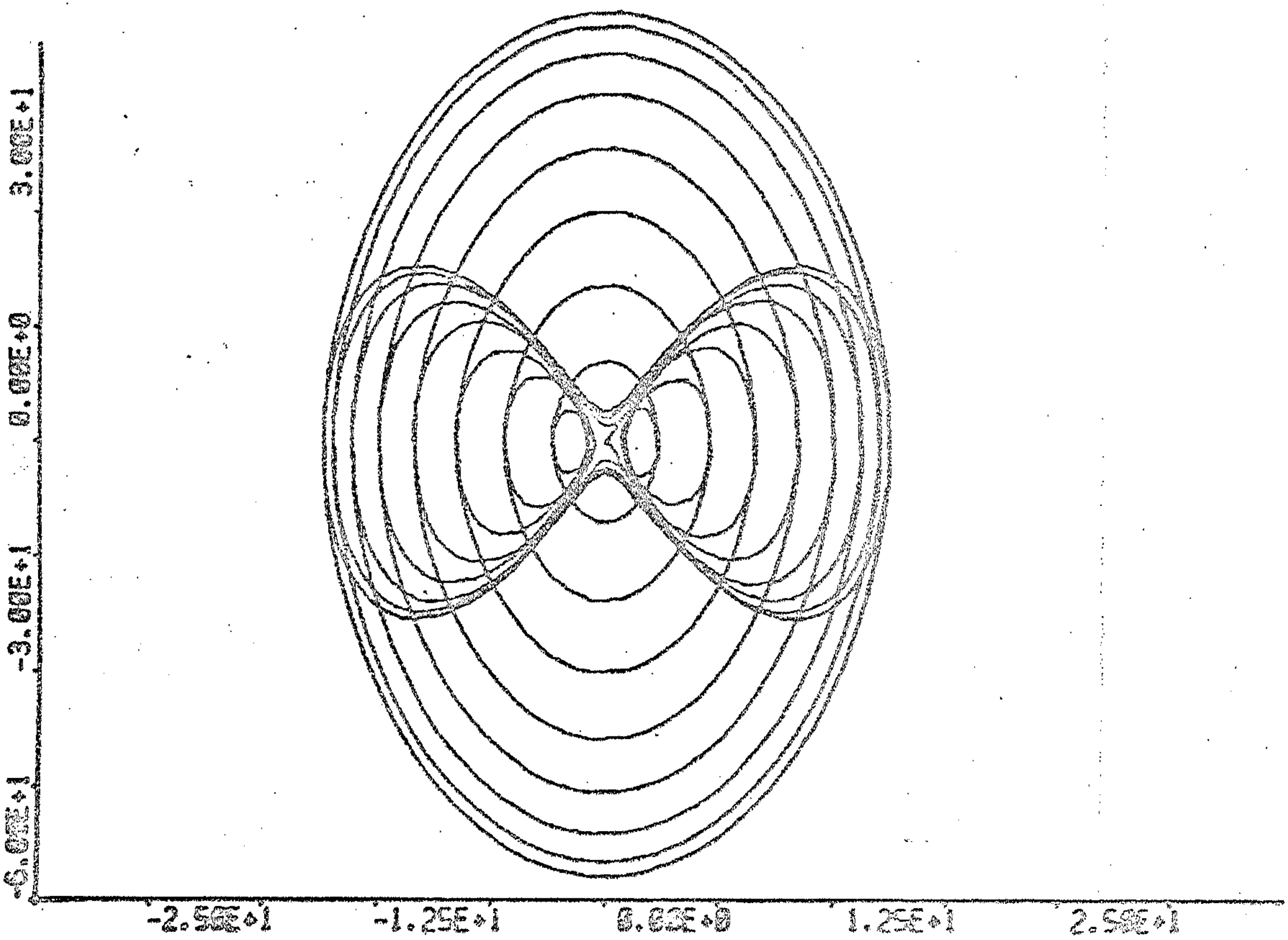
```
$E SIMNON


   SIMNON VIE


>" READ <SYSTEM DESCRIPTION> FOR THE MATHIEU EQUATION
>" FROM THE TELETYPE AND SAVE ON A FILE 'MATH' 'NON'
>SYSTT MATH

   <SYSTEM DESCRIPTION>:

"  MATHIEU EQUATION

DX1=X2
DX2=-(A-2*Q*COS(2*T))*X1

A:0
Q:7.55

*END


>
```

```
>" PLOT X1 VERSUS TIME
>PLOT T←X1   "  PICTURE 1    A:0  Q:7.55
>
>" DRAW AXES
>AXIS H 0 100
>,V -20 20
>
>" CHANGE INITIAL VALUE OF X1 FROM 0 TO 1
>CX0 X1;1
>
>" SIMULATE IN THE INTERVAL 0 - 100
>SIMU 0 100
>" SEE PICTURE 1
>
>" PLOT X2 VERSUS X1
>PLOT X1←X2   "  PICTURE 2    A:0  Q:7.55
>
>" DRAW AXES
>AXIS H -30 30
>,V -50 50
>
>" SIMULATE
>SIMU 0 54
>" SEE PICTURE 2
>
>" CHANGE PARAMETER VALUES
>CPAR A:0.1
>,Q:0.5
>
>" CHANGE HEADING ON THE DISPLAY
>PLOT X1←X2   "  PICTURE 3    A:0.1  Q:0.5
>
>" DRAW AXES
>AXIS H -2 2
>,V -1.5 1.5
>
>" SIMULATE
>SIMU 0 160
>" SEE PICTURE 3
>
>" DISPLAY PRESENT VALUES
>DISP
>" SEE PICTURE 4
>
```

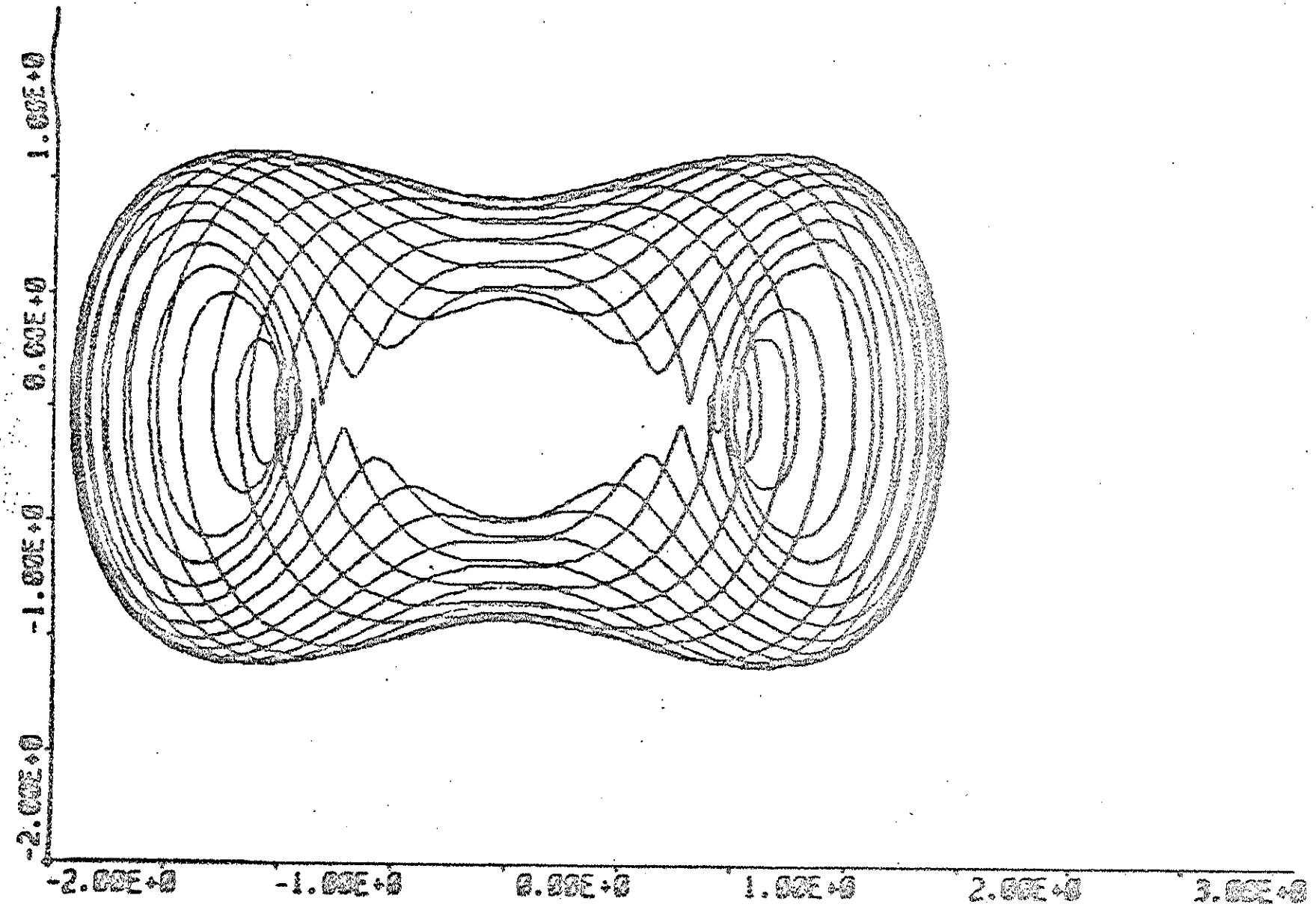
PLOT T-X1 - PICTURE 1 A:0 8:7.55



PLOT X1-X2 * PICTURE 2 A:0 B:7.55



PLOT X1-X2 - PICTURE 3 R:0.1 Q:0.5



PICTURE 4:

PRESENT VALUE OF <IDENTIFIER>S

<STATE VARIABLE>

X1 1.00227 X2 -0.114615

<DERIVATIVE>

DX1 -0.114615 DX2 0.005629

<TIME>

T 160.000

<VARIABLE>/<PARAMETER>

A 0.100000 B 0.500000

INITIAL VALUE OF <STATE VARIABLE>S

X1 1.00000 X2 0.002000

The second example intends to show how different kinds of arithmetic expressions are interpreted. For that purpose a file named 'DEM' 'NON' was prepared on the disk. See picture 5.

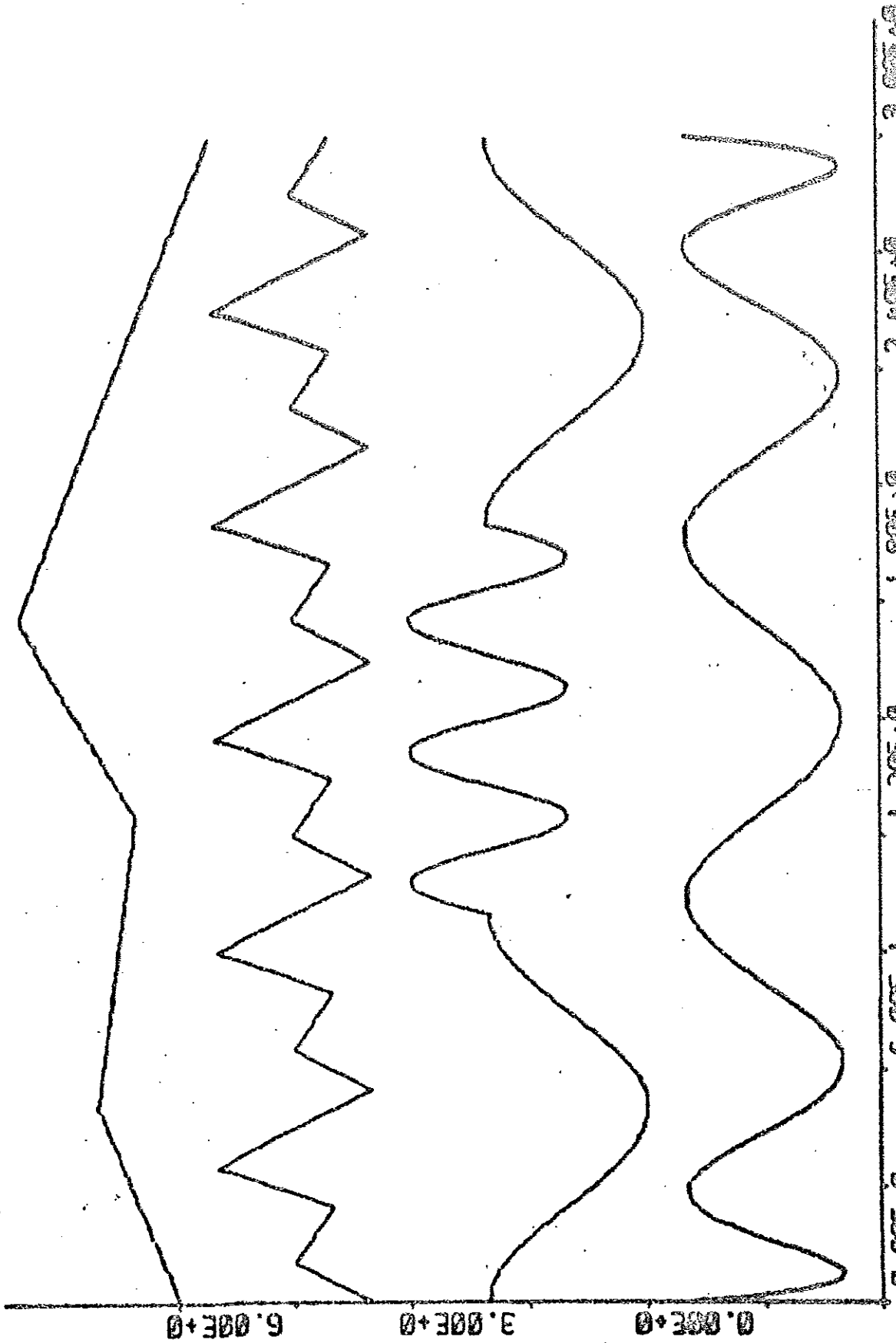
The variables are depending on <time> T and their values are evaluated for different values of T and plotted versus T.

The sequence of commands is shown below and the resulting time dependence of the variables is shown in picture 6.

```
>" READ <SYSTEM DESCRIPTION> FROM THE FILE 'DEM' 'NON'
>SYSDK DEM

>" SEE PICTURE 5
>
>PLOT T←FCN1A FCN1B IF1 CHEB  "  PICTURE 6
>
>AXIS H Ø 3
>,V -1 1Ø
>
>" COMPUTE AND PLOT VARYING T IN THE INTERVAL Ø - 3
>COMPU Ø 3 3ØØ
>" SEE PICTURE 6
>
```

PICTURE 5:

```
* <FILENAME>: DEM MOM

FCN1A=FCN1(2*T) +7.5

FCN1B=FCN1(AMOD(10*T,PER)) +5

PER:5.5
*FCN1                  * <OWN FUNCTION DEFINITION>
0 0                    * <ARGUMENT VALUE><FUNCTION VALUE>
1 1
2.5 0.5
3.5 2
4.5 1
*FIN

IF1=(IF T<1 OR T>2 THEN COS(N*T) ELSE SIN(3.5*T)+1) +2.5
N:6.2832

* CHEBYSHEV POLYNOMIAL
X=T/1.5-1
CHEB=COS(N* ATAN2(SQRT(1-X*X),X) )
N:10

*END
```


PLOT T-FCM1A FCM1B 1F1 CMSB * PICTURE *
0.00E+0
6.00E-1
1.20E+0
1.80E+0
2.40E+0
3.00E+0
0.00E+0
3.00E+0
6.00E+0

Appendix 1

SYNTAX FOR SIMULATION LANGUAGE

SYNTAX FOR SIMULATION-LANGUAGE

A1 <letter>:=A/B/C/D/E/F/G/H/I/J/K/L/M/N/O/P/O/R/S/T/U/V/X/Y/Z

A2 <digit>:=0/1/2/3/4/5/6/7/8/9

A3 <null>:=

A4 <name>:=<letter>{<letter>/<digit>}$_0^4$ --- <reserved word>

A5 <variable>:=<name>

A6 <parameter>:=<name>

Reserved word

B1 <index>:={<digit>}$_1^\infty$

B2 <state variable>:=X<index>

B3 <derivative>:=D<state variable>

B4 <insignal>:=U<index>

B5 <outsignal>:=Y<index>

B6 <time>:=T

B7 <own function>:=FCN<index>

B8 <one argument function>:=SIN/COS/EXP/SORT/ATAN/ABS
/ALOG/TANH/AINT

B9 <two argument function>:=ATAN2/AMOD/SIGN

B10 <boolean word>:=OR/AND/NOT/IF/THEN/ELSE

B11 <reserved word>:=<state variable>/<derivative>
/<insignal>/<outsignal>/<time>
/<own function>/<one argument function>
/<two argument function>/<boolean word>

Function designator

C1 <argument>:=<arithmetic expression>

C2 <function designator>:=
<own function>(<argument>)
/<one argument function>(<argument>)
/<two argument function>(<argument>,<argument>)

Number

D1 <unsigned integer>:={<digit>}$_1^\infty$

D2 <integer>:={<null>/+/-}<unsigned integer>

D3 <decimal fraction>:=.<unsigned integer>

D4 <exponent part>:=E<integer>

D5 <decimal number>:=<unsigned integer>{<null>/.}
/<decimal fraction>
/<unsigned integer><decimal fraction>

D6 <unsigned number>:=
<decimal number>{<null>/<exponent part>}

D7 <number>:={<null>/+/-}<unsigned number>

Arithmetic expression

E1 <adding operator>:=+/-

E2 <multiplying operator>:=*//

E3 <identifier>:=<state variable>/<derivative>/<insignal>
/<outsignal>/<time>/<variable>/<parameter>

E4 <primary>:=<unsigned number>/<identifier>
/<function designator>/(<arithmetic expression>)

E5 <factor>:=<primary>{↑<primary>}$_0^\infty$

E6 <term>:=<factor>{<multiplying operator><factor>}$_0^\infty$

E7 <simple arithmetic expression>:=[<null>/<adding operator>]
<term>{<adding operator><term>}$_0^\infty$

E8 <if clause>:=IF<boolean expression>THEN

E9 <arithmetic expression>:=<simple arithmetic expression>
/<if clause><simple arithmetic expression>ELSE
<arithmetic expression>

Boolean expression

F1 <relational operator>:=</=/>

F2 <relation>:=<simple arithmetic expression>
<relational operator><simple arithmetic expression>

F3 <boolean primary>:=<variable>/<parameter>/<relation>
/(<boolean expression>)

F4 <boolean secondary>:=[<null>/NOT]<boolean primary>

F5 <boolean factor>:=<boolean secondary>{AND<boolean secondary>}$_0^\infty$

F6 <boolean term>:=<boolean factor>{OR<boolean factor>}$_0^\infty$

F7 <boolean expression>:=<boolean term>
/<if clause><boolean term>ELSE<boolean expression>

System description

G1 <line terminator>:=<carriage return or alt mode>

G2 <comment>:=[<null>/"<the characters between " and
<line terminator>>]<line terminator>

G3 <X0 definition>:=<state variable>:<number><comment>

G4 <parameter definition>:=<parameter>:<number><comment>

G5 <own function definition>:=
*<own function>{<comment>}$_1^\infty$
{<argument value><function value>{<comment>}$_1^\infty$}$_2^\infty$
*FIN<comment>

G6 <argument value>:=<number>

G7 <function value>:=<number>

G8 <left part>:=<derivative>/<insignal>/<outsignal>/<variable>

G9 <assignment statement>:=<left part>={<arithmetic expression>
/<boolean expression>}<comment>

G10 <system description>:={<X0 definition>
/<parameter definition>/<own function definition>
/<assignment statement>/<comment>}$_0^\infty$
»END<comment>

Appendix 2

ERROR MESSAGES DURING COMPILATION

ERROR-MESSAGES DURING COMPILATION

A. Syntactical errors (references to SYNTAX FOR SIMULATION-LANGUAGE)

0 The first character in <X0 definition>/<parameter definition> /<assignment statement> must be alfabetic (G3,G4,G9)

1 After <state variable> in <X0 definition> must follow : (G3)

2 After : must follow <number> (G3,G4)

3 After <number> in <X0 definition>/<parameter definition> must follow <comment> (G3,G4)

4 After <derivative>/<insignal>/<outsignal> in <left part> must follow = (G9)

5 <name> must not contain more than five characters (A4)

6 After <parameter> in <parameter definition> must follow : and after <variable> in <left part> must follow = (G4,G9)

7 <variable> must not appear in <parameter definition> (G4)

8 <parameter> must not appear as <left part> (G8)

9 After <own function> in <function designator>, <one argument function> and <two argument function> must follow ((C2)

10 . must be preceeded or followed by <unsigned integer> (D3,D5)

11 Allowed delimiters in <arithmetic expression> are
< > = + - * / ↑ ( ) ,

12 Unrecognized character

20 <time>/<own function>/<one argument function>/<two argument function>/<boolean word> must not appear as <left part> (G8)

30 <primary>/<boolean primary> must not be immediately preceeded by <primary>/<boolean primary>

31 NOT must be immediately preceeded by OR/AND/IF/THEN/ELSE/=/(

32 IF must be immediately preceeded by IF/ELSE/=/(

33 OR/AND/</>/=/*///+/THEN/ELSE/)/,/<comment> must be immediately preceeded by <primary>/<boolean primary>
and +/- must be immediately preceeded by <primary>/<boolean primary>/OR/AND/NOT/</>/=/IF/THEN/ELSE/(/,

34 Too few <argument>:s have been recived for actual function (C2)

35 Too many <argument>:s are recived for actual function (C2)

36 Unrecoverable syntax error

40 After beginning * must follow <own function>/FIN/END (or CLOSE) (G5,G10)

41 After <own function> in <own function definition>, <function value>, FIN and END (and CLOSE) must follow <comment> (G5,G10)

42 <own function definition> line must begin with * , <argument value> or <comment> (G5)

43 After <argument value> must follow <function value> (G5)

44 At least two pairs of <argument value><function value> must be given in <own function definition> (G5)

B. Restrictions

50 <unsigned number> contain too many significant digits

51 It is not possible to define more <name>:s

52 Actual <parameter> has already been defined

53 Before a <variable> is used in right-part it must be defined in an earlier <assignment statement>

54 It is not possible to define more <unsigned number>:s

70 The value of <index> must not be zero

71 The value of <index> must not be greater than ten

72 Before a <derivative> is used in right-part it must be defined in an earlier <assignment statement>

73 The <derivative>:s must be consecutivly defined

74 Before an <insignal> is used in right-part it must be defined in an earlier <assignment statement>

75 The <insignal>:s must be consecutivly defined

76 Before an <outsignal> is used in right-part it must be defined in an earlier <assignment statement>

77 The <outsignal>:s must be consecutivly defined

80 It is not possible to use more RPN

90 The value of <index> in <own function definition> must not be zero

91 The value of <index> in <own function definition> must not be greater than ten

92 The <own function>:s must be consecutivly defined

93 <argument value>/<function value> contain too many significant digits

94 The <argument value>:s must increase

95 It is not possible to define more <argument value>/<function value>

96 Not all <parameter>:s have been defined

97 Not all <derivative>:s have been defined

98 Not all <own function>:s have been defined

Appendix 3

SYNTAX FOR COMMANDS

SYNTAX FOR COMMANDS

SYSTT{<null>/<filename>}<comment>

SYSDK<filename><comment>

<filename>:=$\{$<letter>/<digit>$\}_1^5$

DISP<comment>

CX0<X0 definition>

CPAR<parameter definition>

PLOT<identifier><$\{$<identifier>$\}_1^{10}$<comment>

AXIS{<null>/{H/V}<minimum value><maximum value>}<comment>

<minimum value>:=<number>

<maximum value>:=<number>

COMPU$\{$<start value>/,$\}_i^i\{$<stop value>/,$\}_j^j\{$<number of intervals>/,$\}_k^k$
$\{\{$MARK$\}_1^\infty\}_l^l$<comment>

$1 \geq i \geq j \geq k \geq l \geq 0$

SIMU$\{$<start value>/,$\}_i^i\{$<stop value>/,$\}_j^j\{$<number of intervals>/,$\}_k^k$
$\{\{$CONT/MARK/<upper error bound>$\}_1^\infty\}_l^l$<comment>

$1 \geq i \geq j \geq k \geq l \geq 0$

<start value>:=<number>

<stop value>:=<number>

<number of intervals>:=<integer>

<upper error bound>:=<number>

STOP<comment>