Modpac Commands

User's Guide

Wieslander, Johan

1980

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

CODEN: LUTFD/(TFRT-3158)/1-81/(1980)

Modpac Commands

-User´s Guide

JOHAN WIESLANDER

INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA
1980

MODPAC COMMANDS

-

User's Guide

Johan Wieslander

Organization
LUND INSTITUTE OF TECHNOLOGY
Department of Automatic Control
Box 725
S-220 07 Lund 7 SWEDEN

Document name
Project report STU

Date of issue
February 1980

CODEN: LUTFD2/(TFRT-3158)/1-81/(1980)

Author(s)
Johan Wieslander

Sponsoring organization
Swedish Board for Technical Development (STU), contract No 77-3548

Title and subtitle

Modpac Commands - User's Guide

A4 A5

Abstract

Modpac is an interactive program, command oriented with a powerful macro facility. The program is aimed at transformations and analysis of state space models. Analysis is through computation of eigenvalues or frequency responses as well as Kalman decomposition. The available transformations are between continuous time and discrete time forms as well as coordinate changes to obtain diagonal forms, balanced - Hessenberg - and canonical forms.

Key words

A4 A5

Classification system and/or index terms (if any)

Supplementary bibliographical information

Language
English

ISSN and key title

ISBN

Recipient's notes

Number of pages
81

Price

Security classification

Distribution by (name and address)

Modpac Commands - User's Guide

This is the guide to the commands in the program Modpac. It will give information on how to call the different commands, what their functions are and what methods are used. In many cases hints and examples are included.

Modpac is an interactive program package aimed to be an interface between other program packages in the family. This need arises because Idpac e.g. gives as one of its results a system description on transfer function form, while Synpac is aimed at a system description on state space form.

Modpac resolves this conflict by offering a transformation between these two forms of system descriptions. In short, Modpac will serve as a tool in handling different forms of model descriptions. Today Modpac contains the following facilities:

a) Transformation in both directions between continuous time and discrete time state space systems.
b) Transformation in both directions between multi input - single output transfer functions and state space system descriptions.
c) Transformation of state space descriptions to diagonal, balanced and Hessenberg forms.
d) Computation of the frequency response of a state space system representation.
e) Plotting of frequency responses in Bode, Nichols and Nyquist diagrams.
f) Computation of zeroes for scalar polynomials.
g) Definition and handling of polynomials and matrices and evaluation of matrix expressions.

As indicated above, Modpac is intended as a glue between different program packages. It is therefore of course compatible with them, both regarding its user interaction and its data stuctures. Further reading on the interaction and the general use of these programs can be found in the two reports:

1. J. Wieslander, H. Elmqvist: INTRAC - A Communication Module for Interactive Programs - Language Manual, TFRT-7132, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.

2. J. Wieslander: Interactive Prgrams for Analysis and Design of Control Systems - General Guide, TFRT-3156, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.

These programs were developed with financial support from the Swedish Board of Technical Development, contracts 73-3553, 75-3776 and 77-3548. They represent the combined effort of many members of the department over several years.

Commands Available in Modpac

The following is a structured list of commands available in Modpac, together with a short indication of their use.

1. Input & Output

AGR - Edit an aggregate file
EDIT - Edit a symbolic file
LIST - Output data on user readable form
MOVE - Move data in the data base

2. Graphic Output

BODE - Draw curves in a diagram with logarithmic scales
HCOPY - Take a hard copy of the last graphic output
NIC - Display a frequency response in a Nichols diagram
NYQ - Display a frequency response in a Nyquist diagram
PLEV - Display eigenvalues etc in the complex plane

3. Matrix Operations

ALTER - Alter elements in a matrix
ENTER - Enter a matrix element by element
EXPAN - Generate a matrix from sub-matrices
MATOP - Perform matrix operations
REDUC - Extract a submatrix
UNITM - Generate a unit matrix
ZEROM - Generate a zero matrix

4. Polynomial Operations

POCONV - Polynomial image - polynomial file conversion
POLY - Generate or edit a polynomial
POLZ - Compute the zeroes of a polynomial
ZERPOL - Create a polynomial from its zeroes

5. System Operations

CONT - Convert to continuous time form
KALD - Do a Kalman decomposition
SAMP - Convert to discrete time form
SPSS - Compute the frequency response
SSTRF1 - Convert from state space to transfer function
SYST - Generate a system description
SYSTR - Do a general coordinate transformation
TBALAN - Transform to balanced form
TCON - Transform to controllable form
TDIAG - Transform to diagonal form
THESS - Transform to Hessenberg form

TOBS - Transform to obsevable form
TRFSS1 - Convert from transfer function to state space

6. Miscellaneous

DELET - Delet a file from the data base
FHEAD - Inspect and change file parameters
FTEST - Test the existence of a file
TURN - Change program switches

7. Alphabetical Command List

AGR

Purpose

To create or to edit an aggregate file.

Command

```
AGR AGROUT
   or
AGR [AGROUT] < AGRIN
```

```
AGROUT  - name of resulting aggregate file
          by default AGROUT = AGRIN
AGRIN   - name of original aggregate file
```

Subcommands

The sub-commands implicitly use a pointer to the current component file.

LOOK [NAME]
: Display the table of contents of AGROUT.
If NAME is present, then only the entries named NAME, if any, will be displayed. The pointer is not affected.
For each entry, one may see if it is flagged for insertion, deletion, and/or isolation.

KILL
: Leave sub-command mode. Current AGR-command including sub-commands will have no effect.

X
: Leave sub-command mode. Current AGR-command including sub-commands will take effect.

LOC NAME
: Make the pointer point at the component file NAME.
The scan takes place between the current pointer location plus one and the last entry, at which place the pointer remains in case of no success.

REP [NAME]
: Replace the component file at the current pointer location by the individual file NAME.
The pointer is not affected.
By default NAME equals the name of the component file at the current pointer location.

DEL
Delete the component file at the current pointer location, then auto-decrement the pointer.

ISO
Short for isolate, i.e. copy the component file at the current pointer location to an individual file with the same name. The pointer is not affected.

TOP
Make the pointer point above the first component file.

INS NAME
Insert the individual file NAME after the current pointer location, then auto-increment the pointer.

BOT
Make the pointer point at the last component file.

REM
Remove the REP, DEL and ISO flags from the current pointer location. The pointer is not affected.

ADV [NR]
Advance the pointer NR steps. By default NR = 1. In case of no success, the pointer will remain at the last entry for a positive NR and at location 0 for a negative NR.

Function

The main command specifies the type of operation, i.e. update or generate an aggregate file. The component files are manipulated by subcommands in a way similar to that of a line oriented text editor. Note that no I/O operations (other than file existence tests) take place until the excute subcommand (X) is entered; the specified operations are only entered into a table. This table may be viewed through the subcommand LOOK, and errors may be corrected. At the time of execution, the operations are performed from the top of the table, thus the chronological order in which the operations were entered is immaterial.

EDIT

Purpose

To edit, i.e. create or make changes to a symbolic (text) file. Examples are MACRO-files, system files and symbolic data files from outside.

Command

EDIT TFILE

TFILE	name of symbolic (text) file

Subcommands

The following general notation is used:

n	denotes a positive integer, default 1.
/	denotes any character not included in 'string'.
string	denotes any sequence of printing characters including space.

A string	the string is appended to the current line.
B	the bottom line of the file is made the new current line.
C /string1/string2/	string1 in the current line is changed to string2.
D [n]	n lines are deleted starting with the current line.
E	exit, i.e. close the file and return.
F string	find the first line after the current line starting with string and make it current.
I string	insert string as the new current line after the now current line.
L string	locate the first line after the current line containing string and make it current.
N [n]	make the n:th next line current.

O [n]	overlay the n next lines including the current with keyboard INPUT.
P [n]	print n lines starting with the current line. The last line printed is the new current line.
R string	replace the current line with string.
T	go to the top of the file.
DIS ON DIS OFF	enable/disable output on display.

Function

The editor works in one of two modes, EDIT-mode and INPUT-mode. In EDIT-mode, the editor will read the text-file line by line. At any time, one line is the 'current line'. The subcommands control the position of the 'current line' within the text-file, or modify the 'current line'.

In INPUT-mode, a line typed on the keyboard is made the new 'current line', thus forcing the old one to be written to the output file.

The initial mode of the editor is INPUT if the specified file is not found, otherwise EDIT. An empty line is used to switch the mode.

Cautions, Restrictions

EDIT does not allow subcommands or input from a macro.

Hints

System files are normally generated by the command SYST. Exception: the polynomial image system files in Idpac.

LIST

Purpose

To output a file on lineprinter, teleprinter, or display.

Command

```
LIST [(DEV)][(FEED)][(DMODE)][AGGREG:]FNAME[(A1 A2..)][IF NUM]

DEV      - device = 'DIS'/'LP'/'TP'
             DIS    - display
             LP     - line printer
             TP     - teleprinter
             (by default DEV = 'DIS')
FEED     - form feed parameter = 'FF'/'LF'
             FF     - a form feed will precede output
             LF     - a line feed will precede output
             (by default FEED = 'LF')
DMODE    - data mode indicator = 'D'/'T'/'DS'/'TS'/'FT'/'FTS'
             D      - FNAME is assumed to contain binary data
             T      - FNAME is assumed to contain text
             DS     - same as 'D', but sequence numbers written
             TS     - same as 'T', but sequence numbers written
                      and the text will be truncated after
                      72 characters
             FT     - same as 'T' but 'BEGIN', 'END' not written
                      provided that section names
                      have been given explicitly
             FTS    - same as 'FT' with sequence numbers
             (by default DMODE = 'D')
AGGREG   - aggregate file, invalid in connection with
           DMODE = 'T'
FNAME    - name of file to be listed
A..      - attributes associated with FNAME, if
           DMODE = 'D'/'DS', then A.. denotes column numbers,
           otherwise names of sections within FNAME
IF       - number of 1st record to be listed
           (valid only in connection with DMODE = 'D'/'DS')
NUM      - number of records to be output
           (valid only in connection with DMODE = 'D'/'DS')
```

Function

The data is printed as matrix blocks with NUM lines containing the first few columns, a blank line, NUM lines containing the next few columns etc.

Note, frequency response files are special cases of data files.

Text files: The file is directly copied onto the output medium. Text files are:

a) any file created or manipulated by the EDIT command,
b) MACRO files,
c) system files,
d) structure files.

If a section name is given for a system file, only that section is output.

Cautions, Restrictions

The available output devices may be installation dependent.

Hints

The mechanism that allows listing of selected sections of a system file uses the keywords BEGIN and END. These keywords may be used to produce the same effect in any text file, e.g. to output descriptive text from a macro.

Examples

```
>LIST DATA
>LIST (LP) DATA(3 4 6) 20 10
>LIST (T) MAC
>LIST (LP)(T) SYST(NAME)
```

MOVE

Purpose

To move files in the data base. The columns of a data file may be rearranged.

Command

```
MOVE [(DMODE)] [[AGOUT:]FILOUT[(C11..)]] <
                                  [AGIN:]FILIN[(C21..)]

DMODE   - data mode = 'D'/'T'/'ND' (default: DMODE = 'D')
          D     - the file is assumed to contain binary data
          T     - the file is assumed to contain text
          ND    - same as 'D' but the columns C11.. will,
                  if previously defined, not be overwritten,
                  but placed rightmost in FILOUT, in the
AGOUT   - output aggregate file
FILOUT..- output file name [with column numbers]
AGIN    - input aggregate file
FILIN.. - input file name [with column numbers]
```

Function

The columns C21,.. in the data file FILIN are moved to the columns C11,.. in the data file FILOUT.

Copying is the only function available for symbolic (text) files or for files within aggregates.

Cautions: Restrictions

- Column numbers cannot be used for system- and macro-files.
- Data files may contain up to 20 columns as input files and up to 15 columns as output files.

Examples

```
>MOVE WORK < DATA(2 5 3)
>MOVE (ND) WORK(1 3) < DK DATA(4 1)
```

The results are shown below.

Input & Output
MOVE

DATA

11.0000	21.0000	31.0000	41.0000	51.0000
12.0000	22.0000	32.0000	42.0000	52.0000
13.0000	23.0000	33.0000	43.0000	53.0000
14.0000	24.0000	34.0000	44.0000	54.0000
15.0000	25.0000	35.0000	45.0000	55.0000
16.0000	26.0000	36.0000	46.0000	56.0000
17.0000	27.0000	37.0000	47.0000	57.0000
18.0000	28.0000	38.0000	48.0000	58.0000
19.0000	29.0000	39.0000	49.0000	59.0000
20.0000	30.0000	40.0000	50.0000	60.0000

WORK

21.0000	51.0000	31.0000
22.0000	52.0000	32.0000
23.0000	53.0000	33.0000
24.0000	54.0000	34.0000
25.0000	55.0000	35.0000
26.0000	56.0000	36.0000
27.0000	57.0000	37.0000
28.0000	58.0000	38.0000
29.0000	59.0000	39.0000
30.0000	60.0000	40.0000

WORK

41.0000	21.0000	11.0000	51.0000	31.0000
42.0000	22.0000	12.0000	52.0000	32.0000
43.0000	23.0000	13.0000	53.0000	33.0000
44.0000	24.0000	14.0000	54.0000	34.0000
45.0000	25.0000	15.0000	55.0000	35.0000
46.0000	26.0000	16.0000	56.0000	36.0000
47.0000	27.0000	17.0000	57.0000	37.0000
48.0000	28.0000	18.0000	58.0000	38.0000
49.0000	29.0000	19.0000	59.0000	39.0000
50.0000	30.0000	20.0000	60.0000	40.0000

BODE

Purpose

To plot frequency response files in Bode diagram format.

Command

BODE [(SW)] FRF1[(F11 F12 ..)] [FRF2[(F21 ..)] ..

SW — page switch = 'A'/'P'/'AP'/'AO'
(default: 'AP')
A : plot amplitude only,
then read a sub-command
P : plot phase only, then exit
AP: plot amplitude and phase together,
then exit
AO: plot amplitude only, then exit

FRF.. — frequency response file name(s)
F11.. — frequency response number(s) (default all)

Subcommands

PAGE — request the phase plot (relevant only if SW = 'A')
KILL — skips the phase plot (relevant only if amplitude and phase are to be plotted separately)

Function

The indicated (default all) curves of the frequency response file(s) FRF1 etc., are plotted versus frequency. The abscissa is a logarithmic axis, while the ordinate is logarithmic for the amplitude and linear for the phase. If the phase information is identically zero, as for auto spectra, the phase plot is omitted.

Amplitude values smaller than 1.E-5*(largest value) are replaced by the lower limit.

If more than one set of curves are requested, they are marked with integers representing the order of the corresponding response in the command.

Hints

a) Cf. the command PLOT for methods to include text in the diagram.

b) The command treats frequency response files, see the general guide. Generally, they include frequency information scaled in rad/s. If you want a bode plot in Hz, use the SCLOP command to divide the frequency by 2π = 6.2831853.

Example (cf. the commands NIC and NYQ)

The Bode plot for the system

$$G_D = \frac{3.25\ s + 1}{s^2\ (s + 1.75)}$$

is given through the command (the response is contained in the file FRF):

```
>BODE FRF
```

See Figure 1.

The Bode plot for the system

$$G_c = \frac{3.25\ s + 1}{s^3 + 1.75\ s^2 + 3.25\ s + 1}$$

is shown in Figure 2 (the response was contained in FRF2).

Graphic Output
BODE

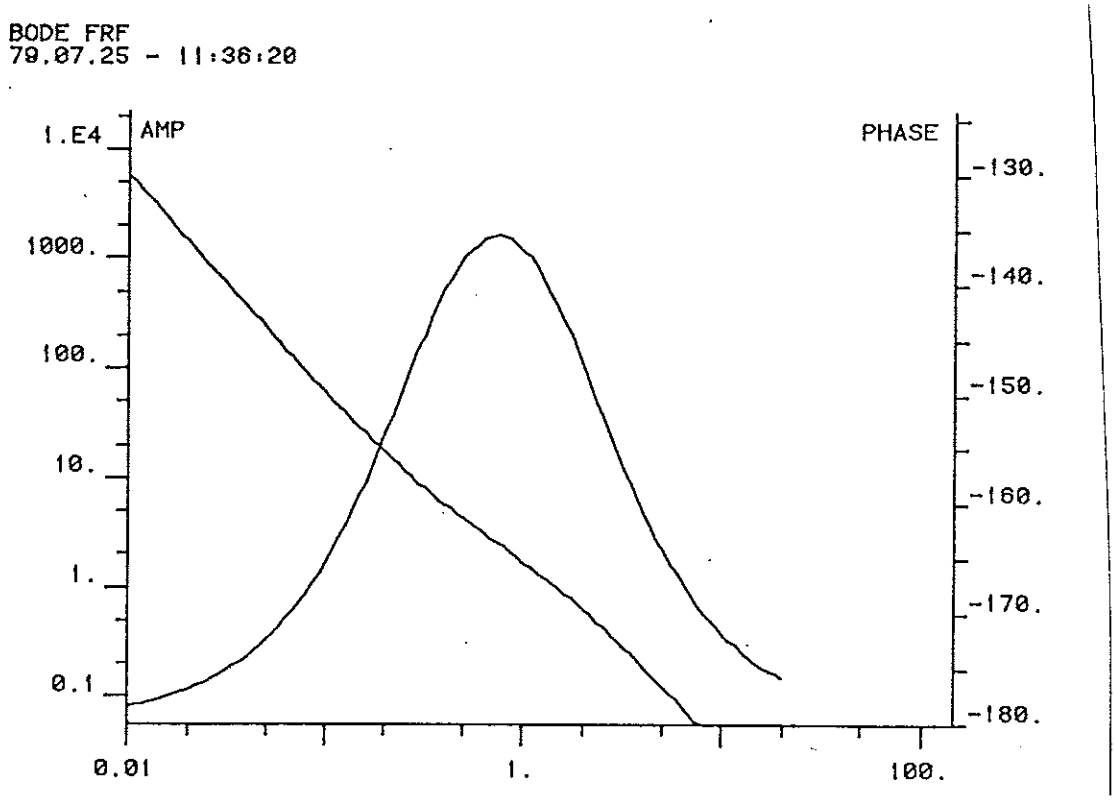

Figure 1. The amplitude and phase of G_O.

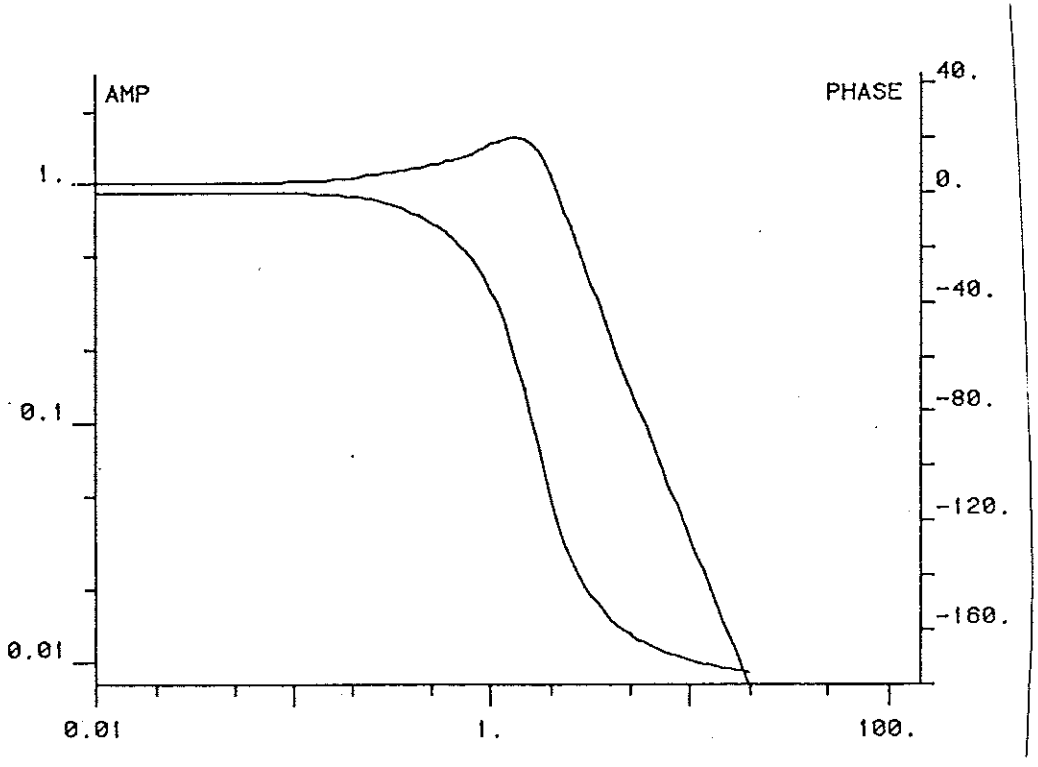

Figure 2. The amplitude and phase of G_C.

HCOPY

Purpose

To generate a hard copy of graphical output.

Command

```
HCOPY [DEV] [FACTOR]
   or
HCOPY SWITCH
```

```
DEV      - hardcopy device = 'L'/'R' (default: 'L')
           L: local hardcopy (Tektronix 4662)
           R: remote hardcopy (Calcomp 1051)
FACTOR   - scale factor (default: 1.)
           For L: .5 < FACTOR < 1.6
           For R: .5 < FACTOR < 4.
SWITCH   - = 'ON'/'OFF'/'T'
           ON : enables hardcopy
           OFF: disables hardcopy
           T  : output a free text string at the current
                joystick location;applies to TEKTRONIX 4662
                only, the textstring in the command line
                being preceded by a double quote
```

Function

After that the command HCOPY has been used with the switch ON; all graphical output that is generated in any command is also saved temporarily. A subsequent use of the command HCOPY will cause the saved information from the last such command to be sent to the selected hard copy device.

Hints

Note that HCOPY actually is available as a subcommand for all graphic generating command.

NIC

Purpose

To plot frequency response files in a Nichols diagram.

Command

NIC [WMIN WMAX] FRF1[F11 ..)] [FRF2...]

WMIN, WMAX- frequency limits, default: all frequencies plotted
FRF.. - frequency response filename(s)
F11.. - frequency response number(s), default: all

Note: max 5 curves may be displayed

Function

The indicated (default all) curves of the frequency file(s) FRF1 etc., are plotted in a rectilinear coordinate system, where the horizontal axis is linear and represents the phase in degrees and the vertical represents the magnitude and is logarithmic. Frequency points of the form $1*10^n$, $2*10^n$, and $5*10^n$ (n integer) are indicated on the curve(s).

If a plotted curve is the frequency response of an open loop transfer function G_O, the corresponding closed loop transfer function $G_c = G_O/1+G_O$ can be read from the curvilinear coordinate system. These curves represent the magnitude of the closed loop transfer function in dB and its phase in degrees.

If more than one curve are requested, they are marked with integers according to the order of the response in the command.

Cautions, Restrictions

There can be no more than five curves in a single diagram. Often only a part of a frequency response will fit into the diagram. Then the optional arguments WMIN and WMAX are recommended.

Graphic Output
NIC

Hints

Cf. the commands BODE and NYQ.

Example

(Cf. the commands BODE and NYQ.) The frequency response for $0.2 \leq \omega \leq 5$ is shown in a Nichols diagram:

```
>NIC 0.2 5 FRF
```

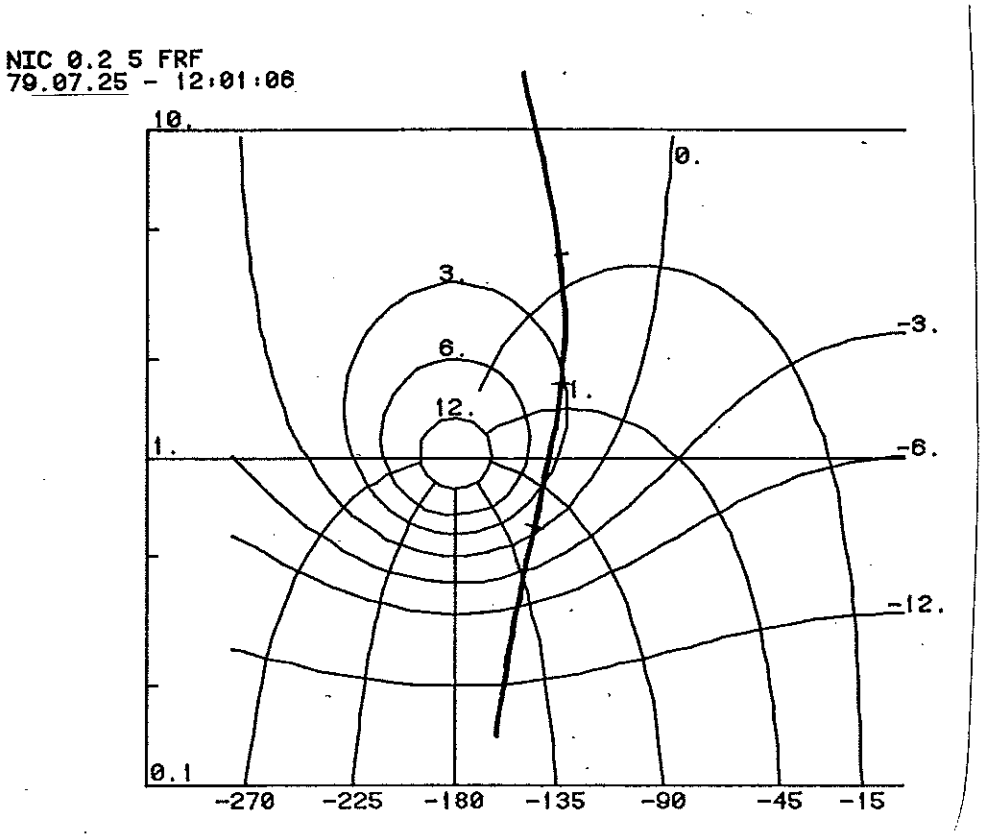

Figure 1

NYQ

Purpose

To plot frequency response files in a Nyquist diagram.

Command

NYQ [WMIN WMAX] FRF1[(F11 ..)] [FRF2 ...]

WMIN, WMAX - frequency limits, default: all frequencies plotted
FRF..   - frequency response filename(s)
F11..   - frequency response number(s), default: all

Function

The indicated (default all) frequency responses are plotted. For each frequency its associated value, being a complex number in polar form, is marked in the complex plane and a line is drawn to the preceding point, so forming a Nyquist curve.

Frequency points of the form $1*10^n$, $2*10^n$, and $5*10^n$ (n integer) are indicated on the curve(s).

Cautions, Restrictions

No more than five curves can be plotted in a single diagram. It may be essential to use the arguments WMIN and WMAX in order to obtain a reasonable scale on the axes.

Hints

Cf. the commands BODE and NIC.

Example

(Cf. the commands BODE and NIC.) The Nyquist plot of the open loop system

$$G_0 = \frac{3.25\ s + 1}{s^2\ (s + 1.75)}$$

looks like (see Figure 1):

Graphic Output
NYQ

>NYQ 0.5 10 FRF

The corresponding closed loop system

$$G_c = \frac{3.25\ s + 1}{s^3 + 1.75\ s^2 + 3.25\ s + 1}$$

looks like (its response is in FRF2):

>NYQ 0.2 5 FRF2

See Figure 2.

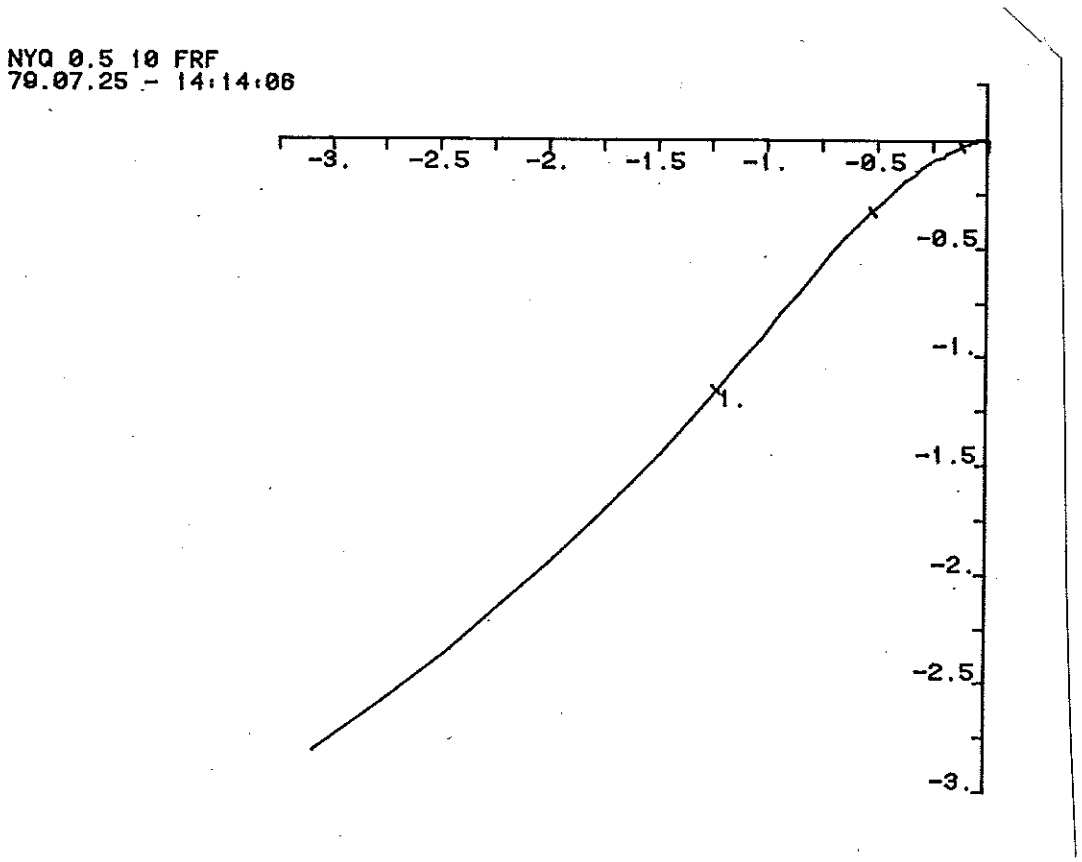

Figure 1. The Nyquist plot of G_0.

Graphic Output
NYQ

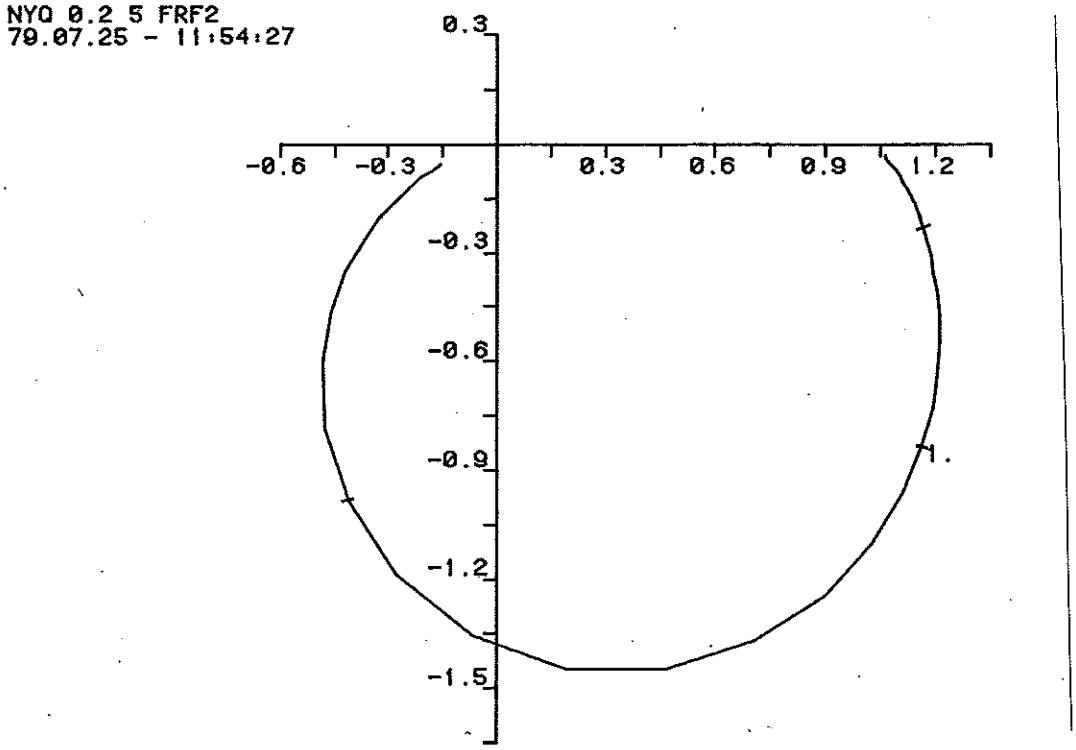

Figure 2. The Nyquist plot of G_c.

PLEV

Purpose

To plot and optionally edit the contents of a locus file, e.g. eigenvalues, poles and zeros.

Command

```
PLEV FNAM2 [FNAM3 FNAM4 ...]
  or
PLEV  FNAM1 < FNAM2
```

FNAM1 - locus file containing new eigenvalues
FNAM2,FNAM3 ... - locus filescontaining original eigenvalues

Subcommands

ALT N VR [VI]

Alter eigenvalue number N: real part to VR and imaginary part to VI (default = 0.). If the former imaginary part of eigenvalue number N is not equal to zero, the complex conjugate is also altered, to VR and -VI. It is not allowed to alter a single real eigenvalue to a complex. In that case, use the second form:

ALT N1 VR VI & N2
Alter the two real eigenvalues number N1 and N2 to complex conjugated eigenvalues. The real parts are altered to VR and the imaginary parts to +VI and -VI respectively.

SCALE N V
Scale eigenvalue number N and its omplex conjugate, if there is any, by the scaling-factor V.

DAMP N Z
Move eigenvalue number N and its complex conjugate along a circle to achieve the relative damping Z, 0 < Z < 1.

EXAM N
Write the real and imaginary parts of eigenvalue number N. If the imaginary part is not equal to zero, the relative damping, the distance to the origin and the angle to the negative real axis are also written.

LOOK
Plot eigenvalues on display and write the numerical values if there is room for them on the screen. The eigenvalues are ordered after decreasing values of the real parts.

Graphic Output
PLEV

X
Ends the cubcommand sequence and generates output.

KILL
Aborts the subcommand sequence.

Function

The information is plotted in a complex plane shown on the terminal. A unit circle is included for discrete time locus files. The command exists in two forms. The first form serves to visualize the information from one or more locus files but does not allow changing it.

The second form allows only one locus file as input, but generates a new one as output with changes made through subcommands.

The operations through the subcommands automatically change complex conjugates and a special form exists to change two reals to a complex conjugate pair. The eigenvalues are numbered in order of decreasing real parts. They are also listed in this order on the terminal (if TEXT is ON).

To identify different sets of eigenvalues (several original sets or an old and a new one), the following sequence of symbols are used: X, □, ▽, , 5, 6,... .

Examples

Assume that the locus file LOCUS is given, containing 3 eigenvalues all equal to zero. The command

```
>PLEV LOCUSN < LOCUS
```

will display the original values, see Figure 1. The subommands

```
>ALT  1  -1
>ALT  2  -0.5  0.5  &  3
>LOOK
```

will then give the output shown in Figure 2.

```
>SCALE  1  1.5
>LOOK
```

then results in Figure 3. Finally the relative damping is specified to 0.9 giving Figure 4:

```
>DAMP  1  0.9
>LOOK
>X
```

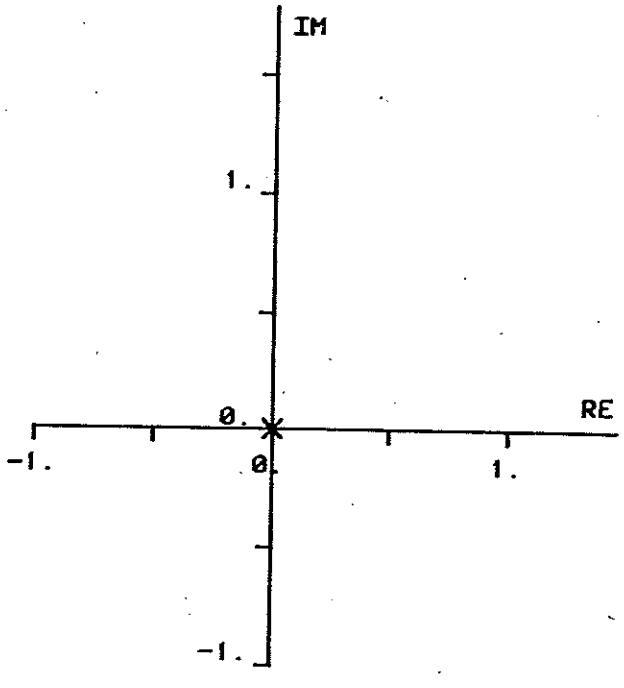

Figure 1. The original set of eigenvalues.

Graphic Output
PLEV

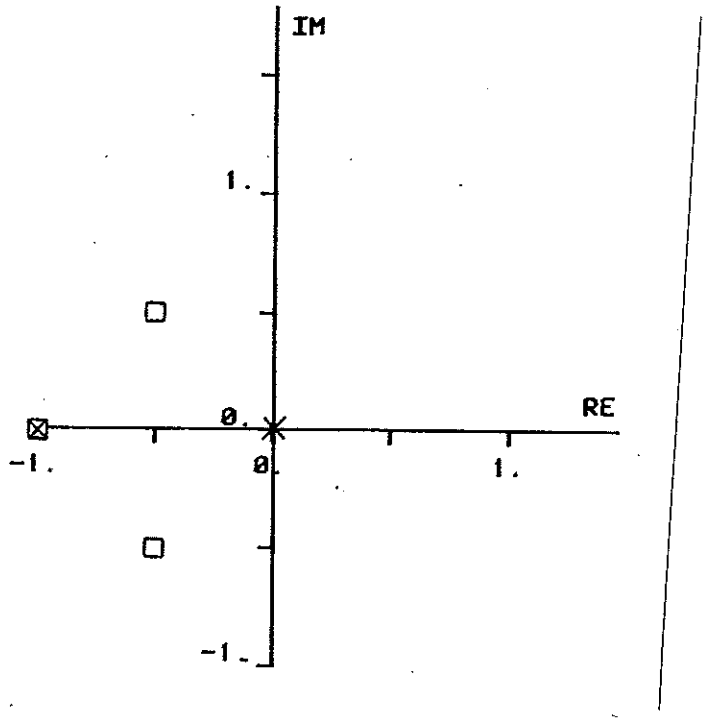

Figure 2. The eigenvalues after the first change.

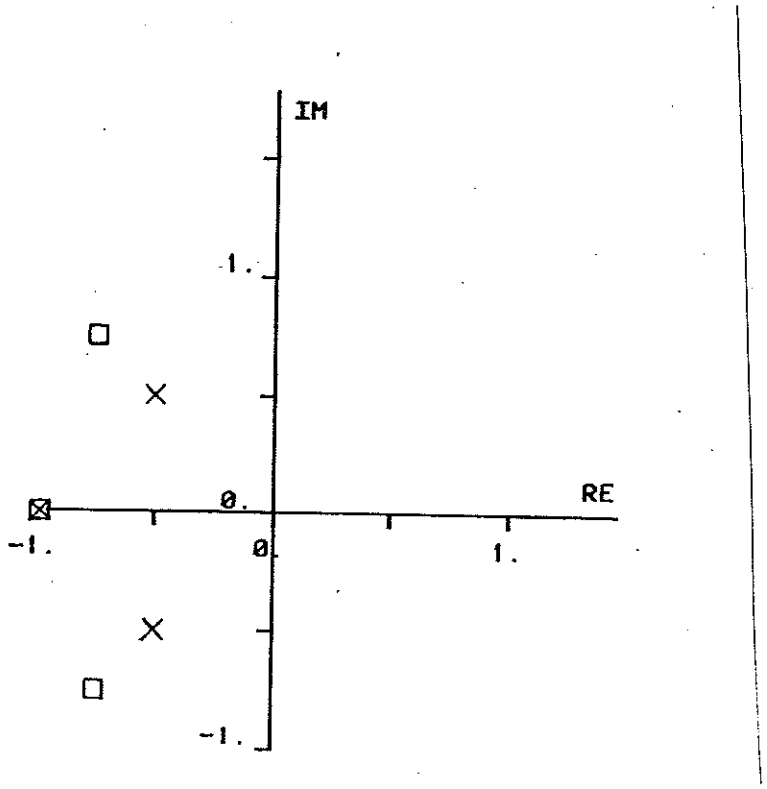

Figure 3. Moving a complex pair with constant damping.

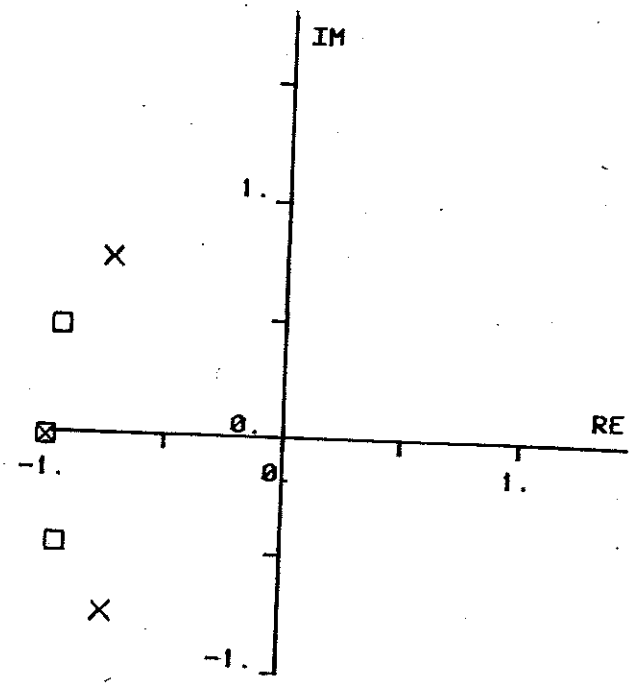

Figure 4. The final eigenvalue positions.

ALTER

Purpose

To alter the elements of a matrix.

Command

ALTER [AGGREG:] MATRIX [(IR IC) VALUE]

AGGREG - aggregeate file name
MATRIX - matrix file name
IR - row index
IC - column index
VALUE - new value

Subcommands

KILL - resumes main command mode; MATRIX is not updated
X - resumes main command mode; MATRIX is updated
IR IC VALUE - change element (IR;IC) to VALUE

Function

This command has two formats. If the matrix is specified together with row and column index and the new value in the same line; the entire operation is performed in one step; i.e. the matrix is read and written back to the data base. If only the matrix is specified; row and column indices and new values are expected as subcommands. Not until the execute command (X) is received; the matrix is written to the data base.

Cautions; Restrictions

If more than one matrix element is to be altered; the subcommand form is much more efficient.

ENTER

Purpose

To enter a matrix, element by element.

Command

ENTER [AG:]MAT NR [NC] [TSAMP]

AG	- name of aggregate file
MAT	- name of matrix file
NR	- number of rows
NC	- number of columns (default NR)
TSAMP	- sample interval (default DELTA.)

Subcommands

KILL	- resume main command mode, no action taken
X	- resume main command mode, MAT is updated/generated

Function

After the command line has been entered, the values of the matrix elements are asked for. They may be entered in free format on one or several lines. An error message is generated if too many values are given. The subcommand KILL may be given if the user wants to abort the operation. It may be used in place of any element value, but it must appear at the beginning of a line. The subcommand X is legal only when all element values have been entered.

Cautions, Restrictions

Do not forget the specification of a sampling interval for a matrix to be part of a discrete time state space description. Cf. DELTA.

Examples

To enter a square (2x2) matrix with sample interval 3 s.

```
>ENTER  A  2  3.
ROW 1
    #> 1 2
ROW 2
    #> 3 4
    #> X
>LIST  A
```

```
Matrix Operations
ENTER


    LIST  A

    1.000     2.000
    3.000     4.000
```

To enter a 2x3 matrix and changing one's mind after the 2 2 element:

```
>ENTER  B  2  3
ROW 1
    #> 1 2
    #> 3
ROW 2
    #> 4 5
    #> KILL
>
```

EXPAN

Purpose

To generate a matrix made up of blocks of other matrices.

Command

```
EXPAN [[AG1:]M1] < [AG2:]M2[(IX2 IY2)] [[AG3:]M3[(..)]..]
```

```
AG       - aggregate file name
M        - matrix file name
IXi,IYi  - the coordinates in the new matrix for
           the upper left corner of matrix Mi
           (by default IX=IY=1)
```

Function

A new matrix is generated from one or more old matrices. Undefined elements are zeroed. The position of the old matrices within the new one is specified by means of the location of its upper left corner (the 1 1 element) in the new matrix. The dimension of the new matrix is determined by the maximum row and column position occupied by any lower right corner of the old matrices.

Example

Let M1 be a 3x3 matrix of threes and M2 a 2x2 matrix of twos. Then

```
>EXPAN  X1 < M2 (1  2)
```

yields

```
     X1:  0  2  2
          0  2  2
```

```
>EXPAN  X2 < M1 (1  1)  M2 (2 1)
```

yields

```
     X2:  3  3  3
          2  2  3
          2  2  3
```

MATOP

Purpose

To evaluate matrix expressions.

Command

MATOP [(EXT)] [[AGGREG:]MATRIX] < algebraic matrix expression

EXT - name extension for global variables
AGGREG - aggregate file name
MATRIX - name of resulting matrix file (by default MATRIX = the 1st matrix name in the algebraic expression)

Function

The right hand side matrix expression is evaluated according to the rules stated below. The value is assigned to the matrix given in the left hand side (possibly a component of an existing aggregate), or if no left hand side was given, to the first matrix specified in the right hand side.

If EXT was included in the command line, those of the global variables DET.EXT, MINMAX.EXT, and TRACE.EXT that exist as real variables will receive the determinant, the minmax norm, and the trace of the result, respectively.

Method

The right hand side expression is evaluated from left to right following standard precedence rules. Parentheses are allowed. Below a formal definition of the allowed expression follows. Some very simple production rules are omitted.

<adding operator>::= +/-

<multiplying operator>::= *

<function operator>::= TR/PSINV/^<signed integer>

<matrix reference>::= <aggregate name>:<matrix name>/
<matrix name>

<scalar>::= <variable reference>/<real constant>

<primary>::= <matrix reference>/<scalar>
<multiplying operator><matrix reference>

<factor>::= <primary>/<primary><function operator>/
(<expression>)

<term>::=<factor>/<term><multiplying operator><factor>

<expression>::= <term>/<adding operator><term>/
<expression><adding operator><term>

Examples

Some simple expressions follow.

2*A
A+B
A*B+C
A^-1
0.5*(A TR + A)

UNITM

Purpose

To generate a unit matrix.

Commands

UNITM [* FACTOR] [AG:]MAT NR [TSAMP]

FACTOR - scale factor (by default FACTOR = 1)
AG - name of aggregate file
MAT - name of matrix file
NR - number of rows
NC - number of columns (default NR)
TSAMP - sample interval (by default TSAMP = DELTA.)

Function

A unit matrix, optionally scaled according to the scale factor given, is generated.

Examples

>UNITM A 2

results in

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

>UNITM * -1 A 3

results in

$$A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

REDUC

Purpose

To pick out a block from an existing matrix.

Command

```
REDUC [[AG1:]M1] < [AG2:]M2 (IX1 IY1 IX2 IY2)
```

AG	- aggregate file name
M	- matrix file name
IX1,IY1	- indices for the upper left corner of the part to be saved
IX2,IY2	- indices for the lower right corner of the part to be saved

Function

A new matrix is generated as a block of an old one. The block is specified in terms of its upper left and lower right corner.

Example

The matrix X2 (cf. EXPAN) is given

```
     X2:  3  3  3
          2  2  3
          2  2  3

>REDUC  X3 < X2 (2  2  3  3)
```

yields

```
     X3:  2  3
          2  3
```

ZEROM

Purpose

To generate a zero matrix.

Commands

ZEROM [+ TERM] [AG:]MAT NR [NC] [TSAMP]

TERM	- constant term (by default TERM = 0)
AG	- name of aggregate file
MAT	- name of matrix file
NR	- number of rows
NC	- number of columns (default NR)
TSAMP	- sample interval (by default TSAMP = DELTA.)

Function

A zero matrix of the specified dimension is generated. Optionally, a specified constant may be added to all elements.

Examples

```
>ZEROM   B  2
```

results in

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

```
>ZEROM  +  -1  B  1  4
```

results in

```
     B = [-1  -1  -1  -1]
```

POCONV

Purpose

To convert MISO Transfer Function models from polynomial image to polynomial file form and vice versa.

Command

```
POCONV [SYSOUT][(NAMOUT)] < SYSIN[(NAMIN)]

SYSOUT  - name of output system file,
          by default SYSOUT = SYSIN
NAMOUT  - name of section within SYSOUT
SYSIN   - name of input system file
NAMIN   - name of section within SYSIN

     or

POCONV POFILE < SYSIN[(NAMIN)] POTYPE [NR]

POFILE  - resulting polynomial file
SYSIN   - see above
NAMIN   - see above
POTYPE  - type of input polynomial image
          = 'A'/'B'/'C'/'D'/'I',   'I' being short for
          'initial'
NR      - polynomial number, if omitted, then all the
          polynomial images of type POTYPE will be converted
```

Function

The polynomial image form is the form of the MISO Transfer Function description found in IDPAC. Here the polynomial coefficients are included in the system file, while in the polynomial file form, coefficients are stored in files. This command allows conversion between the two formats.

When the polynomial image system description is given and appears on the right hand side of the argument string, the action is to produce an appropriate set of polynomial files. In the first form of the command, these are specified in the left hand side system description, while in the second command form, a single polynomial is explicitly referenced.

When the right hand side system description is of polynomial file form, a complete new system description is generated.

POLY

Purpose

To create or update a polynomial with scalar or matrix coefficients.

Command

```
POLY POLOUT [NR NC] [TSAMP]
   or
POLY [[AGOUT:]POLOUT] < [AGIN:]POLIN [NR NC] [TSAMP]
```

AGOUT - name of output aggregate file
POLOUT - name of output polynomial file
AGIN - name of input aggregate file
POLIN - name of input polynomial file
NR - number of rows in a coefficient matrix
(default: 1)
NC - number of columns in a coefficient matrix
(default: 1)
TSAMP - sample interval
by default TSAMP = the reserved variable DELTA.

Subcommands

LOOK [DEG]
display the coefficient of degree DEG
DEG is not affected

KILL
leave sub-command mode, current POLY-command
including sub-commands will have no effect

X
leave sub-command mode, current POLY-command
including sub-commands will take effect

INS [DEG]
insert a coefficient matrix of degree DEG + 1
DEG is auto-incremented.
The contents of the coefficient matrix is
specified by means of sub-commands, see below.

INS [DEG] < VALUE
in this case sub-commands are not expected,
since the coefficient then is assigned VALUE

ALT VALUE [DEG] [NR NC]
alter the (NR,NC):th value of
of the coefficient matrix of degree DEG
to VALUE, by default (NR,NC) = (1,1)

DEG is not affected

ADDZ VRE [VIM]
add the zeroes VRE +- i*VIM to a scalar polynomial, or, in other words, introduce the factor: (Z - (VRE + i*VIM)) * (Z - (VRE - i*VIM)) or, in the absence of VIM, merely: (Z - VRE), where Z denotes the indepent variable

MULC V
multiply the entire polynomial by the constant V

DIVC V
divide the entire polynomial by the constant V

DEL [DEG]
deletes the coefficient matrix of degree DEG DEG is auto-decremented

Sub-commands to INS

NR NC VALUE
defines the value of the (NR,NC):th element in the coefficient matrix of degree DEG

KILL
discards INS and leaves DEG unaffected

X
INS takes effect.
Elements not explicitly given are assumed zero.

Function

In the first form, a polynomial is created and the main command specifies the dimensions and type (continuous or discrete time) of the polynomial.

In the second form, an existing polynomial is edited, possibly giving a new one as result.

The polynomial coefficients (in the general case matrices) are accessed through subcommands. The degree of the coefficient operated upon is given by the pointer DEG which may or may not be given explicitly in the subcommand. DEG is incremented and decremented by INS and DEL respectively to allow an easy succession of such operations.

Note that scalar polynomials may be expanded by factors.

Note also that discrete time polynomials are expressed within POLY as a polynomial in q^{-1}, i.e. the coefficient of the term $c_i q^{-i}$ is accessed with DEG = i. (Still, in the

polynomial file; the polynomial is treated as a polynomial in z, starting with the high order coefficient.)

Hints

Scalar polynomials may also be generated through the command ZERPOL.

Examples

To generate the polynomial

$$AC(s) = s^3 + 6s^2 + 11s + 6$$

```
>POLY  AC  0.
    >INS -1 < 6
    >INS < 11
    >INS < 6
    >INS < 1
    >X
```

To generate the polynomial

$$AD(q^{-1}) = 1 - 2.46439\ q^{-1} + 2.01767\ q^{-2} - 0.548812\ q^{-3}$$

```
>POLY  AD  0.1
    >INS -1 < 1
    >INS < -2.46439
    >INS <  2.01767
    >INS < -0.548812
    >X
```

POLZ

Purpose

To compute the zeros of a polynomial with real, scalar coefficients.

Command

```
POLZ [[AGOUT:]ZERFIL <] SYSIN[(NAMIN)] POTYPE [NR] [EPS]
    or
POLZ [[AGOUT:]ZERFIL <] [AGIN:POLY [EPS]
```

AGOUT	- name of output aggregate file
ZERFIL	- locus file receiving the zeroes
SYSIN	- name of system file
NAMIN	- name of section within SYSIN of the type Miso Transfer Function
POTYPE	- polynomial type = 'A'/'B'/'C'/'D'/'I'
NR	- polynomial number by default NR = 1
EPS	- relative test quantity used to distinguish non-zero coefficients, by default EPS = the reserved variable REPS.
AGIN	- name of input aggregate file
POLY	- name of polynomial file

Function

The command reads the polynomial, either in the first form from a polynomial image system description, or in the second form from a polynomial file. The result is output to a locus file if specified and, if the switch TEXT is ON, to the terminal. If the switch GRAPH is ON, the zero locations are visualized in the complex plane. The unit circle is included in the discrete time case.

Method

The zeros are found using an iterative function minimizing technique based on the Laguerre iteration formula.

If the polynomial has a multiple zero, i.e. it is of the form

$$P(z) = (z-\alpha)^m P_1(z),$$

an ordinary iterative method may give large errors for that zero. This is because the error ε in the computed zero may

be relatively large although the factor ε^m makes the value of the polynomial small.

To avoid this effect, the algorithm also monitors the size of the derivatives, in the case of a multiple zero being of the form:

$$P'(z) = (z-\alpha)^{m-1} P_2(z)$$

$$P''(z) = (z-\alpha)^{m-2} P_3(z)$$

etc.

If the derivatives tend to be small indicating the possibility of a multiple zero, the algorithm tries to find a zero of the first, second etc. derivative.

After a zero has been found, the polynomial is deflated.

Hints

a) The zeros in the locus file produced by POLZ may be viewed and altered in PLEV.

b) ZERPOL is the converse operation to POLZ.

ZERPOL

Purpose

To generate a polynomial with scalar coefficients from a specification of its zeroes.

Command

ZERPOL [AGOUT:]POLOUT < LOCIN [K]

AGOUT - aggregate name for output
POLOUT - resulting polynomial
LOCIN - input locus file
K - gain selector in the locus file
(default: the first row is used)

Function

The information in the input locus file is interpreted as zeroes of a polynomial. This polynomial is computed and output as the result.

The argument K is used to select the desired locations of the poilynomial zeroes in the case when the locus filecontains more than one set of zeroes. The default is to take the first set.

Hints

This command serves as the converse of POLZ. Together with PLEV, they form an easy way of altering a polynomial to achieve certain characteristics.

CONT

Purpose

To transform a system to Continuous State Space Form from Discrete State Space Form.

Command

CONT [SYSOUT][(NAMOUT)] < SYSIN[(NAMIN)] [EPS]

SYSOUT - name of system file for output system
by default SYSOUT = SYSIN
NAMOUT - name of section within SYSOUT
SYSIN - name of system file for input system
NAMIN - name of section within SYSIN
EPS - test quantity
by default EPS = the reserved variable REPS.

Function

The matrices of the input system is read and transformed to their continuous time values. Any structure given, e.g. division of inputs into control inputs and disturbance inputs, is maintained.

Method

We have the following notations in the continuous and discrete time cases:

$$\dot{x} = A_C x + B_C u \qquad x_{t+T} = A_D x_t + B_D u_t$$

$$y = C_C x \qquad y_t = C_D x_t$$

where A_D and B_D are computed from A_C and B_C as (cf. the command SAMP):

$$A_D = e^{A_C T} \tag{1}$$

$$B_D = \int_0^T e^{A_C s}\, ds\, B_C \tag{2}$$

System Operations
CONT

Conversely we have:

$$A_C = \frac{1}{T} \ln A_D \tag{3}$$

$$B_C = (A_D - I)^{-1} A_C B_D. \tag{4}$$

The logarithm in (3) is computed as:

$$Y = A_D^{1/2^n}$$

where the integer n is chosen so that $\|Y-I\| < 1$.

$$X = A_D^{1/2}$$

is found by solving $X^2-A_D = 0$ by an iterative technique. Then by solving $e^Z-Y = 0$ also by an iterative technique A_C is found:

$$Z = \ln Y$$

$$A_C = \frac{2^n}{T} \ln A_D^{1/2^n} = \frac{1}{T} \cdot 2^n \cdot Z = \frac{1}{T} \ln A_D.$$

In evaluating (4) directly, a problem would be encountered for systems containing integrators since A_D-I would be singular. Therefore (4) is evaluated as follows:

$$B_C = (A_D-I)^{-1} A_C B_D = (e^{A_C T}-I)^{-1} A_C B_D =$$

$$= (A_C T \, \Psi(A_C T))^{-1} A_C B_D = \frac{1}{T} \Psi^{-1}(A_C T) B_D,$$

System Operations
CONT

where $\Psi(A_C T)$ is defined by the series expansion

$$\Psi(A_C T) = I + \frac{1}{2!} A_C T + \frac{1}{3!} A_C^2 T^2 + \frac{1}{4!} A_C^3 T^3 + \ldots$$

KALD

Purpose

To perform a Kalman decomposition of a dynamic system, i.e. to decompose the system equations in terms of controllability and observability.

KALD SNAME[(NAME)] [AEPS REPS]

SNAME - name of system description file for original system
NAME - name of section within SNAME
AEPS - absolute test quantity, by default AEPS = the reserved variable AEPS.
REPS - relative test quantity, by default REPS = the reserved variable REPS.

Subcommands

SAVE RNAME[(NAME)] < RESLT [ATTR1 [ATTR2]]
saves a specified piece of information

RNAME - name of output system/matrix
NAME - name of section within RNAME (valid only if RNAME represents a system)
RESLT - mnemonic for data set to be saved
RESLT = 'SYS'/'AMAT'/'BMAT'/'CMAT'/'TMAT'
SYS - save system
AMAT - save A-matrix
BMAT - save B-matrix
CMAT - save C-matrix
TMAT - save transformation-matrix
ATTR1 - data attribute, invalid in connection with 'TMAT'
ATTR1 = 'TOT'/'C'/'NC'/'O'/'NO'/1/2/3/4
TOT - save the total data set
C - save the controllable modes
NC - save the non-controllable modes
O - save the observable modes
NO - save the non-observable modes
1/2/3/4 - save the couplings between the various subsystem (valid together with AMAT only)
ATTR2 - data attribute, invalid with RESLT = 'TMAT' and ATTR1 = 'TOT', but compulsory if ATTR1 = 1/2/3/4
ATTR2 = 'C'/'NC'/'O'/'NO'/1/2/3/4
Note: the attributes 'C'/'NC' may only be combined with 'O'/'NO'

LOOK - show the decomposition schematically

X - end the subcommand-sequence

Function

After the main command has been entered, the specified system is decomposed into its

(1) controllable and observable part
(2) controllable but non-observable part
(3) non-controllable but observable part
(4) non-controllable and non-observable part.

The existence of these four different parts of the system is indicated on the terminal together with the respective number of states. Through the subcommand SAVE, the user has then the possibility to extract selected parts of the information viz.

a) a complete subsystem, e.g. the controllable and observable part.

b) the transformation matrix, resulting in the decomposed system description.

c) a specified block of the transformed A, B, or C matrices.

Method

The essence of the method is the forming of basis vectors for the controllable and observable modes and their orthogonal complements from the columns of $[B \quad AB \quad A^2B \quad \ldots]$ and

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

Cautions: Restrictions

The result of the method depends sometimes on the choice of the test quantities found in the global variables AEPS. and REPS.

Examples

A system is given that is known to contain modes of all four categories above (controllable and observable .. etc.). The following commands will then result in the terminal output on Figure 1.

```
>KALD  S(CSS)
  >LOOK
```

If e.g. non-controllable but observable modes existed, the third row and column would not be included in Figure 1.

The transformation matrix would be found by the subcommand:

```
    >SAVE  T  < TMAT
    >...
    .
    .                    "Other subcommands
    .
    >
    >X
>LIST  T                 "In order to have a look at T
```

To obtain the controllable and observable part of the system, a new system description can be generated (the system file must have been defined previously):

```
    >SAVE COSYS < SYS  C  0
```

Various blocks from the A, B, and C matrices may be saved, if their existence is indicated in the terminal output, cf. Figure 1.

```
    >SAVE  A23  <  AMAT  2  3
    >SAVE  B2   <  BMAT  2
```

System Operations
KALD

```
KALD S(CSS)
79.09.19 - 16:40:49

      A11                 A13                   B1

      A21       A22       A23       A24         B2

                          A33

                          A43       A44


      C1                  C3


    5 STATE(S),       1 INPUT(S),       1 OUTPUT(S)


    2       CONTROLLABLE AND       OBSERVABLE STATE(S) (A11)
    1       CONTROLLABLE AND NON-OBSERVABLE STATE(S) (A22)
    1 NON-CONTROLLABLE AND       OBSERVABLE STATE(S) (A33)
    1 NON-CONTROLLABLE AND NON-OBSERVABLE STATE(S) (A44)
```

Figure 1. The table output on the terminal.

SAMP

Purpose

To compute the discrete time (i.e. sampled) form of a continuous time state space system description.

Command

SAMP [SYSOUT][(NAMOUT)] < SYSIN[(NAMIN)]

SYSOUT - name of system file for output system
by default SYSOUT = SYSIN
NAMOUT - name of section within SYSOUT
SYSIN - name of system file for input system
NAMIN - name of section within SYSIN

Function

The matrices of the input system description are read. Any structural information, i.e. presence of input/output matrix blocks, is noted and used in the output of the result.

The system matrices are then transformed to a discrete time description according to the sample interval found in the output system description.

Method

The input system matrices are blocked together to a system description of the form:

$$\begin{cases} \dot{x} = A_C x + B_C u \\ y = C_C x + D_C u \end{cases}$$

These equations are then transformed into

$$x_{t+T} = A_D x_t + B_D u_t$$
$$y_t = C_D X_t + D_D u_t$$

where

$$A_D = e^{A_C T} = A_C T \, \Psi(A_C T) + I$$

$$B_D = \int_0^T e^{A_C s}\, ds\, B_C = \Psi(A_C T)\, B_C\, T$$

$$\Psi(A_C T) = I + \frac{1}{2!} A_C T + \frac{1}{3!} A_C^2 T^2 + \ldots$$

$$C_D = C_C$$

$$D_D = D_C$$

(cf. the command CONT). $\Psi(A_C T)$ is computed from its series expansion.

Reference

C. Kallstrom: Evaluation of e^A and $\int e^{As} ds$. Report TFRT-3053, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.

SPSS

Purpose

To compute the frequency characteristics (power spectrum or amplitude and phase) between one input and one output for a system on state space form.

Command

SPSS [('POW'/'AMP')] FRF[(F)] < SYST[(NAME)] NY NU [FREQ]

'POW'/'AMP'	- switch choosing a power spectrum or an amplitude and phase computation default is 'AMP'
FRF	- frequency response file
F	- frequency response number (default value 1)
SYST	- system file name
NAME	- section name of system file
NY	- output (measurement) signal number
NU	- input (control) signal number
FREQ	- file with frequency values in the first column

Function

The frequency response between input NU and output NY is computed for the system description specified. The system description may be on continuous time or discrete time form.

The frequency points are logarithmically distributed between the two values WMIN. and WMAX. (two reserved global variables); or if the file FREQ is present, the frequency points are taken from its first column.

Method

In the continuous time case, the evaluation of the frequency response amounts to computing the matrix $H(j\omega)$ for different values of ω.

$$H(j\omega) = C(j\omega - A)^{-1} B + D.$$

To avoid computation with complex matrices we reformulate the expression to

$$H(j\omega) = - C(\omega^2 I + A^2)^{-1} (j\omega I + A) B + D.$$

Here, the real and imaginary parts may be computed separately.

In the discrete time case, the formulas look like:

$$H(z) = C(zI-A)^{-1}B + D; \quad z = e^{j\omega T} = \cos \omega T + j \sin \omega T.$$

The real and imaginary parts are:

$$\text{Re } H(e^{j\omega T}) = C(A^2 + I - 2\cos(\omega T)A)^{-1}(I\cos(\omega T)-A)\,B + D$$

$$\text{Im } H(e^{j\omega T}) = -C(A^2 + I - 2\cos(\omega T)A)^{-1}\sin(\omega T)\,B.$$

Knowing the real and imaginary parts, it is a simple matter to compute the power spectrum or the amplitude and phase information. The phase is expressed in degrees and an effort is made to make it continuous across the $\pm 180^{\circ}$ boundaries.

SSTRF1

Purpose

To transform a state space model with a single output to transfer function form.

Command

SSTRF1 [SYSOUT][(NAMOUT)] < SYSIN[(NAMIN)]

SYSOUT - name of system file for output system
by default SYSOUT = SYSIN
NAMOUT - name of section within SYSOUT
SYSIN - name of system file for input system
NAMIN - name of section within SYSIN

Function

The given system is read and converted to a MISO transfer function description (on polynomial file form). The global variable REPS. is used in the test of observability in the conversion algorithm. The system may be on either discrete or continuous time form.

Method

The state space system is transformed to its observable canonical form by the Faddeev-Leverriere algorithm. The polynomial coefficients are then found by inspection.

Cautions, Restrictions

The system must be completely observable.

SYST

Purpose

To aid in the generation of system description files.

Command

```
SYST [(SUBSW)] SYSNAM[(SECNAM)] [< [(SYSTYP)] [SYSMNEM]
     [DT] [AGRNAM] [(TIMTYP)/OP/LAMVAL] [ATRNAM]]

SUBSW   - subcommand switch = 'SC'/'VOID' (default: 'VOID')
          SC     - subcommands wanted
          VOID   - no subcommands wanted
SYSNAM  - name of resulting system file
SECNAM  - see subcommand BEGIN
SYSTYP  - system type = 'SS'/'MTF'/'PM' (default: 'SS')
          SS     - State Space
          MTF    - Miso Transfer Function
          PM     - Polynomial Matrix
SYSMNEM - system mnemonic, a short form used to specify
          the system equation
          SS : SYSMNEM = 'ABC'/'ABCD'/'ABCXO'/'ABCDXO',
               where A, B, C etc. denote system matrices
               (default: 'ABC')
          MTF: SYSMNEM = 'AB'/'ABC'/'ABCI' (default: 'AB')
          PM : SYSMNEM = 'TUUV'/'TUUVWU' (default: 'TUUV')
DT      - see subcommand TSAMP (default DELTA.)
AGRNAM  - see subcommand AG
TIMTYP  - see subcommand TIME
OP      - see subcommand SHIFT
LAMVAL  - see subcommand LAMBDA
ATRNAM  - see subcommand AT
```

Subcommands

```
lambda lamval
BEGIN [SECNAM]  "defines a section name
   SECNAM       - section name (default: missing)

TSAMP DT   "defines a sample interval
   DT      - sample interval expressed in seconds

LOOK   "displays the current contents of SYSNAM[(SECNAM)]

AG [(AGTYP)] [AGRNAM]   "declares an aggregate file
   AGTYP   - aggregate type = 'S'/'L'/'C'/'E' (default: 'S')
             S  - system equation, the only valid one for
                  SYSTYPs differing from 'SS'
             L  - loss function
             C  - covariance function
             E  - extended loss function
   AGRNAM  - aggregate file name (default: SECNAM (SYSNAM if
```

```
                SECNAM is omitted) for the main command and
                AGTYP = 'S', otherwise missing)

AT [ATRNAM]     "declares an attribute file name, valid
                "only if SYSTYP = 'PM'
   ATRNAM       - attribute file name (default: missing)

TIME TIMTYP     "defines whether a system is time variable or
                "not, valid only if SYSTYP = 'SS'
   TIMTYP       - time switch = 'VAR'/'INV' (default: 'INV')
                  VAR  - time variable
                  INV  - time invariant

LAMBDA LAMVAL   "defines lambda, i.e. the noise standard
                "deviation,valid only if SYSTYP = 'MTF' and
                "if SYSMNEM contains the letter 'C'
    LAMVAL      - the value of lambda (default: 1, if SYSMNEM
                  contains the letter 'C', else missing)

SHIFT OP    "defines the shift operator,
            "valid only if SYSTYP = 'PM'
    OP      - operator = '+'/'-' (default: '+')
              +   - forward shift
              -   - backward shift

INS MNEM [< NAME] "inserts terms into the system equations
    MNEM           - matrix or polynomial mnemonic, see below
    NAME           - name of the matrix or polynomial
                     corresponding to MNEM (default: MNEM)

DEL MNEM    "removes terms from the system equations
    MNEM    - matrix or polynomial mnemonic, see below

KILL    "skips the previous subcommand sequence including
        "the last SYST command, i.e. inhibits the generation
        "of SYSNAM[(SECNAM)]

X       "closes SYSNAM[(SECNAM)] with the current contents,
        "then exits from SYST
```

Function

The command generates a new system description file or, if the specified output file already exists, a new section within it.

The command uses a series of defaults such that many of the desires encountered in normal use will be possible to satisfy with a single main level command. If more freedom is required, a series of subcommands is available to specify these details.

The subcommands serve to control the inclusion/deletion of various items in the system description and to allow the user to select names for aggregates and polynomials or matrices. Deletion is by specifying a blank name for e.g. an aggregate, or through the DEL command.

Note that the subcommand LOOK will at any time display the current form of the system description.

The different matrices and polynomials are referenced through mnemonics shown below. The mnemonics are also the default names of the respective items.

The main command argument SYSMNEM serves to choose between standard sets of these matrices/polynomials. These standard sets are also given below.

Matrix mnemonics (SYSTYP = 'SS'):

```
A       B       BW      BV      "DX/DT = .., or XNEW = ..
C       D       DW      DE      "    Y = ..
G       H       HW              "    Z = ..
XO                              "initial state vector

QO      Q1      Q12     Q2      "loss function

RO      R1      R12     R2      "covariance function

EQO     EQ1     EQ12    EQ2     "extended loss function
EQ3     EQ4     EQ5
```

For the different possibilities of SYSMNEM, the following is included:

```
ABC         A   B   C
ABCD        A   B   C   D
ABCXO       A   B   C   XO
ABCDXO      A   B   C   D   XO
```

Polynomial mnemonics (SYSTYP = 'MTF'):

```
A     B     C     D     "Ay = Bu + Ce
I                       "initial output values
```

For the different possibilities of SYSMNEM, the following is included:

```
AB          A   B
ABC         A   B   C
ABCI        A   B   C   I
```

Polynomial mnemonics (SYSTYP) = 'PM':

T	UU	UW	UV	"..*X = ..
V	WU	WW	WE	" Y = ..
G	HU	HW		" Z = ..

For the different possibilities of SYSMNEM, the following is included:

TUUV	T	UU	V	
TUUVWV	T	UU	V	WV

Examples

The following serves to illustrate the operation of SYST. The global variable DELTA. = 1.0 during these examples.

a) The two commands

```
>SYST S(CSS) < 0.
>LIST (T) S
```

produces:

```
LIST (T) S
79.10.05 - 16:46:08

BEGIN CSS

"SYST S(CSS)<0.
"79.10.05 - 16:45:53
"
CONTINUOUS STATE SPACE REPRESENTATION

DYNAMICS, AGGREGATE: CSS,

DX/DT = A*X + B*U

Y = C*X

END
```

while

b)
```
>SYST S2(DSS) < ABCD S2DSS
>LIST (T) S2
```

produces

```
LIST(T)S2
79.10.05 - 16:47:16

BEGIN DSS

"SYST S2(DSS)<ABCD S2DSS
"79.10.05 - 16:47:06
"
DISCRETE STATE SPACE REPRESENTATION

SAMPLE INTERVAL 1. S

DYNAMICS, AGGREGATE: S2DSS,

XNEW = A*X + B*U

Y = C*X + D*U

END
```

Note in case b) that the default value of the sample interval takes effect producing a discrete time state space representation, and that in both a) and b) the state space form is assumed. In b) a fourth system matrix (D) is included and the dynamics aggregate name is changed from its default, viz. the section name.

c) Here we use the subcommand switch:

```
>SYST (SC) S3 (S3CSS)
    >LOOK
```

The subcommand LOOK gives the present form of the system description.

```
"SYST(SC)S3(S3CSS)
"79.10.05 - 16:48:46

BEGIN S3CSS

DISCRETE STATE SPACE REPRESENTATION

SAMPLE INTERVAL 1. S

DYNAMICS, AGGREGATE: S3CSS,

XNEW = A*X + B*U

Y = C*X

END
```

d) Assuming we did not want a discrete time representation and intending to use a linear quadratic loss function, we do:

```
>TSAMP 0.
>AG (L) CLS3
>INS Q1
>INS Q2
>LOOK
```

We have now obtained the following:

```
"SYST(SC)S3(S3CSS)
"79.10.05 - 16:50:21

BEGIN S3CSS

CONTINUOUS STATE SPACE REPRESENTATION

DYNAMICS, AGGREGATE: S3CSS,

DX/DT = A*X + B*U

Y = C*X

LOSS FUNCTION, AGGREGATE: CLS3,

Q1: Q1, Q2: Q2

END
```

If this is what we wanted, we finish the subcommand sequence by:

```
>X
```

SYSTR

Purpose

To introduce a specified change of state space coordinates, viz. $\xi = Tx$, where ξ and x are the new and old coordinates.

Command

```
SYSTR  [SYST1][(NAME1)] < SYST2[(NAME2)] T [EPS]
```

```
SYST1 - system description file name for the transformed
        system
        by default SYST1 = SYST2
NAME1 - section name of the transformed system, SYST1
SYST2 - system description file name for the original
        system
NAME2 - section name of the original system, SYST2
T     - name of transformation matrix
EPS   - test quantity for matrix inversion
        by default EPS = the reserved varible REPS.
```

Function

The given system matrices are read, together with the desired transformation matrix T. If T is not singular, the transformed system matrices are then output according to the resulting system description. If the two system descriptions contain blocked matrices, their structure must be compatible.

Method

If the given system is described by

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

the new system equation is given by:

$$\begin{cases} \dot{\xi} = TAT^{-1}\xi + TBu \\ y = CT^{-1}\xi + Du \end{cases}$$

If the system is on blocked form, the B and C matrices are first constructed from their components. After the transformation, the results are again blocked with the same structure as before.

Cautions; Restrictions

In the inversion of T (actually solving a system of equations), the global variable REPS. is used as test quantity, if EPS is absent from the command arguments.

TBALAN

Purpose

To determine a coordinate transformation $\xi = Tx$ for a system on state space form such that the A-matrix of the transformed system is balanced with nearly minimum norm.

Command

```
TBALAN  T [ [SYST1][(NAME1)] ] < SYST2[(NAME2)] [EPS]
```

T - name of transformation matrix
SYST1 - system description file name for the transformed system
NAME1 - section name of the new system, SYST1
SYST2 - system description file name for the original system
NAME2 - section name of the original system, SYST2
EPS - test quantity
by default EPS=the reserved varible REPS.

Function

The transformation matrix T and optionally the transformed system are computed from the given system. The given system may be blocked in which case the resulting system will have the same structure.

Method

The operation takes place in two steps. First the algorithm tries to isolate eigenvalues of A. This is done by successively searching for rows or columns where all off-diagonal elements are zero. Such rows or columns are moved to the bottom or to the left by a series of permutations. The remaining part of the matrix is scaled by a diagonal transformation matrix to achieve a lowest possible norm of A. The scaling is subject to the constraint that scale factors be powers of 2, whereby (on a binary computer) no loss of numeric precision is introduced.

References

B.N. Parlett, C. Reinsch: Balancing a Matrix for Calculation of Eigenvalues and Eigenvectors. In J.H. Wilkinson, C. Reinsch: Linear Algebra, Springer-Verlag, 1971.

Hints

The balanced form will in many cases prove useful in heavy numeric calculation.

Example

A system S with the A-matrix below was defined. The command:

```
>TBALAN T < S
```

gives the matrix T shown below. The new A-matrix is also given.

The original A-matrix:

6	0	0	0	0	1	0
0	4	0	0.0003	0.01	0.02	0.1
1	100	7	0	0	-2	20
0	20000	0	1	-400	300	-4000
-2	-300	0	0.01	2	2	40
0	0	0	0	0	0	0
0	10	0	0.004	0.1	-0.2	3

The transformation matrix T:

0	0	1	0	0	0	0
0	1024	0	0	0	0	0
0	0	0	0	0	0	64
0	0	0	0.0625	0	0	0
0	0	0	0	8	0	0
1	0	0	0	0	0	0
0	0	0	0	0	1	0

System Operations
TBALAN

Using T, the transformed (i.e. balanced) A-matrix is:

7	0.09765625	0.3125	0	0	1	-2
0	4	1.6	4.9152	1.28	0	20.48
0	0.625	3	4.096	0.8	0	-12.8
0	1.2207	-3.90625	1	-3.125	0	18.75
0	-2.34375	5	1.28	2	-16	16
0	0	0	0	0	6	1
0	0	0	0	0	0	0

Note that the eigenvalues 7, 6, and 0 have been isolated.

Cautions: Restrictions

The system must have a single input. The system must be controllable.

Example

The system (A,B,C) with the following matrices

$$\dot{x} = \begin{bmatrix} -3 & 0 & 0 & 0 \\ 1 & -2 & -2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.5 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x$$

is transformed to

$$\dot{z} = \begin{bmatrix} -5.5 & -10.5 & -100 & -3 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} z + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x$$

Compare this with the example in TOBS.

TCON

Purpose

To determine a coordinate transformation $\xi = Tx$ for a system on state space form such that the transformed system is on controllable canonical form.

Command

```
TCON  T [ [SYST1][(NAME1)] ] < SYST2[(NAME2)] [EPS]
```

```
T     - name of transformation matrix
SYST1 - system description file name for the transformed
        system
        by default SYST1 = SYST2 if NAME1 is present
NAME1 - section name of the transformed system, SYST1
SYST2 - system description file name for the original
        system
NAME2 - section name of the original system, SYST2
EPS   - test quantity
        by default EPS = the reserved varible REPS.
```

Function

The transformation matrix T and optionally the transformed system are computed from the given system. The outputs of the given system may be blocked in which case the resulting system will have the same structure. The system must have a single input.

Method

The coefficients a_i of the characteristic equation are found using the Faddeev-Leverrier algorithm. Using this information, the transformation matrix is constructed recursively from

$$t_1 = B$$

$$t_{i+1} = At_i + a_iB$$

where t_i are the columns of T^{-1}. (A and B are matrices in the given system description.)

TDIAG

Purpose

To compute a coordinate transformation $\xi = Tx$ for a system on state space form such that the A-matrix of the transformed system is diagonal (for complex eigenvalues, block-diagonal).

NB. The output matrix is not T but rather T^{-1}, being the matrix of eigenvectors.

Command

```
TDIAG  EIGVEC [ [SYST1][(NAME1)] ] < SYST2[(NAME2)] [EPS]

EIGVEC- name of eigenvector matrix
        (=name of inverted transformation matrix )
SYST1 - system description file name for the transformed
        system
        by default SYST1 = SYST2 if NAME1 is present
NAME1 - section name of the transformed system, SYST1
SYST2 - system description file name for the original
        system
NAME2 - section name of the original system, SYST2
EPS   - test quantity
        by default EPS = the reserved variable REPS.
```

Function

The inverse of the transformation matrix T and optionally the transformed system are computed from the given system. The given system may be blocked, in which case the resulting system will have the same structure.

Note that the inverse of the transformation matrix, called EIGVEG in the argument list, contains the eigenvectors of the original A-matrix as columns. Cf. below.

Method

The transformation matrix is determined in three steps. First a transformation is determined that balances the given A-matrix, cf. the command TBALAN. This is done to improve numeric precision since it is known that errors in the eigenvalues usually are proportional to the norm of the matrix. The second step consists of a transformation to upper Hessenberg form, cf. the command THESS. Finally, the matrix is transformed using the QR method to triangular form where eigenvalues and eigenvectors are readily available.

To avoid complex numbers, the triangulation is allowed to leave 2x2 blocks on the diagonal representing a pair of complex conjugated eigenvalues.

The eigenvectors are output in the columns of the matrix EIGVEC, while the eigenvalues are found in the diagonalized A-matrix. To a real eigenvalue corresponds a real eigenvector. For a complex eigenvalue, the real part of the eigenvector is stored in the first corresponding column while the imaginary part of that eigenvector is stored in the second corresponding column. The second eigenvector corresponding to that complex conjugated pair is not output but is known to be the complex conjugate of the first one.

The representation of eigenvectors used has the advantage that if this eigenvector matrix is used as T^{-1} in the similarity transformation $D = TAT^{-1}$, the matrix D is diagonal except for complex conjugate eigenvalue pairs $x \pm iy$ which are represented as diagonal 2x2 blocks of the form:

$$\begin{matrix} x & y \\ -y & x \end{matrix}$$

Thus D is a real matrix. Cf. the example below.

References

G. Peters, J.H. Wilkinson: Eigenvectors of Real and Complex Matrices by LR and QR Triangularisations. In J.H. Wilkinson, C. Reinsch: Linear Algebra, Springer-Verlag, 1971.

Example

A system S with the following A-matrix is defined.

```
-3    0    0    0
 1   -2   -2    0
 0    1    0    0
 0    0    1  -0.5
```

The command

```
>TDIAG E < S
```

gives the following E as result

```
 0     0     0     1
 0     1     0  -0.6
 0  -0.5  -0.5   0.2
 1  -0.2   0.6  -0.08
```

Using this matrix in a similarity transformation $\xi = E^{-1}x$ either in TDIAG or through

```
>MATOP D < E^-1 * A * E
```

gives the following diagonal matrix D

-0.5	0	0	0
0	-1	1	0
0	-1	-1	0
0	0	0	-3

Note the eigenvalues: -0.5, -1±i, and -3.

THESS

Purpose

To compute a coordinate transformation $\xi = Tx$ for a system on state space form such that the A-matrix of the transformed system has upper Hesserberg form.

Command

THESS T [[SYST1][(NAME1)]] < SYST2[(NAME2)] [EPS]

T - name of transformation matrix
SYST1 - system description file name for the transformed system
by default SYST1 = SYST2 if NAME1 is present
NAME1 - section name of the transformed system, SYST1
SYST2 - system description file name for the original system
NAME2 - section name of the original system, SYST2
EPS - test quantity
by default EPS=the reserved varible REPS.

Function

The transformation matrix T and optionally the transformed system are computed from the given system. The given system may be blocked, in which case the resulting system will have the same structure.

Method

First, a transformation T_1 is determined that balances the A-matrix, cf. the command TBALAN. Then a transformation T_2 is determined consisting of a series of stabilized elementary similarity transformations designed to make all elements $a_{ij} = 0$, $i \geq j + 2$. Finally, $T = T_2 T_1$.

References

R.S. Martin, J.H. Wilkinson: Similarity Reduction of a General Matrix to Hessenberg Form. In J.H.Wilkinson, C. Reinsch: Linear Algebra, Springer-Verlag, 1971.

Hints

The Hessenberg form will in many cases prove useful in heavy numeric calculations.

TOBS

Purpose

To determine a coordinate transformation $\xi = Tx$ for a system on state space form such that the transformed system is on observable canonical form.

Command

```
TOBS  T [ [SYST1][(NAME1)] ] < SYST2[(NAME2)] [EPS]
```

```
T     - name of transformation matrix
SYST1 - system description file name for the transformed
        system
        by default SYST1 = SYST2 if NAME1 is present
NAME1 - section name of the transformed system, SYST1
SYST2 - system description file name for the original
        system
NAME2 - section name of the original system, SYST2
EPS   - test quantity
        by default EPS = the reserved varible REPS.
```

Function

The transformation matrix T and optionally the transformed system are computed from the given system. The inputs of the given system may be blocked in which case the resulting system will have the same structure. The system must have a single output.

Method

The coefficients a_i of the characteristic equation are found using the Faddeev-Leverrier algorithm. Using this information, the transformation matrix is constructed recursively from

$$t_1 = C$$

$$t_{i+1} = t_i A + a_i C$$

where t_i are the rows of T. (A and C are matrices in the given system description.)

Cautions; Restrictions

The system must have a single output. The system must be observable.

Example

The system (A;B;C) with the following matrices

$$\dot{x} = \begin{bmatrix} -3 & 0 & 0 & 0 \\ 1 & -2 & -2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.5 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x$$

is transformed to

$$\dot{z} = \begin{bmatrix} -5.5 & 1 & 0 & 0 \\ -10.5 & 0 & 1 & 0 \\ -10 & 0 & 0 & 1 \\ -3 & 0 & 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} z$$

Note that the coefficients of the transfer function are available by inspection in the transformed A and B matrices, cf. the command SSTRF1. The corresponding transfer function is

$$G(s) = \frac{1}{s^4 + 5.5\, s^3 + 10.5\, s^2 + 10\, s + 3}$$

TRFSS1

Purpose

To transform a multi input single output transfer function to state space observable canonical form.

Command

TRFSS1 [SYSOUT][(NAMOUT)] < SYSIN[(NAMIN)]

SYSOUT - name of system file for output system
by default SYSOUT = SYSIN
NAMOUT - name of section within SYSOUT
SYSIN - name of system file for input system
NAMIN - name of section within SYSIN

Function

The given MISO transfer function is represented on state space form. The global variable REPS. is used to test for a non-zero high order coefficient in the given polynomials.

Method

The state space respresentation is found through directly introducing the polynomial coefficients in an observable canonical form.

Cautions, Restrictions

The method is restricted to systems with a single output.

DELET

Purpose

Deletes files from the data base.

Command

```
DELET [(DMODE1)]FNAM1 [ [[(DMODE2)]FNAM2] ... ]

DMODE    - data mode indicator = 'D'/'T'/'A'
           D      - FNAM is assumed to contain binary data
           T      - FNAM is assumed to contain text
           A      - FNAM is assumed to be an aggregate file
(DMODE='D' by default)
FNAM     - file name
```

Function

If the specified file exists in the data base it is deleted, otherwise an error message is given.

Hints

The existence of the file may be tested by the command FTEST.

Example

Cf. the macro in the example in RANPA.

FHEAD

Purpose

To display file head parameters of data files and enable the user to change them.

Command

```
FHEAD [AGGREG:]FILE

AGGREG  - aggregate file name
FILE    - file name
```

Subcommands

```
INDEX VALUE
          set the INDEX:th parameter to VALUE
LOOK[K] - display the K:th (default all) file head
          parameter(s)
KILL    - exit from FHEAD without updating the file head
X       - exit from FHEAD and update the file head

Note    - an immediate, READ-ONLY form also exists:
          >FHEAD [AGGREG:]FILE 'LOOK' [K]  equivalent to:
          >FHEAD [AGGREG:]FILE
               >LOOK [K]
               >KILL
```

Function

The command has two different forms. One takes subcommands and also allows changes to be made. The other form is a read only form that does not enter subcommand mode. The output on the terminal is shown in the example below.

Cautions, Restrictions

It is not possible to change parameter 7. Attempts to alter parameters 1, 2, 3, and 10 will produce a warning message and should generally be avoided by users not very familiar with the internal orginazation of data files.

Hints

Reorganization of data files is often best done by special purpose commands like CUT, CONC, PICK, MOVE etc. In extreme cases the sequence FORMAT - (EDIT) - CONV could be used.

Miscellaneous
FHEAD

Examples

Two forms of the command are shown below with the accompanying output.

```
>FHEAD A
     >LOOK

FHEAD A

 1. NUMBER OF ROWS              2
 2. NUMBER OF COLUMNS           2
 3. THIRD DIMENSION             1
 4. SAMPLE INTERVAL             3 TICKS = 3.00000     S
 5. DATE RECORDER               0 (YY.MM.DD)
 6. TIME RECORDED               0 (HH:MM:SS)
 7. CONSTANT RECORD LENGTH 0    1 (1 MEANS YES)
 8. GENERATED BY COMMAND NR     2
 9. FILE TYPE                   3 = MATRIX
10. SKIP COUNT                  0

     >KILL

>FHEAD A LOOK 4

 4. SAMPLE INTERVAL             3 TICKS = 3.00000     S
```

FTEST

Purpose

To test the existence of files.

Command

```
FTEST [(DMODE)] FNAME

DMODE   - data mode indicator = 'A'/'D'/'T'
            A      - aggregate file
            D      - binary data file
            T      - text file
            (by default DMODE = 'D')
FNAME   - file name
FTEST.  - reserved variable returned = 0/1
            0      - the file does not exist
            1      - the file exists
```

Function

The command tries to access a file with the given name and type. If it was possible, the reserved global variable FTEST. is given the value 1, otherwise it is given the value 0.

TURN

Purpose

To alter the value of some program switches.

Command

TURN SWITCH STATE

SWITCH - switch name = 'TEXT'/'TIME'/'GRAPH'/'DK'
- TEXT - enables/disables all text output on the display (default: TEXT is enabled)
- TIME - if disabled, data will be plotted versus sample number, else versus time units (default: TIME is disabled)
- GRAPH - enables/disables all graphics output (default: GRAPH is enabled)
- DK - enables/disables command logging into text files (default: DK is enabled)

STATE - switch state = 'ON'/'OFF', if switch = 'TIME', 'H'/'M'/'S' is used instead of 'ON'
- ON - the switch is enabled
- H - plotting will be versus time in hours
- M - plotting will be versus time in minutes
- S - plotting will be versus time in seconds
- OFF - the switch is disabled

Function

The switches are set according to the value in the argument list.

Cautions, Restrictions

Note that the operation of the program also is controlled through the more flexible method of global variables, cf. the general guide.

Alphabetical Command List

AGR 5
ALTER 27
BODE 13
CONT 43
DELET 76
EDIT 7
ENTER 28
EXPAN 30
FHEAD 77
FTEST 79
HCOPY 16
KALD 46
LIST 9
MATOP 31
MOVE 11
NIC 17
NYQ 19
PLEV 22
POCONV 36
POLY 37
POLZ 40
REDUC 33
SAMP 50
SPSS 52
SSTRF1 54
SYST 55
SYSTR 61
TBALAN 63
TCON 66
TDIAG 68
THESS 71
TOBS 73
TRFSS1 75
TURN 80
UNITM 34
ZEROM 35
ZERPOL 42