ΠΑΡΑΡΤΗΜΑ

ΣΥΝΟΠΤΙΚΟΣ ΟΔΗΓΟΣ ΤΟΥ COMPREHENSIVE CONTROL

Commands: Names and descriptions

Prompts:

```
  CC>    Laplace transfer functions (ANALOG mode)
  DIG>   z-transfer functions (DIGITAL mode)
  STATE> real matrices and state space quadruples
  DATA>  real and complex data files
  MACRO> macro text editor
```

Data Entry:

```
 ENTER                                   Front end for all data entry commands
 GENTER  [,Name,coeffs]                  Transfer function
 SENTER  [,Name,coeffs]                  Transfer function, shorthand notation
 ZENTER  [,Name,coeffs,{delay}]          Transfer function, z^-1 notation
 FENTER  [prompts only]                  Function of transfer functions
 HENTER  [,Name,#rows,#cols]             Transfer function matrix, elements prompted
 PENTER  [,Name,#rows,#cols,#states,coeffs] Quadruple, elements row-wise
 CENTER  [,Name,#rows,#cols,coeffs] Complex matrix, elements real,imag row-wise
 DENTER  [,Name,#rows,#cols,type]   Data file, real or complex
 INPUT   [,Name,#rows]                   Time series (real data file)
```

Data Inquiry:

```
 WHAT    [,Name]                         Determines data type of Name
```

Data Display:

```
 DISPLAY     [,Name]                 Display any data
```

```
type
 PZF         [,Name]                 Display tf in
pole-zero-form
 TCF         [,Name]                 Display tf in
time constant form
 SHORTHAND   [,Name]                 Short hand form,
complex= [zeta,mag]
 SINGLE      [,Name]                 Display tf as
ratio of single polys
 UNITARY     [,Name]                 Display all
polynomials as unitary
 PFE         [,Name]                 Partial fraction
expansion
 ILT         [,Name{,CAUSAL/,ANTI/,roc}]   Inverse
Laplace transform
 IZT         [,Name{,CAUSAL/,ANTI/,roc}]   Inverse
z transform
 ZDISPLAY    [,Name]                 Transfer function
in z^-1 notation
 ZPZF        [,Name]                 PZF in z^-1
notation
 LPDISPLAY   [,Name{,'message'}]  Line printer
display of any data type
 LPPZF       [,Name{,'message'}]  Line printer PZF
display
 LPSHORTHAND [,Name{,'message'}]  Line printer
SHORTHAND display
 LPZDISPLAY  [,Name{,'message'}]  LPDISPLAY in z^-1
notation
 LPZPZF      [,Name{,'message'}]  LPPZF in z^-1
notation
 SCAN        [,Name]                 Scan data file
index
 FORMAT      [precision,zeros]    Set format for
displays
```

Data changing:

```
 CHANGE    [,Name1,Name2]      Front end to other
change commands
 GCHANGE   [,Name1,Name2]      Change coeffs of
```

transfer function
FCHANGE [,Name1,Name2] Change function of tf's
PCHANGE [,Name1,Name2] Change coeffs of state space quadruple
DELAY [,time] Multiply all tf's by exp(-sT) delay
COPY [,Name1,Name2] Copy Quadruple to and from Data file
CHPZF [,G1,G2] G2=Pole-Zero-Form of G1
CHTCF [,G1,G2] G2=Time-Constant-Form of G1
CHUNITARY [,G1,G2] G2=Unitary polynomial form of G1
CHSINGLE [,G1,G2] G2=Single polynomial form of G1
CHGCD [,G1,G2] G2=tfm G1 with G1_c=Greatest Common Divisor
CHLCM [,G1,G2] G2=tfm G1 with G1_c=Least Common Multiple
CHPOLY [,G1,G2] G2=tfm G1 with polynomial elements
CHONE [,G1,G2] G2=tfm G1 with G1_c=1
EXTRACT [,G1,G2, i,j/ALL] G2=tf extracted from tfm G1

Equation interpreters symbolically enter, display and change data elements:

CC> real matrices, transfer functions, transfer functions matrices
STATE> real matrices, state space quadruples, transfer functions (input only)
DATA> real matrices, complex matrices, real and complex data files

Reduced order transfer functions:

NEAR [,G1,G2{,option,eps}] Cancel poles/zeros with eps tolerance

LFAPPROX [,G1,G2,omega,order] Low freq approximation
HFAPPROX [,G1,G2,omega,order] High freq approximation

DOS file transfer:
STORE [,Name,DOS filename] Store Name in DOS filename
RECALL [,DOS filename,Name] Recall Name from DOS filename

MACRO Executution:
Syntax: @<name>,&1,&2,... (up to 40 parameters)
Macros can be called from CC, STATE, or DATA command levels
Execution always begins in CC level, even if called from STATE or DATA.

Interactive graphics and spreadsheet:
PLOT [,Name,axis,AUTO] Plot of real or complex data file
STRIP CHART [,Name,AUTO] Strip chart of real vector data file
SPREAD SHEET [,Name] Matrix spreadsheet

HP plotter support
! plot option from PLOT commands drives plotter or creates file
HPINIT Initialize HP plotter communication (otherwise use defaults)
HPPLOT [,File] Plot from previously created .HP file

Flow control, nesting okay:
FOR x=matrix & statements & END
WHILE expression & statements & END
IF expression & ELSEIF expression & ELSE statements & END

Screen control:

SCREEN [,mode{,#lines}]   Screen modes 0-
3,8-12, # text lines
COLOR [,back,axis,lines,text] Color choices
WINDOW [,#windows,#lines,#scroll lines,x,y-
offset,erase] 0 to 4 windows
CLS   [,option]   Clear screen,
options are for windows
HARDCOPY [,option]   Screen dump

Operating system commands:

FILES {,DOS file}   DOS directory (use
wildcards, defaults to data dir.)
KILL [,DOS file]   Kill DOS file (use
wildcards, defaults to data dir.)
RENAME [,DOS file1,DOS file2] Renames (default to
data directory)
OLDNEW [,CC Ver 3,CC Ver 4]   Convert data files
SHELL   {,DOS command}   Shell to DOS,
return with EXIT
ANALOG   Switch to analog mode,
CC> prompt
DIGITAL [,time]   Switch to digital mode,
DIG> prompt
ECHO [,message/ON/OFF]   Display message, on/off
for command echos from macro
LPECHO {,message}   Print message on
lineprinter
                           (n)=ASCII character, most
useful is (12)=form feed
PAUSE   {,seconds}   Pause, option for timed
pause
BACKUP {,#lines}   Back command prompt up
screen, default=1
PROMPT [,X{{,low},high}] Prompt for variable,
optional ranges, use in macros
HELP   [1/2/3/4/NAME]   Help files: options or
individual NAME (use ??'s)
QUIT   Quit
EXIT   Quit without prompt
CLOCK {DIG}/{back,pallete,face,hands} CC clock

back=0-31, pallete=0-1, face=0-3, hands=0-3
CALCULATOR HP-like Reverse Polish calculator
TIC-TAC-TOE Nonsense, beat the computer!
KILL [,DOS file] Kill DOS file (use wildcards, defaults to data dir.)
SETUP Setup operating environment, only from CC level:
Screen parameters
Location of CC modules (e.g. for RAM disk)
History file continuation
Establish links to user defined commands

Classical control plots, parameters shown for AUTO scaling:
FREQUENCY [,G,low,high,#pts,type] Create FREQ = data file for tf's
DFREQUENCY [,G,low,#pts,#reps,type] Create FREQ using foldover pi/T
POINT [G,omega] Name at s=j*omega or z=e^(j*omega*T)
MARGIN [,G{,omega}] Robustness margins, default omega=1 r/s
LPMARGIN [,G{,omega}] Line printer display of MARGIN
BODE [,type,AUTO] Bode plot using FREQ data file
NYQUIST [,AUTO] Nyquist plot using FREQ data file
INVERSE NYQUIST [,AUTO] Inverse Nyquist plot using FREQ data file
LOG NYQUIST [,AUTO] Nyquist is log axes using FREQ data file
NICHOLS [,AUTO] Nichols chart using FREQ data file
TIME [,G,option,AUTO] Time response using inverse Laplace of G(s)
DTIME [,G,option,AUTO] Time response

using inverse z of G(z)
 SIMULATION   [,G,T,opt,AUTO]   Dig. sim. using bilinear of G(s)
 DSIMULATION   [,G,option,AUTO] Dig. sim. using G(z)
 ROOT LOCUS   [,G,INFO/,AUTO]   Root locus, gain step algorithm
 FASTRL   [,G,INFO/,AUTO]   Root locus, curve tracing algorithm
 SIGGY   [,G,option,AUTO] Siggy and Bode root locus
 ASYMPTOTE   [,G,type,AUTO]   Same as SIGGY

Transfer function filters:
 1POLE   [,G,a]   Single pole
 2POLES   [,G,zeta,wn]   Two poles
 1ZERO   [,G,a]   Single zero
 2ZEROS   [,G,zeta,wn]   Two zeros
 LEADLAG   [,G,freq,phase]   Leadlag, -90 to 90 phase at freq
 NOTCH   [,G,wn,zetan,zetad] Notch at wn, depth = zetan/zetad
 INTEGRATOR [,G]   Integrator
 BUTTERWORTH [,G,freq,order]   Butterworth
 BESSEL   [,G,freq,order]   Bessel
 CHEBYSHEV   [,G,freq,dB,order]   Chebyshev, 0-3dB passband ripple
 ITAE   [,G,freq,order]   ITAE lowpass filter
 PADEDELAY   [,G,time,order]   Pade approx of exp(-s*time)

Transfer function sampled-data and multi-rate commands:
 CONVERT [,G1,G2,option,time] Convert G1(s) to G2(z), 10 options
 WPLANE   [G1,G2]   Convert G1(z) to G1(w)
 SKIP   [,G1,G2,n]   Skip sampling algorithm, every nth sample

EXPAND [,G1,G2,n] Samples spaced n apart with zero fill
REPLACE [,G1,G2,n] z^n*i replaced with z^i

Mix Laplace and z-transforms using functions of transfer functions.
Create these functions using the FENTER, FCHANGE, and FDISPLAY commands.
Perform frequency analysis using FREQUENCY,DFREQUENCY,BODE,NYQUIST, and NICHOLS.

Miscellaneous analysis:
STABILITY [,G] Tf unity feedback c.l. stability.
ROUTH [,G] Tf range of stable gains
MEAN SQUARE [,G{,Y}] Error, set Y=error if Y present

TF Wiener-Hopf, IMC, Hinf & Diophantine commands:
INNER [,G,Gin{,div}] Inner factor, div=stable/unstable boundary
OUTER [,G,Gout{,div}] Outer factor
BLASHKY [,G,B{,div}] Blashky product
ADJOINT [,G,Gstar{,div}] Adjoint
SPECTRAL [,option,G,G1,G2{,div}] Spectral factorization
PARTIAL [,option,G,G1,G2{,div}] Partial factorization
WIENER [,a,b,c{,div}] Wiener-Hopf minimization
WHOF [,a,b,c,d{,div}] Wiener-Hopf Optimal Filter
WHLQR [,G,Glqr{,rho,div}] Wiener-Hopf Linear Quadratic Regulator
WHLQG [,G,Gd,K{,mu,div,delay}] Wiener-Hopf Linear Quadratic Gaussian
HINFINITY [,a,b{,c}{,h}{,gamma}] H-infinity minimization
DIOPHANTINE [G,Phi,K] Diophantine

equation
 DIOPOLE   [,G,K]   Diophantine eq.
pole-placement
 DIOLQG   [,G,Gd,K{,mu}]   Diophantine eq.
LQG

STATE command level uses state space quadruples, packed A,B,C,D matrices

  P = [A B]   dx/dt = A*x + B*u   H(s) =
C*(sI-A)^-1*B + D
      [C D]   y = C*x + D*u

Poles and zeros:
 POLES   [,P]   Poles of P =
eigenvalues of P(A)
 ZEROS   [,P{,thresh}] Multivariable zeros
of P, by QZ algorithm
 CONMATRIX   [,Pi,P]   Create P =
controllability matrix for system Pi
 OBSMATRIX   [,Pi,Q]   Create Q =
observability matrix for system Pi
 CONTROLLABILITY [,P,radius]   Determine cont. and
obs.
 OBSERVABILITY   [,P,radius]   Determine cont. and
obs.
 POLE PLACEMENT [option,Pi,Pj] SISO pole placement

Transformations:
 CCF   [,G,P]   P = controllable
canonical form of G
 OCF   [,G,P]   P = observable
canonical form of G
 DCF   [,G,P]   P = diagonal
canonical form of G
 FADEEVA   [,P,H]   H = P by Fadeeva's
method (H is a tf or tf matrix)
 GEP   [,P,H{,thresh}]   H = P by QZ algorithm
(H is a tf or tf matrix)
 NUMERATOR   [prompts only]   Coupling numerators

of quadruple by QZ algorithm
 BALANCE   [,Pi,Pj{,INFO}]   Pj = balanced Pi, INFO to print diag scaling
 SIMILARITY [,1,Pi,T,Pj]   Pj = simil. trans. of Pi using T
    or [option,Pi,Pj,order]   options 2,3,4 use T=Hess./Schur/Eig-vectors of Pi
 RESIDUALIZATION [,Pi,Pj,n]   Pj = residualized Pi, keep n states
 SUBSYSTEM [option,Pi,Pj]   Pj = subsystem of Pi (stable, contr, obs, etc)
 BALREAL [,Pi{,In,Out},Pj]   Pj = balanced realization of Pi; In,Out=weights
 SBALANCE [,Pi{,Pj,In},Out,k] Pj = Schur balanced model reduction

Matrix decompositions:
 HESSENBERG [,A,H,U]   Hessenberg decomp: A=U*H*U'
 SCHUR   [,A,T,U,order]   Schur decomp: A=U*T*U'
 EIGENVALUE [,A,D,X,order]   Eigenvalue decomp: A=X*diag(D)*X^-1
 GHESSENBERG [,A,B,H,T,Z]   Gen. Hess. decomp: A=Q*H*Z, B=Q*T*Z
 GSCHUR   [,A,B,Tb,T,Z,order] Gen. Schur decomp: A=Q*Tb*Z, B=Q*T*Z
 GEIGENVALUE [,A,B,D,X,order]   Gen. eig. decomp: A*X=B*X*diag(D)

Singular value decomposition (SVD) and applications:
 SVD   [,A,S,U,V]   SVD: A=U*diag(S)*V'
 PSEUDO INVERSE [,A,Aplus,eps]   Aplus = V*S^+*U'
 SUBSPACE   [option,A,B,eps]   B = Range or Null space of A or A'
 PROJECTION   [option,A,B,eps]   B = projection onto subspace of A

NORM [A] or [option,A,B] 6 norms of A
RANK [A,{B},eps] Rank of A
CONDITION [A,{B},eps] Condition number of A

Matrix equations:
LYAPUNOV [,F,G,P] F'P + PF + G = 0
SYLVESTOR [,F,G,H,P] FP + PG + H = 0
RICCATI [,F,G,H,P] F'P + PF + G - PHP = 0
DLYAPUNOV [,F,G,P] F'PF - P + G = 0
DRICCATI [,F,G1,G2,H,P] F'PF - P + H - F'PG1(G2-G1'PG1)^-1G1'PF = 0

Optimal control:
LQR [as prompted] Linear quadratic regulator
KBF [as prompted] Kalman Bucy filter
LQG [as prompted] Linear quadratic Gaussion problem
DLQR [as prompted] Digital Linear quadratic regulator
DKBF [as prompted] Digital Kalman Bucy filter
DLQG [as prompted] Digital Linear quadratic Gaussian problem

Optimal control using connection matrix:
H2 [M,#inputs,#outputs,K,{X},{Y}{,comp}] H2 minimization
HINFINITY [M,#inputs,#outputs,K,{Xinf},{Yinf}{,comp}] Hinf minimization

Sampled-data and multi-rate analysis:
CONVERT [Pi,Pj,option,T] Pj = discretized Pi, 11 options
WPLANE [Pi,Pj] Pj = W-plane version of digital Pi
SKIP [,Pi,Pj,n] Pj = skip sampled Pi
EXPAND [,Pi,Pj,n] Pj = expanded Pi
EXPONENTIAL [,Pi,Pj,T{,n}] Pj = discretized

Pi using matrix exponential
WPRIME [Pi,Pj] Pj = W-plane version of analog Pi
ADVANCE [,Pi,Pj] Pj = z*Pi
DELAY [,Pi,Pj] Pj = Pi/z
RIPPLE [,P,r,n] P = [I I/z ... I/z^(n-1)], r x r*n
CONNECT [,P,r,m,nhat] P = connection matrix
VECTORIZE [,Name1,Name2,n] Name2 = input vectorized Name1
REDUCE [,Name1,Name2,n] Name2 = output reduced Name1
KRANC [,Pi,Pj,option,T,m,n] Pj = Kranc operator of Pi

State space filters:
1POLE [,P,a] Single pole
2POLES [,P,wn,zeta] Two poles
LEADLAG [,Pi,freq,phase] Leadlag, -90 to 90 phase at freq
NOTCH [,P,wn,zetan,zetad] Notch at wn, depth = zetan/zetad
INTEGRATOR [,P] Integrator
BUTTERWORTH [,P,freq,order] Butterworth
BESSEL [,P,freq,order] Bessel
CHEBYSHEV [,P,freq,dB ripple,order] Chebyshev, 0-3dB passband ripple
ITAE [,Pi,freq,ordr] ITAE lowpass filter
PADEDELAY [,P,time,order] Pade approx of exp(-s*time)

Miscellaneous:
FEEDBACK [,option,P1,P2,P3,n] Pj or Pk = Pi or Pi,Pj with feedback
PACK [,A,B,C,D,P] P=(A,B;C,D)
UNPACK [,P,A,B,C,D] (A,B;C,D)=P
MEAN SQUARE [,P{,Y}] Error, set Y=error if Y present

GRAMMIAN [,option,P,Y] Control- and observ-ability Grammians

Data files are indexed real or complex matrices:
Header
<index>
Matrix ordered row-wise
<index>
Matrix ordered row-wise
Usually, but not necessarily, data files contain time or frequency data

High level functions [Available only in DATA level]:
EIGENVALUE [,Name1,Name2] Name2 = eigenvalues of Name1
SVD [,Name1,Name2] Name2 = singular values of Name1
FMU [,Name1,Name2{,structure}] Name2 = structured s.v.'s of Name1
RMU [,Name1,Name2{,structure}] Name2 = real param. structured s.v.'s of Name1

Equations

CC equation interpreter: <LHS>=<RHS>
Allowable data types: tf's, tf matrices, real matrices
Rules for <LHS> (left hand side):
If <LHS> is missing then display <RHS>
<LHS>=Name, store <RHS> at Name
<LHS>=Name(N/D{,poly{,coeff}}), change numer, denom, polynomial, and/or coeff.
<LHS>=Name(row,col), change tf or real matrix element, augment with matrix
Rules for <RHS> (right hand side):
Operators and precedence: () : ! ' ^ | */\ +- , ; #
Feedback operation: a|b = a/(i+b*a)
Real or tf matrix building: ,;# = row, col, diagonal augmentation

scalar + square matrix = scalar*identity + square matrix

Colon convention: i:j:k = row vector containing i to k step j, default j=1

Special numbers: pi=3.14..., rnd=random number, i=1

Rules for selecting parts of data elements:

Name(rows,cols)= selected rows and cols of real or tf matrix

where 'rows' and 'cols' are row vectors, e.g. 1 1:4 1:2:7 (1,2,5)

Name(C) = tf which multiplies each element of a tf matrix

Name(N/D) = numerator or denominator of transfer function

Name(N/D,i) = i'th polynomial counting left to right as displayed

Name(N/D,i,j) = s^j'th coefficient of i'th polynomial

Functions used in CC equation interpreter:

DET(Name) = determinant

TRACE(Name) = trace

DER(Name) = derivative

EPS(Name,eps) = set elements or coefficients < eps to zero

NORM(Name) = Frobinius norm (sqr of sum of squares) of real matrix

ZERO(r,m) = r x m matrix of zeros

ONES(r,m) = r x m matrix of ones

RND(r,m) = r x m matrix of random numbers

IDEN(r) = r x r identity matrix

RDIM(Name) = row dimension

CDIM(Name) = column dimension

ODIM(Name) = output dimension, same as row dimension

IDIM(Name) = input dimension, same as column dimension

NUM(A,B,outputs,input) = coupling numerator, A,B = tf (usually poly) matrices

Matrix element functions:
LOG SIN ASIN SINH INT
SQR
LOG10 COS ACOS COSH SGN
EXP TAN ATN TANH ABS

STATE equation interpreter: <LHS>=<RHS>
Allowable data types: real matrices, quadruples,
tf's (input only)
Rules for <LHS> (left hand side):
If <LHS> is missing then display <RHS>
<LHS>=Name, store <RHS> at Name
<LHS>=Name({A/B/C/D},row,col), change element,
augment with matrix or quad
row=0 implies (max row)+1, col=0 implies (max
col)+1, for block augments
Rules for <RHS> (right hand side):
Operators and precedence: () : ! ' ^ |
*/\ +- , ; #
Feedback operation with minimal realization: a|b =
a/(i+b*a)
Matrix/quadruple building: ,;# = row, col,
diagonal augmentation
scalar + square matrix = scalar*identity + square
matrix
Colon convention: i:j:k = row vector containing i
to k step j, default j=1
Special numbers: pi=3.14..., rnd=random number,i=1
If Name=tf then convert to quadruple using CCF
realization
Dot operations (element by element): .^ .*
./ .\ .+ .-
Relations: == > >= < <= <>
Logic: @and @or not()
Rules for selecting parts of a quadruple:
Name({A/B/C/D},rows,cols) = selected A,B,C,D;
rows, and cols
Name(S,states) = selected states

Functions used in STATE equation interpreter:
DET(Name) = determinant
TRACE(Name) = trace
EPS(Name,eps) = set elements of Name < eps to zero
NORM(Name) = Frobinius norm (sqr of sum of squares) of Name
ZERO(r,m) = create r x m matrix of all zeros
ONES(r,m) = create r x m matrix of all ones
RND(r,m) = create r x m random matrix
IDEN(r) = create r x r identity matrix
DIAG(Name) = change matrix with 1 or 2 rows (or cols) to diag. matrix
CHST(Name,n) = change # states of Name to n
ADJ(Name) = Adjoint of Name, (-A',C';-B',D')
SCALE(Name,alpha) = Scaled Name, (1/alpha*A,1/alpha*B;C,D)
EXIST(name) = true(1) if datafile name exists, else false(0)

Matrix/Quadruple dimensions:
SDIM(P) = # states
RDIM(P) = ODIM(P) = # rows (equivalently # outputs)
CDIM(P) = IDIM(P) = # columns (equivalently # inputs)

Matrix element functions

LOG	SIN	ASIN	SINH	INT	SQR
LOG10	COS	ACOS	COSH	SGN	
EXP	TAN	ATN	TANH	ABS	

DATA equation interpreter: <LHS>=<RHS>
Allowable data types: data files, real & complex matrices
Algebra is performed for data files at each matching index.
Indexes for binary operations must match within

relative distance of 10^-5
Options for <LHS> (left hand side):
 If <LHS> is missing then display <RHS>
 <LHS>=Name, store <RHS> at Name
 <LHS>=Name(row,col), change element, augment with data file or matrix
Rules for <RHS> (right hand side):
 Operators and precedence: () : ! ' ^ | */\ +- , ; #
 Feedback operation: a|b = a/(i+b*a), compex conjugate: a'
 Matrix building: ,;# = row, col, diagonal augmentation
 scalar + square matrix = scalar*identity + square matrix
 Colon convention: i:j:k = row vector containing i to k step j, default j=1
 Special numbers: pi=3.14..., rnd=random number, i=1, j=sqrt(-1)
 Enter complex number as real + j*imag
Rules for selecting parts of a data file or matrix:
 Name = entire data file or matrix
 Name(A/B/C/D) = part of quadruple (entire quadruple not allowed)
 Name(rows,cols) = selected rows and cols of data file or matrix
  where rows,cols = row vectors containing entries

Functions used in DATA equation interpreter:
 DET(Name) = determinant
 TRACE(Name) = trace
 EPS(Name,eps) = set elements of Name < eps to zero
 NORM(Name) = Frobinius norm (sqr of sum of squares) of Name
 ZERO(r,m) = create r x m matrix of all zeros
 ONES(r,m) = create r x m matrix of all ones
 RND(r,m) = create r x m random matrix

```
 IDEN(r)                   = create r x r identity
matrix
 RDIM(Name)                = row dimension
 CDIM(Name)                = column dimension
 ODIM(Name)                = output dimension, same as
row dimension
 IDIM(Name)                = input dimension, same as
column dimension
 SKIP(Name,n)              = select every n'th index
 MERGE(Name1,Name2)        = merge indexes with
preference for Name1
 INDEX(Name,low,high)      = select indexes from low to
high
 CHIN(Name,alpha,beta) = change index to
alpha*index + beta
Matrix element functions
  LOG       SIN      ASIN      SINH      INT
SQR      CONJ
  LOG10     COS      ACOS      COSH      SGN
REAL     PHASE
  EXP       TAN      ATN       TANH      ABS
IMAG
```