

RANGE DI UNA MATRICE**RANGE**

Tale programma permette di determinare il range di una matrice Sotto forma di equazioni

1: MATRICE

```

« TRN NUCLEO DROP
DUP
  IF { } ==
  THEN
"IL RANGE
RAPPRESENTA TUTTO
LO SPAZIO"
MSGBOX DROP
  ELSE DUP SIZE
'DIM' STO
  IF 'DIM(2)==2'
  THEN { x1 x2 }
'vet' STO
[ 0 0 ]
'NUL' STO
  END
  IF 'DIM(2)==3'
  THEN { x1 x2 x3 }
'vet' STO
[ 0 0 0 ]
'NUL' STO
  END
  IF 'DIM(2)==4'
  THEN { x1 x2 x3
x4 } 'vet' STO
[ 0 0 0 0 ]
'NUL' STO

DIM(1)' EVAL 'DIM'
STO 0 'cont' STO 1
DIM EVAL
  FOR j DUP
  IF NUL ==
  THEN DROP DIM
1 - 'DIM' STO
  ELSE HOME
HURWI UTILITY A~~L
HOME RANGE vet *

ROLLD cont 1 +
'cont' STO
  END

{ cont NUL vet DIM
n } PURGE
  END
»

```

NUCLEO DI UNA MATRICE**NUCLEO**

Tale programma permette di determinare il nucleo di una matrice Sia sotto forma di vettori che sotto forma di equazioni

1: MATRICE

```

« 'Fr' STO Fr Fr
SIZE 'DIM' STO { '
DIM(1)' } 0 CON '
DIM(2)+1' EVAL COL+
'Fr' STO
  IF 'DIM(2)==2'
  THEN { x1 x2 }
'vet' STO
[ 0 0 ]
'NUL' STO
  END
  IF 'DIM(2)==3'
  THEN { x1 x2 x3 }
'vet' STO
[ 0 0 0 ]
'NUL' STO
  END
  IF 'DIM(2)==4'
  THEN { x1 x2 x3
x4 } 'vet' STO
[ 0 0 0 0 ]
'NUL' STO

EVAL 'DIM' STO 1
DIM 0 'cont' STO
  FOR j DUP
  IF NUL ==
  THEN DROP DIM 1
- 'DIM' STO
  ELSE DUP HOME
HURWI UTILITY A~~L
HOME RANGE vet *

+ ROLLD DIM 1 +
ROLLD cont 1 +
'cont' STO
  END

'matr' STO { NUL
vet cont Fr } PURGE
matr SIZE 'DIM' STO
'DIM(2)-DIM(1)'

```

```

EVAL 'DIM3' STO
  IF DIM3 0 >
    THEN 0 'n' STO
    DO { DIM3 'DIM(
2)' } 0 CON 'matr2'

      IF n 0 <
        THEN 1 DIM3
        FOR q DROP
        NEXT
      END matr2

STO n 1 + 'n' STO n
'm' STO 'DIM(1)'
EVAL 1 + 'DIM(2)'
EVAL
  FOR j
  IF 'n>DIM(2)' THEN
  1 'n' STO END
matr {j n } 1 PUT 'matr'
STO n 1 + 'n' STO
  NEXT m 'n'
STO
  UNTIL matr RANK
'DIM(2)' EVAL ==
  END { matr2 m }
PURGE 'DIM(1)' EVAL
1 + 'DIM(2)' EVAL
  FOR j { 'DIM(2)
' } 0 CON { j } 1
PUT matr LSQ 4 RND

  ELSE { }
  END eqz { DIM
DIM3 matr eqz n }
PURGE »

```

AUTOVETTORI GENERALIZZATI

AUTGEN

Tale programma permette di determinare gli autovettori generalizzati

- 3: autovalore
- 2: autovettore generalizzato di ordine appena minore
- 1: matrice

```

« 'Fr' STO 'avt'
STO 'avl' STO Fr
SIZE 'DIM' STO DIM
HEAD IDN avl * Fr -
'Fr' STO Fr avt NEG
DIM HEAD 1 + COL+
'Frplus' STO
  IF Fr RANK Frplus
RANK ==

```

```

  THEN Frplus 'Fr'
STO
  IF 'DIM(2)==2'
  THEN
  [ 0 0 ]
  'NUL' STO
  END
  IF 'DIM(2)==3'
  THEN
  [ 0 0 0 ]
  'NUL' STO
  END
  IF 'DIM(2)==4'
  THEN
  [ 0 0 0 0 ]
  'NUL' STO
  END Fr RREF

'DIM(1)' EVAL 'DIM'
STO 1 DIM
  FOR j DUP
  IF NUL ==
  THEN DROP DIM
  1 - 'DIM' STO avt

  ELSE DIM

'nue' STO DIM ROLLD

  END

'matr' STO matr
SIZE 'DIM' STO 'DIM
(2)-DIM(1)' EVAL
'DIM3' STO
  IF DIM3 0 >
  THEN 0 'n' STO
  DO { DIM3 '
DIM(2)' } 0 CON
'matr2' STO { DIM3
} 0 CON 'avt2' STO

  IF n 0 <
  THEN 1 DIM3
  FOR q
DROP
  NEXT
  END avt

  IF n 0 <
  THEN 1 DIM3
  FOR q
DROP nue 1 - 'nue'
STO
  NEXT
  END nue

```

```
'avt' STO n 1 + 'n'
STO n 'm' STO 'DIM(
1)' EVAL 1 + 'DIM(2
)' EVAL
      FOR j matr
{ j n } 1 PUT
'matr' STO n 1 +
'n' STO
      NEXT m 'n'
STO
      UNTIL matr
RANK 'DIM(2)' EVAL
==
      END 'DIM(1)'
EVAL 1 + 'DIM(2)'
EVAL
      FOR j avt { j
} 1 PUT matr LSQ 4
RND
      NEXT DIM3

      ELSE { }
      END
      ELSE
" Non ci sono
altri
autovettori
generalizzati"
MSGBOX
      END { DIM Fr avt2
nue Frplus avl
matr2 m NUL vet
cont DIM3 eqz n
matr avt } PURGE
»
```

ELEVAMENTO A POTENZA

ELEV

Tale programma permette l'elevamento a potenza di una matrice dove la potenza deve essere necessariamente un numero naturale

2: POTENZA

1: MATRICE

```
« 'Fe' STO 'POT'
STO Fe SIZE HEAD
'DIM' STO
IF POT 0 ==
```

```
THEN DIM IDN
END
IF POT 1 ==
THEN Fe
END
IF POT 1 >
THEN Fe 1 POT 1 -
      FOR i Fe *
      NEXT
END { POT DIM Fe
} PURGE
»
```

COEFFICIENTI DEL POLINOMIO CARATTERISTICO

COEFF

Tale programma permette di determinare i coefficienti del polinomio caratteristico.

Il risultato è invertito rispetto all'ordine delle potenze del polinomio

1: MATRICE

```
«
[ 0 1 ]
'a' STO 'F' STO F
TRACE NEG 'a1' STO
a { 1 } a1 PUT 'a'
STO F SIZE HEAD
'dim' STO 2 dim
      FOR n 0 'SOMM'
STO 1 n
      FOR j 'a(j)'
EVAL j F ELEV TRACE
* SOMM + 'SOMM' STO
      NEXT 1 n / NEG

'a' STO
      NEXT a { DIM SOMM
j dim a1 a F }
PURGE
»
```

MATRICE P PER L'EQUAZIONE DI LYAPUNOV DI SISTEMI CONTINUI

PRUN

Tale programma ti permette di determinare, se esiste la matrice P dell'equazione di Lyapunov. In base al risultato ottenuto è possibile in base al teorema di Lyapunov caratterizzare immediatamente il sistema lineare tempo continuo che si sta valutando

1: MATRICE F

```

« 'Fp' STO
[[ 0 0 0 ]
 [ 0 0 0 ]
 [ 0 0 0 ]]
'Q' STO Q { 1 1 }
Fp { 1 1 } GET 2 *
PUT 'Q' STO Q { 1 2 }

} Fp { 2 1 } GET 2
* PUT 'Q' STO Q { 2
1 } Fp { 1 2 } GET
PUT 'Q' STO Q { 2 2
} Fp { 1 1 } GET Fp
{ 2 2 } GET + PUT
'Q' STO Q { 2 3 }
Fp { 2 1 } GET PUT
'Q' STO Q { 3 2 }
Fp { 1 2 } GET 2 *
PUT 'Q' STO Q { 3 3
} Fp { 2 2 } GET 2
* PUT 'Q' STO
  IF Q RANK 3 ==
    THEN Q INV
  [ -1 0 -1 ]
* 'VAL' STO
[[ 0 0 ]
 [ 0 0 ]]
'P' STO P { 1 1 }
VAL 1 GET PUT 'P'
STO P { 1 2 } VAL 2
GET PUT 'P' STO P {
2 1 } VAL 2 GET PUT

```

```

'P' STO P { 2 2 }
VAL 3 GET PUT 'P'
STO { VAL Fp }
PURGE P END »

```

MATRICE P PER L'EQUAZIONE DI LYAPUNOV DI SISTEMI DISCRETI

PRUN

Tale programma ti permette di determinare, se esiste la matrice P dell'equazione di Lyapunov. In base al risultato ottenuto è possibile in base al teorema di Lyapunov caratterizzare immediatamente il sistema lineare tempo discreto che si sta valutando

1: MATRICE F

```

« 'Fd' STO
[[ 0 0 0 ]
 [ 0 0 0 ]
 [ 0 0 0 ]]
'Q' STO Fd { 1 1 }
GET 'a11' STO Fd {
1 2 } GET 'a12' STO
Fd { 2 1 } GET
'a21' STO Fd { 2 2
} GET 'a22' STO Q {
1 1 } 'a11^2-1'
EVAL PUT 'Q' STO Q
{ 1 2 } '2*a11*a21'
EVAL PUT 'Q' STO Q
{ 1 3 } 'a21^2'
EVAL PUT 'Q' STO Q
{ 2 1 } 'a11*a12'
EVAL PUT 'Q' STO Q
{ 2 2 } 'a22*a12+
a11*a22-1' EVAL PUT
'Q' STO Q { 2 3 } '
a21*a22' EVAL PUT
'Q' STO Q { 3 1 } '
a12^2' EVAL PUT 'Q'
STO Q { 3 2 } '2*
a12*a22' EVAL PUT
'Q' STO Q { 3 3 } '
a22^2-1' EVAL PUT
'Q' STO { a11 a12
a21 a22 } PURGE
  IF Q RANK 3 ==
    THEN Q INV
  [ -1 0 -1 ]
* 'VAL' STO
[[ 0 0 ]

```

```
[ 0 0 ]]
'P' STO P { 1 1 }
VAL 1 GET PUT 'P'
STO P { 1 2 } VAL 2
GET PUT 'P' STO P {
2 1 } VAL 2 GET PUT
'P' STO P { 2 2 }
VAL 3 GET PUT 'P'
STO 'VAL' PURGE P
END 'Fd' PURGE »
```

DETERMINANTI DEI MINORI DOMINANTI DI UNA MATRICE

MINOR

Tale programma permette di determinare tutti i determinanti dei minori dominanti di una matrice

1: MATRICE

```
« 'HUR' STO HUR
SIZE HEAD 'DIM' STO
1 DIM
FOR j HUR { 1 1 }
{ j j } SUB DET
HUR DIM } PURGE»
```

MATRICE JACOBIANA

1: {f1 f2 f3}

```
« 1 Z
START DUP DUP
dx ^ ROT dy ^ ROT
IF Z 2 >
THEN dz ^
ELSE DROP
Z ROLL
» » »
```

ANALISI DELLA STABILITA' DI UN SISTEMA NON LINEARE TEMPO CONTINUO 2x2

Vx

Tale programma permette di linearizzare un sistema non lineare 2x2 tempo continuo, analizza la stabilità e se per caso risulta asintoticamente stabile allora ne calcola la V(x)

1: { f1 f2 }

```
« 'X' STO SYMB { x1
x2 0 } JACOBI UPDIR
'X(1)' EVAL 'x1'
STO 'X(2)' EVAL
'x2' STO EVAL EVAL
3 ROLL EVAL { 2 2 }
0 CON 'F1' STO EVAL
F1 { 1 2 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 1 1 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 2 2 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 2 1 } 3 ROLL
PUT 'F1' STO { x2
x1 } PURGE F1 EGVL
'AUT' STO
IF 'AUT(1)' EVAL
RE 4 RND 0 == 'AUT(
2)' EVAL RE 4 RND 0
== AND THEN
"un autovalore é nullo"
MSGBOX ELSE
IF 'AUT(1)'
EVAL RE 4 RND 0 < '
AUT(2)' EVAL RE 4
RND 0 < AND
THEN F1 HOME
PRUN CONTINUO
RUNP
HOME VLIN CONTINUO
'P' STO
" l'equilibrio è asintoticamente
stabile"
MSGBOX { 3 } 0 CON
'V' STO V { 1 } 'P(
1,1)' EVAL PUT 'V'
STO V { 2 } 'P(1,2)
+P(2,1)' EVAL PUT
'V' STO V { 3 } 'P(
2,2)' EVAL PUT 'V'
STO V MATUTILS A~~L
UPDIR { '(x1-X(1))^
2' '(x1-X(1))*(x2-X
(2))' '(x2-X(2))^2'
} EVAL EVAL SWAP
EVAL 3 ROLL EVAL
```

```
"La funzione V(x)"
"è la seguente" V
  ELSE
"L'equilibrio è instabile"
MSGBOX END
END 'X' PURGE »
```

ANALISI DELLA STABILITA' DI UN SISTEMA NON LINEARE TEMPO DISCRETO 2x2

Vx

Tale programma permette di linearizzare un sistema non lineare 2x2 tempo discreto, analizza la stabilità e se per caso risulta asintoticamente stabile allora ne calcola la V(x)

1: { f1 f2 }

```
« 'X' STO SYMB { x1
x2 0 } JACOBI UPDIR
'X(1)' EVAL 'x1'
STO 'X(2)' EVAL
'x2' STO EVAL EVAL

3 ROLL EVAL { 2 2 }
0 CON 'F1' STO EVAL
F1 { 1 2 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 1 1 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 2 2 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 2 1 } 3 ROLL
PUT 'F1' STO { x2
x1 } PURGE F1 EGVL
'AUT' STO
  IF 'AUT(1)' EVAL
4 RND ABS 1 < 'AUT(
2)' EVAL 4 RND ABS
1 < AND
  THEN
  IF 'AUT(1)'
EVAL ABS 1 < 'AUT(2
)' EVAL ABS 1 < AND
  THEN F1 HOME
PRUN DISCRETO RUNP
HOME VLIN DISCRETO
'P' STO
"L'equilibrio è asintoticamente
stabile"
MSGBOX { 3 } 0 CON
'V' STO V { 1 } 'P(
```

```
1,1)' EVAL PUT 'V'
STO V { 2 } 'P(1,2)
+P(2,1)' EVAL PUT
'V' STO V { 3 } 'P(
2,2)' EVAL PUT 'V'
STO V MATUTILS A~L
UPDIR { '(x1-X(1))^
2' '(x1-X(1))*(x2-X
(2))' '(x2-X(2))^2'
} EVAL EVAL SWAP
EVAL 3 ROLL EVAL
```

```
"La funzione V(x)"
"è la seguente" V
  ELSE
"L'equilibrio è instabile"
MSGBOX
  END
  ELSE
"un autovalore é 1"
MSGBOX
  END 'X' PURGE »
```

ANALISI DELLA STABILITA' DI UN SISTEMA NON LINEARE TEMPO CONTINUO 3x3

Flin

Tale programma permette di linearizzare un sistema non lineare 2x2 tempo continuo e analizza la stabilità

1: { f1 f2 f3 }

```
« 'X' STO { x1 x2
x3 } UPDIR UPDIR
CONTINUO SYMB
JACOBI UPDIR ~X3
CONTINUO 'X(1)'
EVAL 'x1' STO 'X(2)
' EVAL 'x2' STO 'X(
3)' EVAL 'x3' STO
```

```

EVAL { 3 3 } 0 CON
'F1' STO EVAL EVAL
F1 { 3 3 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 3 2 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 3 1 } 3 ROLL
PUT 'F1' STO EVAL
EVAL F1 { 2 3 } 3
ROLL PUT 'F1' STO
EVAL F1 { 2 2 } 3
ROLL PUT 'F1' STO
EVAL F1 { 2 1 } 3
ROLL PUT 'F1' STO
EVAL EVAL F1 { 1 3
} 3 ROLL PUT 'F1'
STO EVAL F1 { 1 2 }
3 ROLL PUT 'F1' STO
EVAL F1 { 1 1 } 3
ROLL PUT 'F1' STO
F1 EGVL 'AUT' STO
  IF 'AUT(1)' EVAL
RE 4 RND 0 == 'AUT(
2)' EVAL RE 4 RND 0
== 'AUT(3)' EVAL RE
4 RND 0 == OR OR
  THEN
"E' presente un autovalore con parte
reale uguale a 0"
MSGBOX
  ELSE
  IF 'AUT(1)'
EVAL RE 4 RND 0 < '
AUT(2)' EVAL RE 4
RND 0 < 'AUT(3)'
EVAL RE 4 RND 0 <
AND AND
  THEN
"L'equilibrio è asintoticamente
stabile"
MSGBOX
  ELSE
"L'equilibrio è instabile"
MSGBOX
  END
  END { x3 x2 x1 X
} PURGE »

```

ANALISI DELLA STABILITA' DI UN SISTEMA NON LINEARE TEMPO DISCRETO 3x3

F1n

*Tale programma permette di
linearizzare un sistema non lineare*

*2x2 tempo discreto e analizza la
stabilità*

1: { f1 f2 f3 }

```

« 'X' STO { x1 x2
x3 } UPDIR UPDIR
CONTINUO SYMB
JACOBI UPDIR ~X3
DISCRETO 'X(1)'
EVAL 'x1' STO 'X(2)
' EVAL 'x2' STO 'X(
3)' EVAL 'x3' STO
EVAL { 3 3 } 0 CON
'F1' STO EVAL EVAL
F1 { 3 3 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 3 2 } 3 ROLL
PUT 'F1' STO EVAL
F1 { 3 1 } 3 ROLL
PUT 'F1' STO EVAL
EVAL F1 { 2 3 } 3
ROLL PUT 'F1' STO
EVAL F1 { 2 2 } 3
ROLL PUT 'F1' STO
EVAL F1 { 2 1 } 3
ROLL PUT 'F1' STO
EVAL EVAL F1 { 1 3
} 3 ROLL PUT 'F1'
STO EVAL F1 { 1 2 }
3 ROLL PUT 'F1' STO
EVAL F1 { 1 1 } 3
ROLL PUT 'F1' STO
F1 EGVL 'AUT' STO
  IF 'AUT(1)' EVAL
ABS 4 RND 1 == 'AUT
(2)' EVAL ABS 4 RND
1 == 'AUT(3)' EVAL
ABS 4 RND 1 == OR
  OR
  THEN
"E' presente un autovalore con modulo
uguale 1"
MSGBOX
  ELSE
  IF 'AUT(1)'
EVAL ABS 4 RND 1 <
'AUT(2)' EVAL ABS 4
RND 1 < 'AUT(3)'
EVAL ABS 4 RND 1 <
AND AND
  THEN
"L'equilibrio è asintoticamente
stabile"
MSGBOX
  ELSE
"L'equilibrio è instabile"
MSGBOX
  END
  END

```

```
END { x1 x2 x3 X
} PURGE
»
```

MATRICE DI HURWITZ

HURWI

Tale programma permette di determinare la matrice di Hurwitz
 In base a tale matrice è possibile valutare la posizione degli autovalori della matrice F e quindi valutare la stabilità del sistema che si sta valutando

1: Matrice F

oppure

1: [coefficienti del polinomio caratteristico]

```
« 'CALC' STO CALC
SIZE SIZE 'DIM' STO
IF DIM 2 ==
THEN CALC UTILITY
COEFF A~~L REVLIST
A~~L UPDIR 'CALC'
STO
END
IF CALC SIZE HEAD
2 MOD 1 ==
THEN CALC UTILITY
A~~L 0 + A~~L UPDIR
'CALC' STO
END CALC SIZE
HEAD 1 - 'DIM' STO
{ DIM DIM } 0 CON
'Fh' STO 0 DIM
FOR i 1 DIM 2 / 1
+
FOR j j 2 *
'jj' STO Fh { 'j+i/
2' '1+i' } 'CALC(jj
)' EVAL PUT 'Fh'
```

```
STO
NEXT
IF '2+i' EVAL
DIM <
THEN Fh { '1+i/
2' '2+i' } 1 PUT
'Fh' STO 2 DIM 2 /
1 +
FOR j j 2 *
'jj' STO Fh { 'j+i/
2' '2+i' }
'CALC(jj-1)'
EVAL PUT 'Fh' STO
NEXT END 2
STEP Fh { p jj Fh
DIM CALC } PURGE »
```

DETERMINAZIONE DI $P((w+1)/(w-1))=0$

Lamda-omega

Questo programma permette di valutare il polinomio $P((w+1)/(w-1))=0$ in base alla quale è possibile applicare la matrice di Hurwitz anche per i sistemi lineari tempo discreti

Una volta eseguito il programma bisogna eseguire ripetutamente COLCT ed EXPAND che si trovano nelle applicazioni SYMBOLIC dell'HP

1: [coefficienti del polinomio caratteristico]

```
« 'P-' STO P- SIZE
HEAD 1 - 'DIM' STO
P- UTILITY A~~L
UPDIR 0 DIM
FOR j '(š+1)^(DIM
-j)*(š-1)^j' EVAL
NEXT DIM 1 +
DIM P- } PURGE »
```


DETERMINAZIONE DELLA MATRICE DI KALMAN

K

*Tale programma permette di calcolare
la matrice di Kalman*

2: matrice F
1: matrice G (o vettore)

```

« 'Gp' STO 'Fp' STO
Fp SIZE HEAD 'DIM'
STO 0 'num' STO Gp
SIZE SIZE 'dimG'
STO
  IF dimG 1 ==
  THEN 0 DIM 1 -
    FOR j j Fp
UPDIR T ELEV UPDIR
KALMAN Gp *

  ELSE 0 DIM 1 -
    FOR j j Fp
UPDIR T ELEV UPDIR

num + 'num' STO

  END { DIM num
dimG Gp Fp } PURGE »

```

DETERMINAZIONE DELLA MATRICE DI OSSERVABILITA'

K0

*Tale programma permette di calcolare
la matrice di osservabilità*

2: matrice F
1: matrice H (o vettore)

```

« 'Hp' STO 'Fp' STO
Fp SIZE HEAD 'DIM'
STO 0 'num' STO 0
DIM 1 -
  FOR j Hp j Fp
UPDIR T ELEV UPDIR

'num' STO

DIM num Fp Hp }
PURGE »

```

DETERMINAZIONE DELLA MATRICE DI CONTROLLABILITA' DELL' USCITA

Ky

*Tale programma permette di calcolare
la matrice di controllabilità
dell'uscita*

3: matrice F
2: matrice G (o vettore)
1: matrice H (o vettore)

```

« 'Hp' STO 'Gp' STO
'Fp' STO Fp SIZE
HEAD 'DIM' STO 0
'num' STO Gp SIZE
SIZE 'dimG' STO
  IF dimG 1 ==
  THEN 0 DIM 1 -
    FOR j Hp j Fp
UPDIR T ELEV UPDIR
KALMAN * Gp *

  ELSE 0 DIM 1 -
    FOR j Hp j Fp
UPDIR T ELEV UPDIR

num + 'num' STO

  END { DIM num
dimG Gp Fp Hp }
PURGE »

```

**DETERMINAZIONE DEL RANGE
DELL' SOTTOSPAZIO DI
RAGGIUNGIBILITA' PER I
SISTEMI LINEARI TEMPO
CONTINUI (E DISCRETI)**

Xr

2: *MATRICE F*
1: *MATRICE G (o vettore)*

```
« HOME KALMAN K TRN
HOME RANGE NUCLEO
DROP NUCLEO HOME
Xorc CONTINUO (DISCRETO)»
```

**DETERMINAZIONE DEL RANGE
DELL' SOTTOSPAZIO DI NON
RAGGIUNGIBILITA' PER I
SISTEMI LINEARI TEMPO
CONTINUI (E DISCRETI)**

Xnr

2: *MATRICE F*
1: *MATRICE G (o vettore)*

```
« HOME KALMAN K TRN
HOME RANGE NUCLEO
DROP HOME Xorc
CONTINUO (DISCRETO)»
```

**DETERMINAZIONE DEL RANGE
DELL' SOTTOSPAZIO DI
OSSERVABILITA' PER I
SISTEMI LINEARI TEMPO
CONTINUI (E DISCRETI)**

Xo

2: *MATRICE F*
1: *MATRICE G (o vettore)*

```
« HOME KALMAN K0
HOME RANGE NUCLEO
DROP NUCLEO DROP
HOME Xorc CONTINUO (DISCRETO)»
```

**DETERMINAZIONE DEL RANGE
DELL' SOTTOSPAZIO DI NON
OSSERVABILITA' PER I
SISTEMI LINEARI TEMPO
CONTINUI (E DISCRETI)**

Xno

2: *MATRICE F*
1: *MATRICE G (o vettore)*

```
« HOME KALMAN K0
HOME RANGE NUCLEO
DROP HOME Xorc
CONTINUO (DISCRETO)»
```

**DETERMINAZIONE DEL RANGE
DEI SOTTOSPAZI DI
CONTROLLABILITA' NEGLI n
PASSI PER I SISTEMI LINEARI
TEMPO DISCRETI**

Xcn

2: *MATRICE F*
1: *MATRICE G (o vettore)*

```
« 'Gc' STO 'Fc' STO
Fc SIZE HEAD 'DF'
STO Gc UTIL TRNG
UPDIR TRN SIZE 'DG'
STO 1 DF
  FOR n n Fc UTIL
ELEV UPDIR TRN Fc
Gc HOME KALMAN K
HOME Xorc DISCRETO
  IF n DF <

1 '(DF-n)*DG(2)'
EVAL
  FOR p DROP

  NEXT 'n*DG(2)

  END UTIL TRNG
UPDIR HOME RANGE
NUCLEO DROP DUP
HOME Xorc DISCRETO
IF { } <
```

```

    THEN UTIL TRNG
* TRNG UPDIR DUP
    IF RANK 0 <
    THEN HOME
RANGE NUCLEO DROP
HOME Xorc DISCRETO
    ELSE DROP 'X'
    END
    ELSE DROP DROP
'X' END
    NEXT { DG DF Fc
Gc } PURGE»

```

TRASPOSTA DI MATRICI E DI VETTORI

Sottoprogramma utilizzato dal programma precedente

TRNG

1: MATRICE (o vettore)

```

« 'A' STO A SIZE
'dim' STO
  IF dim SIZE 1 ==
  ELSE
    IF 'dim(1)<1'
    THEN A TRN
    ELSE
    END
    IF 'dim(1)==1'
  DROP
  END
  END { dim A }
PURGE »

```

MATRICE COMPAGNA

1: MATRICE

```

« 'F' STO F COEFF
'vect' STO F SIZE
HEAD 'dim' STO {
dim dim } 0 CON

```

```

'Fc' STO 1 dimm 1 -
  FOR E E 1 + 'β'
STO Fc { E β } 1
PUT 'Fc' STO
  NEXT 1 dimm
  FOR p Fc { dimm p
} 'vect(p)' EVAL
NEG PUT 'Fc' STO
  NEXT { β dimm
vect } PURGE Fc»

```

MATRICE DI PASSAGGIO ALLA FORMA CANONICA DI CONTROLLO

T

Tale programma permette di calcolare la matrice T utile per passare alla rappresentazione della forma canonica di controllo. In tale programma è possibile conoscere anche F_c e G_c

2: MATRICE F
1: VETTORE G

```

« 'G' STO 'F' STO F
G UPDIR KALMAN K
UPDIR T 'K' STO F
COMP F SIZE HEAD
'dim' STO { diM } 0
CON 'Gc' STO Gc {
diM } 1 PUT 'Gc'
STO Gc UPDIR KALMAN
K UPDIR T 'Kc' STO
K Kc INV * { diM }
PURGE»

```

MATRICE DI PASSAGGIO ALLA FORMA CANONICA DI OSSERVAZIONE

To

Tale programma permette di calcolare la matrice T_o utile per passare alla rappresentazione della forma canonica di osservazione. In tale programma è possibile conoscere anche F_o e H_o

2: MATRICE F
1: VETTORE [[RIGA]] H

```

« 'H' STO 'F' STO F
H UPDIR KALMAN K0
UPDIR T 'K0' STO F
COMP TRN 'Fo' STO
Fo F SIZE HEAD
'dim' STO { 1 dim }
0 CON { 1 dim } 1
PUT 'Ho' STO Ho
UPDIR KALMAN K0
UPDIR T 'Ko' STO K0
INV Ko * { diM Fc }
PURGE »

```

```

IF 'IM(
val(i))>IM(val(j))'
EVAL
THEN '2
*k*‡/IM(val(i)-val(
j))' EVAL COLCT
END
END
END
NEXT
NEXT
ELSE
"S non è complet. raggiungibile e
controllabile"
MSGBOX
END { DIM dimval
G1 val F1 } PURGE »

```

RAGGIUNGIBILITA' DEI SISTEMI A DATI CAMPIONATI

T

Questo programma determina i valori di T (periodo di campionamento) tale che il sistema a dati campionati perde la completa raggiungibilità. Questo si fa a patto che il sistema lineare tempo continuo presente all'interno sia completamente raggiungibile

2: MATRICE F (del sistema continuo S)
1: MATRICE G (del sistema continuo S)

```

« 'G1' STO 'F1' STO
F1 SIZE HEAD 'DIM'
STO
IF F1 G1 HOME
KALMAN K HOME CAMP
RANK DIM ==
THEN
"S é complet.
raggiungibile e controllabile"
MSGBOX F1 EGVL
'val' STO val SIZE
HEAD 'dimval' STO 1
dimval
FOR i 1 dimval
FOR j
IF i j <
THEN
IF 'val(i
)' EVAL RE 'val(j)'
EVAL RE ==
THEN

```

REALIZZAZIONE MINIMA

REALIZZ

Tale programma determina le matrici F G H sia in forma canonica di controllo che in forma canonica di osservazione partendo dalla funzione di trasferimento

3: COSTANTE CHE MOLTIPLICA IL
NUMERATORE DELLA f.d.t.
2: [COEFFICIENTI DEL
NUMERATORE DELLA f.d.t.]
1: [COEFFICIENTI DEL
DENOMINATORE DELLA f,d,t.]

```

« 'Den' STO 'Num'
STO 'cost' STO 0

```

```

'L' STO 'Num' EVAL
SIZE HEAD 1 - 'a'
STO 'Den' EVAL SIZE
HEAD 1 - 'b' STO
  IF a b %
  THEN
    IF a b ==
    THEN Num Den
POLY DIVV UPDIR
'Den' STO 'Num' STO
'L' STO a 1 - 'a'
STO
  END
  ELSE
"IL NUMERATORE NON PUO' AVERE GRADO
MAGGIORE SUL
DENOMINATORE"
MSGBOX
  END
  IF a 0 >
  THEN Num PROOT
'Num' STO Den PROOT
3 RND 'Den' STO { }
'zero' STO Num POLY
A~~L UPDIR 'Num'
STO Den POLY A~~L
UPDIR 'Den' STO 1 a
  FOR j Num HEAD
'ELnum' STO Num
TAIL 'Numrest' STO
1 b
  FOR q Den
HEAD 'ELden' STO
Den TAIL 'Denrest'
STO
  IF ELden
ELnum ==
  THEN
Denrest 'Den' STO
zero ELnum + 'zero'
STO
  ELSE
Denrest ELden +
'Den' STO
  END
  NEXT
NEXT zero SIZE
'c' STO
  IF c 0 <
  THEN 1 c
  FOR jj zero
HEAD 'ELzero' STO
zero TAIL
'zerorest' STO 1 a
  FOR qq Num
HEAD 'ELnum' STO
Num TAIL 'Numrest'
STO
  IF ELnum
ELzero ==

```

```

  THEN
Numrest 'Num' STO
  ELSE
Numrest ELnum +
'Num' STO
  END
  NEXT
NEXT
  END { zerorest
ELzero c Denrest
ELden Numrest ELnum
zero } PURGE
  IF Num { } ==
  THEN
[ 1 ]
cost * 'Num' STO
  ELSE Num POLY
A~~L UPDIR PCOEF
cost * 'Num' STO
  END Den POLY
A~~L UPDIR PCOEF
'Den' STO
  END { cost }
PURGE Den SIZE HEAD
1 - 'dim' STO { dim
dim } 0 CON 'F' STO
{ 1 dim } 0 CON 'H'
STO { dim 1 } 0 CON
'G' STO
  IF dim 1 >
  THEN 1 dim 1 -
  FOR G G 1 + 'B'
STO F { G B } 1 PUT
'F' STO
  NEXT 1 dim
  FOR p F { dim p
} 'Den(dim+2-p)'
EVAL 3 RND NEG PUT
'F' STO
  NEXT G { dim 1
} 1 PUT 'G' STO Num
SIZE HEAD 'dim2'
STO 1 dim2
  FOR p H { 1 p }
'Num(dim2+1-p)'
EVAL 3 RND PUT 'H'
STO NEXT
  ELSE F { 1 1 } '
Den(2)' EVAL NEG
PUT 'F' STO H { 1 1
} 1 PUT 'H' STO G {
1 1 } 1 PUT 'G' STO
  END { dim2 B dim
Num Den } PURGE F
TRN 'Fo' STO H TRN
'Go' STO G TRN 'Ho'
STO { b a } PURGE »

```

DIVISIONE TRA DUE POLINOMI**DIVV**

Programma utilizzato dal programma precedente

2: [COEFFICIENTI DEL DIVIDENDO]
1: [COEFFICIENTI DEL DIVISORE]

```

« a 1 GET c / DUP
b * a SIZE RDM a

DROP b
» »

```

**PASSAGGIO DA MATRICE
A LISTA****A~~L**

Programma utilizzato da molti programmi precedenti, utile per rappresentare una matrice tramite lista e quindi dare la possibilità di affrontare anche calcoli simbolici

1: MATRICE

```

« IF DUP TYPE 5 ==
  THEN
    IF DUP 1 GET
TYPE 5 ==
      « 1 a SIZE
        FOR i a i

          NEXT a SIZE

      »

    END
  ELSE
    IF DUP SIZE
SIZE 2 ==
SWAP 1
  FOR i i 1 +

    STEP

  END
END»

```