Mult&T

for use with MATLAB

Franklin Pineda

Contents

Appendix A

Function Reference

A.1. About Mult&T

Mult&T is user's interface created with Matlab to facilitate the handling of lineal models of control multivariable, likewise it spreads to incorporate developments that allow the design of controllers multivariables for different methods. Among the main functions that at the moment presents:

- Find minimal realizations for different methods.
- Conversions and characteristic main of the models MIMO (matrix transfer functions (MTF), the matrix polinomial fraction(MPF) and the models in state space (SS)).
- Allows similarity transformations for realizations in canonical form.
- Incorporates balanced realizations and order reduction in the models

March 2009

A.2. Installation

The installation is straightforward just copy the directory *'Multitool'* and add the path to the MATLAB search path.

See *path*, in the MATLAB documentation for more information.

A.3. Requeriments

Mult&T was created in Matlab 7.4 (R2007a), and requires the Symbolic and Control Toolboxes.

A.4. Contact

- `http://www.mathworks.com`
- `http://www.matlabcentral.com`
- `fe.pineda92@uniandes.edu.co`

A.5. Function Description

A.5.1. mtfsp

Sintax

```
[Gsp Ginf]=mtfsp(G)
```

Description

`mtfsp` separates a matrix transfer functions in their respective matrix transfer strictly proper and the matrix in infinite.

Example

$>>$`[G]=mtf(3)`

$>>$`[Gsp Ginf]=mtfsp(G)`

$$G = \begin{bmatrix} \frac{1}{s+1} & \frac{1}{s^2+3s+2} & \frac{s}{s+1} \\ 0 & \frac{1}{s^2+4s+3} & \frac{s+1}{s+3} \end{bmatrix} \tag{A.1}$$

$$G_{sp} = \begin{bmatrix} \frac{1}{s+1} & \frac{1}{s^2+3s+2} & \frac{-1}{s+1} \\ 0 & \frac{1}{s^2+4s+3} & \frac{-2}{s+3} \end{bmatrix} \quad G_{inf} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.2}$$

A.5.2. mresidue

Sintax

```
[rp k]=mresidue(G)
```

Description

`mresidue` achieve partial fraction expansion of matrix transfer function. The matrix k is the same G_{inf} i.e.(A.5.1). rp is a hypermatrix with the residues and the pole in the last column. When the poles are multiple, you begins with that of smaller order toward that of more order.

Example

`>>[G]=mtf(12)`

`>>[rp k]=mresidue(G)`

$$G = \begin{bmatrix} \frac{2}{s^2-2s+1} & \frac{1}{s-1} \\ \frac{-6}{s^2+2s-3} & \frac{1}{s+3} \end{bmatrix} = \frac{\begin{bmatrix} 0 & 0 \\ 3/2 & 1 \end{bmatrix}}{(s+3)} + \frac{\begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}}{(s-1)} + \frac{\begin{bmatrix} 1 & 1 \\ -3/2 & 0 \end{bmatrix}}{(s-1)^2} \tag{A.3}$$

A.5.3. mindeg

Sintax

```
k=mindeg(sys)
```

Description

`mindeg` Find the system order. *sys* can be in state-space or matriz transfer function.

Example

`>>[G]=mtf(12)`

`>>k=mindeg(G)`

k=3

A.5.4. mtf

Sintax

```
G=mtf(n)
```

Description

`mtf` Load a Matrix transfer function that is specified with n. You can annex new MTF in the last part. Remember to change variable z with the total MTF see(A.5.2).

A.5.5. rga

Sintax

`A=rga(G)`[1]

A.5.6. polezero

Sintax

`[p z]=polezero(G)`

Description

`polezero` return the poles and zeros of a system MIMO.

Example

```
>>[p z]=mtf(72)
>>[p z]=polezero(G)
```

$$G = \begin{bmatrix} \frac{s}{s+2} & 0 \\ 0 & \frac{s+2}{s} \end{bmatrix} \quad p = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \quad z = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \tag{A.4}$$

[1] Author: Oskar Vivero[4]

A.5.7. smform

Sintax

`S=smform(G)`[2]

A.5.8. coXm

Sintax

`k=coXm(A,B,C)`

Description

`coXm` check the observability and controlability for mode for a system in space-state format. The input system should be in Jordan form see(A.5.39). The matrix k has 3 columns, the first column goes the poles, the second column if it is 1 it means that the pole associated to the line is controllable otherwise the poles is uncontrolable. The third column if it is 1 it means that the pole associated to the line is observable otherwise the pole is unobservable.

Example

>>`[A B C D]=state(27)`

>>`k=coXm(A,B,C)`

$$k = \begin{bmatrix} -2 & 1 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{A.5}$$

[2] Author: Oskar Vivero[4]

A.5.9. polindex

Sintax

```
[polo n ni]=polindex(Aj)
```

Description

`polindex` find the index of polynomial characteristic and the index of polynomial minimal. The input matrix A should be in Jordan form see(A.5.39). n this associated with the indexes of the characteristic polynomial and the ni variable this associated with the indexes of the minimal polynomial.

Example

```
>>[Aj B C D]=state(27)
>>[polo n ni]=polindex(Aj)
```

$$polo = \begin{bmatrix} -2 \\ -1 \end{bmatrix} \quad n = \begin{bmatrix} 2 \\ 5 \end{bmatrix} \quad ni = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \tag{A.6}$$

$$pc = (\lambda + 2)^2(\lambda + 1)^5 \quad pm = (\lambda + 2)^2(\lambda + 1)^2 \tag{A.7}$$

A.5.10. state

Sintax

```
[A B C D]=state(n)
```

Description

`state` Load a System in state-space that is specified with n. You can annex new SS in the last part. Remember to change case z with the total SS see(A.5.8).

A.5.11. cx2rJ

Sintax

```
[A B C P]=cx2rJ(A,B,C)
```

Description

`cx2rJ` find a new system in real matrices of a system is Jordan form with poles complex conjugated. The input matrix A should be in Jordan form and the see(A.5.39) and and the conjugated complex poles one of another should be. The matrix P is the transformation matrix.

Example

```
>>[A B C D]=state(43)
>>[Ar Br Cr P]=cx2rJ(A,B,C)
```

$$A = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1+2j & 1 & 0 & 0 \\ 0 & 0 & 0 & 1+2j & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2j & 1 \\ 0 & 0 & 0 & 0 & 0 & 1-2j \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 2 \\ 2-3j \\ 1 \\ 2+3j \\ 1 \end{bmatrix} \tag{A.8}$$

$$C = \begin{bmatrix} 1 & 2 & 1 & -j & 1 & j \end{bmatrix} \quad D = 0 \tag{A.9}$$

$$A_r = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & -2 & 0 & 1 & 1 \\ 0 & 0 & 0 & -2 & 0 & 1 \end{bmatrix} \quad B_r = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 2 \\ 6 \\ 2 \end{bmatrix} \tag{A.10}$$

$$C_r = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & -1 \end{bmatrix} \quad D = 0 \tag{A.11}$$

A.5.12. canonform

Sintax

```
[v Q Af Bf Cf]=canonform(A,B,C,nf)
```

Description

`canonform` find a new system in canonical form controlability if $9 < nf < 16$ and observabilty if $1 < nf < 8$. The input system should be controllable to find controllable canonical form the same case of observable canonical form. v is the size of blocks in the canonical form and Q is the transformation matrix.

Example

```
>>[A B C D]=state(6)
>>[vf Q Af Bf Cf]=canonform(A,B,C,2)
```

```
>>[vg Q Ag Bg Cg]=canonform(A,B,C,10)
```

$$A_f = \begin{bmatrix} -3 & -2 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad B_f = \begin{bmatrix} -1 & -2 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \tag{A.12}$$

$$C_f = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad v_f = [2 \quad 1] \tag{A.13}$$

$$A_g = \begin{bmatrix} -2 & 1 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & -2 \end{bmatrix} \quad B_g = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.14}$$

$$C_g = \begin{bmatrix} -1 & 1 & 1 \\ -1 & 1 & 0 \end{bmatrix} \quad v_g = [2 \quad 1] \tag{A.15}$$

A.5.13. findv

Sintax

```
v=findv(A,op)
```

Description

`findv` find the number blocks in matrix jordan with $op =' jordan'$, also find the indices of matrix polinomial fraction with $op =' cf'$.

Example 1

```
>>[Aj B C D]=state(6)
>>vj=findv(Aj,'jordan')
```

$$A_j = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix} \quad v_j = [2 \quad 1] \tag{A.16}$$

Example 2

```
>>[P Q]=lmpf(1)
>>[P Q]=cell2sym(P,Q)
>>vc=findv(P,'cf')
```

$$P = \begin{bmatrix} D^2 - 2D & 1 \\ D - 2 & D \end{bmatrix} \quad v_c = [2 \quad 1] \tag{A.17}$$

A.5.14. iscolred

Sintax

```
i=iscolred(M)
```

Description

`iscolred` determine if a matrix polinomial fraction M is of column reduced. $i = 1$ indicate M is column reduced.

Example

```
>>[P Q]=lmpf(1)
>>i=iscolred(P)
```

>> i=1

A.5.15. isrowred

Sintax

```
i=isrowred(M)
```

Description

`isrowred` determine if a matrix polinomial fraction M is of row reduced. $i = 1$ indicate M is row reduced.

Example

>>`[P Q]=lmpf(1)`

>>`i=isrowred(P)`

>> i=1

A.5.16. lmpf

Sintax

```
[P Q]=lmpf(n)
```

Description

`lmpf` Load a Matrix polynomial fraction that is specified with n. You can annex new LMPF in the last part. Remember to change variable z with the total LMPF.

A.5.17. cell2sym

Sintax

```
[P Q]=cell2sym(P,Q)
```

Description

`cell2sym` change matrix in cell format to matrix in symbolic format. The inputs variables should always be similar to the outputs variables.

Example

```
>>[P Q]=lmpf(1)
>>[Q]=cell2sym(Q)
```

$$Q = \left\{ \begin{array}{cc} \begin{bmatrix} 1 & 1 \end{bmatrix} & 4 \\ \begin{bmatrix} -1 & -1 \end{bmatrix} & -3 \end{array} \right\} \Rightarrow Q = \begin{bmatrix} D+1 & 4 \\ -D-1 & -3 \end{bmatrix} \tag{A.18}$$

A.5.18. sym2cell

Sintax

```
[P Q]=sym2cell(P,Q)
```

Description

`sym2cell` change matrix in simbolic format to matrix in cell format. The inputs variables should always be similar to the outputs variables.

Example

```
>>syms D Q
>>Q(1,1)=D+1
>>Q(2,1)=-D-1
>>Q(1,2)=4
>>Q(2,2)=-3
>>Q=sym2cell(Q)
```

$$Q = \begin{bmatrix} D+1 & 4 \\ -D-1 & -3 \end{bmatrix} \Rightarrow Q = \left\{ \begin{array}{cc} \begin{bmatrix} 1 & 1 \end{bmatrix} & 4 \\ \begin{bmatrix} -1 & -1 \end{bmatrix} & -3 \end{array} \right\} \tag{A.19}$$

A.5.19. normcell

Sintax

```
[M]=normcell(M,op)
```

Description

`normcell` normalize cell for rows if $op =' r'$, for columns if $op =' c'$, for rows and columns if $op =' rc'$ and a especific value if $op = n$ where n is the especific value.

Example 1

```
>>Q{1,1}=[1 2 3 4]
>>Q{1,2}=[1 -1]
>>Q{2,1}=[2 3]
>>Q{2,2}=-3
```

>>Q=normcell(Q,'c')

$$Q = \left\{ \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & -3 \end{bmatrix} \right\} \tag{A.20}$$

Example 2

>>Q=normcell(Q,8)

$$Q = \left\{ \begin{array}{cc} 1 \times 8double & 1 \times 8double \\ 1 \times 8double & 1 \times 8double \end{array} \right\} \tag{A.21}$$

A.5.20. cellround

Sintax

```
[M]=cellround(M)
```

Description

`cellround` round to nearest integer in each cell with two values in point flotating. The inputs variables should always be similar to the outputs variables.

Example

>>Q$\{1,1\}$=[1e-6 2.00005 0.654]

>>Q$\{2,1\}$=[2 3.05]

>>Q$\{1,2\}$=0;

>>Q$\{2,2\}$=-3

>>Q=cellround(Q)

$$Q = \left\{ \begin{array}{cc} \begin{bmatrix} 0 & 2 & 0.65 \end{bmatrix} & 0 \\ \begin{bmatrix} 2 & 3.05 \end{bmatrix} & -3 \end{array} \right\} \tag{A.22}$$

A.5.21. ss2sym

Sintax

```
g = ss2sym(a,b,c,d)
```
[3]

A.5.22. mpfred

Sintax

```
[P Q]=mpfred(P,Q)
```

Description

`mpfred` search the left matrix polynomial proper. If matrix P is diagonal, the system is proper and $mpfred$ return the same matrix. The inputs can be in cell or sym format.

Example

```
>>[P Q]=lmpf(9)
>>[Pm Qm]=mpfred(P,Q)
```

$$P = \begin{bmatrix} D^2+2D-3 & D^2-1 \\ D^2+8D+12 & D^2+5D+4 \end{bmatrix} \quad Q = \begin{bmatrix} 1 & 4 \\ -4 & -5 \end{bmatrix} \tag{A.23}$$

$$P_m = \begin{bmatrix} D^2-D & 0 \\ 6/5D+3 & D+1 \end{bmatrix} \quad Q_m = \begin{bmatrix} -5D & -9D-11 \\ -1 & -9/5 \end{bmatrix} \tag{A.24}$$

A.5.23. islmpfc

Sintax

```
[i pnc]=islmpfc(P,Q)
```

[3] Author: Oskar Vivero[4]

Description

`islmpfc` determine if the left matrix polinomial fraction is controlable. If lmpf not is controlable in the *pnc* variable gives the poles uncontrolables and the $i = 0$.

Example

(see A.23) >>`[P Q]=lmpf(9)`

>>`i=islmpfc(P,Q)` >>i=1

A.5.24. isrmpfo

Sintax

```
[i pno]=isrmpfo(P,Q)
```

Description

`isrmpfo` determine if the right matrix polinomial fraction is observable. If rmpf not is observable in the *pno* variable gives the poles unobservables and the $i = 0$.

Example

(see A.23) >>`[P Q]=lmpf(9)`

>>`i=isrmpfo(P,Q)` >>i=1

A.5.25. isctrb

Sintax

```
i=isctrb(A,B)
```

Description

`isctrb` determine if the state-space is controlable. If (A, B) not is controlable $i = 0$.

Example

(see A.5.37) >>`G=mtf(15)`

>>`[A B C D]=gilbertform(G)` >>`i=isctrb(A,B)`

>>i=0

A.5.26. uplowM

Sintax

`[Dhc Dlc]=uplowM(M)`

Description

`uplowM` determine in the matrix Dhc the coefficients of high orden and the matrix Dlc the resulting matrix removing the coefficients of high order. The input matrix M should be in cell format, likewise the output matrices.

Example

(see A.23) >>`[P Q]=lmpf(9)`

>>`[Dhc Dlc]=uplowM(P)`

>>`Dlc=cell2sym(Dlc)`

$$Dhc = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad Dlc = \begin{bmatrix} 2D-3 & -1 \\ 8D+12 & 5D+4 \end{bmatrix} \tag{A.25}$$

A.5.27. mSilvester

Sintax

```
[S dep lidep]=mSilvester(D,N,op,mode)
```

Description

`mSilvester` determine the matrix Silvester S an ascending ($mode =' ascend'$) or descending ($mode =' descend'$) form. In the input variable $op = 1$ especified if the system MPF is Right or $op = 0$ is left MPF. The output variable dep indicates which of the columns(if mode='ascend') or rows(if mode='descend') are linearly dependent, also in the variable $lidep$ indicates the quantity of rows of columns are linearly independent of N. The sum of $lidep$ is the order system [2].

Example 1

(see A.23) >>`[P Q]=lmpf(9)`

>>`[S cdep clidep]=mSilvester(P,Q,0,'ascend')`

$$S = \begin{bmatrix} -3 & -1 & 1 & 4 & 0 & 0 & 0 & 0 \\ 12 & 4 & -4 & -5 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & -3 & -1 & 1 & 4 \\ 8 & 5 & 0 & 0 & 12 & 4 & -4 & -5 \\ 1 & 1 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 8 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad cdep = \begin{bmatrix} 8 & 2 \end{bmatrix} \quad clidep = \begin{bmatrix} 2 & 1 \end{bmatrix} \tag{A.26}$$

Example 2

(see A.23) >>`[P Q]=lmpf(9)`

>>`[S rdep rlidep]=mSilvester(P,Q,0,'descend')`

$$S = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 8 & 5 & 0 & 0 & 1 & 1 & 0 & 0 \\ -3 & -1 & 1 & 4 & 2 & 0 & 0 & 0 \\ 12 & 4 & -4 & -5 & 8 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3 & -1 & 1 & 4 \\ 0 & 0 & 0 & 0 & 12 & 4 & -4 & -5 \end{bmatrix} \quad rdep = \begin{bmatrix} 7 & 1 \end{bmatrix} \quad rlidep = \begin{bmatrix} 1 & 2 \end{bmatrix} \tag{A.27}$$

A.5.28. ss2mtf

Sintax

`[G]=ss2mtf(A,B,C,D)`

Description

`ss2mtf` converts state-space system to matrix transfer function.

Example 1

(see A.12) >>`[A B C D]=state(6)`

>>`G=ss2mtf(A,B,C,D)`

$$G = \begin{bmatrix} \frac{s^2+3s+2}{s^3+4s^2+5s+2} & \frac{1}{s+2} \\ \frac{1}{s+1} & 0 \end{bmatrix} \tag{A.28}$$

A.5.29. ss2lmpf

Sintax

```
[P Q]=ss2lmpf(sys)
```

Description

`ss2lmpf` converts state-space system to left matrix polynomial Fraction. The input system SS should be observable.

Example

(see A.12) >>`[A B C D]=state(6)`

>>`sys=ss(A,B,C,D)`

>>`[P Q]=ss2lmpf(sys)`

$$P = \begin{bmatrix} D^2+3D+2 & -1 \\ 0 & D+1 \end{bmatrix} \quad Q = \begin{bmatrix} D+2 & D+1 \\ 1 & 0 \end{bmatrix} \tag{A.29}$$

A.5.30. ss2rmpf

Sintax

```
[P Q]=ss2rmpf(sys)
```

Description

`ss2rmpf` converts state-space system to right matrix polynomial Fraction. The input system SS should be controlable.

Example

(see A.12) >>`[A B C D]=state(6)`

>>`sys=ss(A,B,C,D)`

>>`[P Q]=ss2rmpf(sys)`

$$P = \begin{bmatrix} D^2+2D+1 & 0 \\ -1 & D+2 \end{bmatrix} \quad Q = \begin{bmatrix} D+1 & 1 \\ D+1 & 0 \end{bmatrix} \tag{A.30}$$

A.5.31. mtf2lmpf

Sintax

```
[P Q]=mtf2lmpf(G)
```

Description

`mtf2lmpf` matrix transfer function to left matrix polynomial fraction. The marix output P is always diagonal.

Example

(see A.28) >>`G1=tf([1 3 2],[1 4 5 2])`

>>`G2=tf(1,[1 2])`

>>`G3=tf(1,[1 2])`

>>`Gt=[G1 G2;G3 0])`

>>`[P Q]=mtf2lmpf(Gt)`

>>`[P Q]=cell2sym(P,Q)`

$$P = \begin{bmatrix} D^3+4D^2+5D+2 & 0 \\ 0 & D+1 \end{bmatrix} \quad Q = \begin{bmatrix} D^2+3D+3 & 2D+1+D^2 \\ 1 & 0 \end{bmatrix} \tag{A.31}$$

A.5.32. mtf2rmpf

Sintax

```
[P Q]=mtf2rmpf(G)
```

Description

`mtf2rmpf` matrix transfer function to right matrix polynomial fraction. The marix output P is always diagonal.

Example

(see A.28) >>`G1=tf([1 3 2],[1 4 5 2])`

>>`G2=tf(1,[1 2])`

>>`G3=tf(1,[1 2])`

>>`Gt=[G1 G2;G3 0])`

>>`[P Q]=mtf2rmpf(Gt)`

>>`[P Q]=cell2sym(P,Q)`

$$P = \begin{bmatrix} D^3+4D^2+5D+2 & 0 \\ 0 & D+2 \end{bmatrix} \quad Q = \begin{bmatrix} D^2+3D+3 & 1 \\ D^2+3D+2 & 0 \end{bmatrix} \tag{A.32}$$

A.5.33. mtf2lcf

Sintax

```
[P Q]=mtf2lcf(G)
```

Description

`mtf2rmpf` matrix transfer function to left coprime fraction.

Example

(see A.28) >>`G1=tf([1 3 2],[1 4 5 2])`

>>`G2=tf(1,[1 2])`

>>`G3=tf(1,[1 2])`

>>`Gt=[G1 G2;G3 0])`

>>`[P Q]=mtf2lcf(Gt)`

$$P = \begin{bmatrix} D^2+3D+2 & -1 \\ 0 & D+1 \end{bmatrix} \quad Q = \begin{bmatrix} D+2 & D+1 \\ 1 & 0 \end{bmatrix} \tag{A.33}$$

A.5.34. mtf2rcf

Sintax

```
[P Q]=mtf2rmpf(G)
```

Description

`mtf2rcf` matrix transfer function to right coprime fraction.

Example

(see A.28) >>`G1=tf([1 3 2],[1 4 5 2])`

>>`G2=tf(1,[1 2])`

>>`G3=tf(1,[1 2])`

>>`Gt=[G1 G2;G3 0])`

>>`[P Q]=mtf2rcf(Gt)`

$$P = \begin{bmatrix} D^2+2D+1 & 0 \\ -1 & D+2 \end{bmatrix} \quad Q = \begin{bmatrix} D+1 & 1 \\ D+1 & 0 \end{bmatrix} \tag{A.34}$$

A.5.35. rcf2mtf

Sintax

```
G=rcf2mtf(P,Q)
```

Description

`rcf2mtf` right coprime fraction or right matrix polynomial fraction to matrix transfer function.

Example

(see A.34) >>`G=rcf2mtf(P,Q)`

$$P = \begin{bmatrix} D^2+2D+1 & 0 \\ -1 & D+2 \end{bmatrix} \quad Q = \begin{bmatrix} D+1 & 1 \\ D+1 & 0 \end{bmatrix} \Rightarrow G = \begin{bmatrix} \frac{s^2+3s+2}{s^3+4s^2+5s+2} & \frac{1}{s+2} \\ \frac{1}{s+1} & 0 \end{bmatrix} \tag{A.35}$$

A.5.36. lcf2mtf

Sintax

```
G=lcf2mtf(P,Q)
```

Description

`lcf2mtf` left coprime fraction or left matrix polynomial fraction to matrix transfer function.

Example

(see A.31) >>`G=lcf2mtf(P,Q)`

$$P = \begin{bmatrix} D^3+4D^2+5D+2 & 0 \\ 0 & D+1 \end{bmatrix} \quad Q = \begin{bmatrix} D^2+3D+3 & 2D+1+D^2 \\ 1 & 0 \end{bmatrix} \Rightarrow \tag{A.36}$$

$$G = \begin{bmatrix} \frac{s^2+3s+2}{s^3+4s^2+5s+2} & \frac{1}{s+2} \\ \frac{1}{s+1} & 0 \end{bmatrix} \tag{A.37}$$

A.5.37. gilbertform

Sintax

```
[A B C D]=gilbertform(G)
```

Description

`gilbertform` realization by method gilbert form of a LTI MIMO sys model.

Example

>>`[G]=mtf(15)`

>>`[A B C D]=gilbertform(G)`

$$G = \begin{bmatrix} \frac{s+2}{s+1} & \frac{1}{s+3} \\ \frac{s}{s+1} & \frac{1}{s+2} \end{bmatrix} \Rightarrow A = \begin{bmatrix} -3 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{A.38}$$

$$C = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \tag{A.39}$$

A.5.38. hoform

Sintax

```
[A B C D]=hoform(sys,op)
```

Description

`hoform` realization by method Ho form of a LTI MIMO sys model. If $op = 1$ the realization is controllable and $op = 0$ the realization is observable. sys can be in space-state or matrix transfer function.

Example 1

```
>>[G]=mtf(15)
>>[A B C D]=hoform(G,0)
```

$$G = \begin{bmatrix} \frac{s+2}{s+1} & \frac{1}{s+3} \\ \frac{s}{s+1} & \frac{1}{s+2} \end{bmatrix} \Rightarrow A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -6 & 0 & -11 & 0 & -6 & 0 \\ 0 & -6 & 0 & -11 & 0 & -6 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ -1 & -3 \\ 1 & -2 \\ 1 & 9 \\ -1 & 4 \end{bmatrix} \tag{A.40}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \tag{A.41}$$

Example 2

```
>>[G]=mtf(15)
```

```
>>[A B C D]=hoform(G,1)
```

$$G = \begin{bmatrix} \frac{s+2}{s+1} & \frac{1}{s+3} \\ \frac{s}{s+1} & \frac{1}{s+2} \end{bmatrix} \Rightarrow A = \begin{bmatrix} 0 & 0 & 0 & 0 & -6 & 0 \\ 0 & 0 & 0 & 0 & 0 & -6 \\ 1 & 0 & 0 & 0 & -11 & 0 \\ 0 & 1 & 0 & 0 & 0 & -11 \\ 0 & 0 & 1 & 0 & -6 & 0 \\ 0 & 0 & 0 & 1 & 0 & -6 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{A.42}$$

$$C = \begin{bmatrix} 1 & 1 & -1 & -3 & 1 & 9 \\ -1 & 1 & 1 & -2 & -1 & 4 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \tag{A.43}$$

A.5.39. jordanform

Sintax

```
[A B C D]=jordanform(sys)
```

Description

`jordanform` realization in Jordan form [1] of a LTI MIMO sys model. *sys* can be in space-state or matrix transfer function.

Example

```
>>[G]=mtf(15)
>>[A B C D]=jordanform(G)
```

$$G = \begin{bmatrix} \frac{s+2}{s+1} & \frac{1}{s+3} \\ \frac{s}{s+1} & \frac{1}{s+2} \end{bmatrix} \Rightarrow A = \begin{bmatrix} -3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{A.44}$$

$$C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \tag{A.45}$$

A.5.40. wolovichform

Sintax

```
[A B C D]=wolovichform(G)
```

Description

`wolovichform` realization by method Wolovich form of a LTI MIMO sys model.

Example

```
>>[G]=mtf(15)
>>[A B C D]=wolovichform(G)
```

$$G = \begin{bmatrix} \frac{s+2}{s+1} & \frac{1}{s+3} \\ \frac{s}{s+1} & \frac{1}{s+2} \end{bmatrix} \Rightarrow A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -6 & -5 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.46}$$

$$C = \begin{bmatrix} 1 & 2 & 1 \\ -1 & 3 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \tag{A.47}$$

A.5.41. lmpf2ss

Sintax

```
[A B C D]=lmpf2ss(P,Q)
```

Description

`lmpf2ss` convert a left matrix polynomial fraction to state-space by method fraction coprime.

Example

(see A.23) >>`[P Q]=lmpf(9)`

>>`[A B C D]=lmpf2ss(P,Q)`

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & -11 \\ -5 & -9 \\ 0 & 33 \\ 16 & 42 \\ 5 & 9 \end{bmatrix} \tag{A.48}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{A.49}$$

A.5.42. rmpf2ss

Sintax

```
[A B C D]=rmpf2ss(P,Q)
```

Description

`rmpf2ss` convert a right matrix polynomial fraction to state-space by method fraction coprime. Although the algorithm is different, the result is similar applying `wolovichform`.

Example

```
>>G=mtf(15)
>>[P Q]=mtf2rmpf(G)
>>[A B C D]=rmpf2ss(P,Q)
```

$$A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -6 & -5 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.50}$$

$$C = \begin{bmatrix} 1 & 2 & 1 \\ -1 & 3 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \tag{A.51}$$

A.5.43. minhoform

Sintax

```
[A B C D]=minhoform(sys)
```

Description

`minhoform` minimal realization by method Ho-Kalman [6] form of a LTI MIMO sys model.

Example

```
>>[G]=mtf(15)
>>[A B C D]=minhoform(G)
```

$$G = \begin{bmatrix} \frac{s+2}{s+1} & \frac{1}{s+3} \\ \frac{s}{s+1} & \frac{1}{s+2} \end{bmatrix} \Rightarrow A = \begin{bmatrix} -2.9503 & -0.0872 & -0.2287 \\ 0.8670 & -1.4414 & 0.9880 \\ -0.5401 & 0.2711 & -1.6083 \end{bmatrix} \quad B = \begin{bmatrix} -0.1027 & -1.0679 \\ 0.9485 & -0.9875 \\ 0.5139 & 0.6033 \end{bmatrix} \tag{A.52}$$

$$C = \begin{bmatrix} -0.9905 & 0.5294 & 0.7709 \\ -0.4639 & -0.8255 & -0.5150 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \tag{A.53}$$

A.5.44. minjordanform

Sintax

```
[A B C D]=minjordanform(sys)
```

Description

`minjordanform` minimal realization in Jordan format [1] [7] form of a LTI MIMO sys model.

Example

```
>>[G]=mtf(67)
>>[A B C D]=minjordanform(G)
```

$$G = \frac{1}{(s+1)^3(s+2)} \tag{A.54}$$

$$\begin{bmatrix} (s+1)(s+2)^3 & (s+2)^2(s+2)(2s+3) & (s+2)^3 \\ (s+1)(s+2)(2s^2+5s+4) & (s+1)^2(s+2)^2 & (s+2)^2(s^2+2s+2) \\ (s+1)^2(s+2)^2 & (s+1)^3(s+2) & (s+1)(s+2)(2s+3) \\ (s+1)^2(s+2)^2 & (s+1)^3 & (s+1)(2s^2+6s+5) \end{bmatrix} \tag{A.55}$$

$$A = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 1 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.56}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 2 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{A.57}$$

A.5.45. silvermanform

Sintax

```
[A B C D]=silvermanform(sys)
```

Description

`silvermanform` minimal realization by method Silverman form [6] form of a LTI MIMO sys model.

Example

see (A.54) >>`[G]=mtf(67)`

>>`[A B C D]=silvermanform(G)`

$$A = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0.5 & -1.5 & 0.5 & 0 \\ -0.5 & 0.5 & -0.5 & 0 \\ 0 & 0 & 1 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 1 & 2 \end{bmatrix} \tag{A.58}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{A.59}$$

A.5.46. lcfform

Sintax

`[A B C D]=lcfform(sys)`

Description

`lcfform` minimal realization by method coprime fraction [3]. At this moment it is restricted square matrix transfer function.

Example

```
>>[G]=mtf(12)
```

```
>>[A B C D]=lcfform(G)
```

$$G = \begin{bmatrix} \frac{2}{s^2-2s+1} & \frac{1}{s-1} \\ \frac{-6}{s^2+2s-3} & \frac{1}{s+3} \end{bmatrix} \Rightarrow A = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 2 & 0 \\ 0 & 24 & -3 \end{bmatrix} \quad B = \begin{bmatrix} 0 & -1 \\ 1 & 1 \\ 6 & 7 \end{bmatrix} \tag{A.60}$$

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -6 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{A.61}$$

A.5.47. rcfform

Sintax

```
[A B C D]=rcfform(sys)
```

Description

`rcfform` minimal realization by method coprime fraction [3]. At this moment it is restricted square matrix transfer function.

Example

>>`[G]=mtf(12)`

>>`[A B C D]=rcfform(G)`

$$G = \begin{bmatrix} \frac{2}{s^2-2s+1} & \frac{1}{s-1} \\ \frac{-6}{s^2+2s-3} & \frac{1}{s+3} \end{bmatrix} \Rightarrow A = \begin{bmatrix} 0 & 1 & 0 \\ 3 & -2 & -0.57 \\ 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 1.7544 & 1 \\ -1.7544 & 0 \end{bmatrix} \tag{A.62}$$

$$C = \begin{bmatrix} 3 & 1 & 0.43 \\ -1 & 1 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{A.63}$$

A.5.48. gauthierform

Sintax

```
[A B C D]=gauthierform(P,Q)
```

Description

`gauthierform` minimal realization by method gauthier [5]. The P, Q matrices are left coprime fraction model.

Example

>>`[P Q]=lmpf(1)`

>>`[A B C D]=gauthierform(P,Q)`

$$\begin{bmatrix} D^2-2D & 1 \\ D-2 & D \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} D+1 & 4 \\ -D-1 & -3 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \tag{A.64}$$

$$A = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 2 & 0 \\ 0 & 2 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 4 \\ 1 & 0 \\ -1 & -3 \end{bmatrix} \tag{A.65}$$

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} \tag{A.66}$$

A.5.49. Mult&T

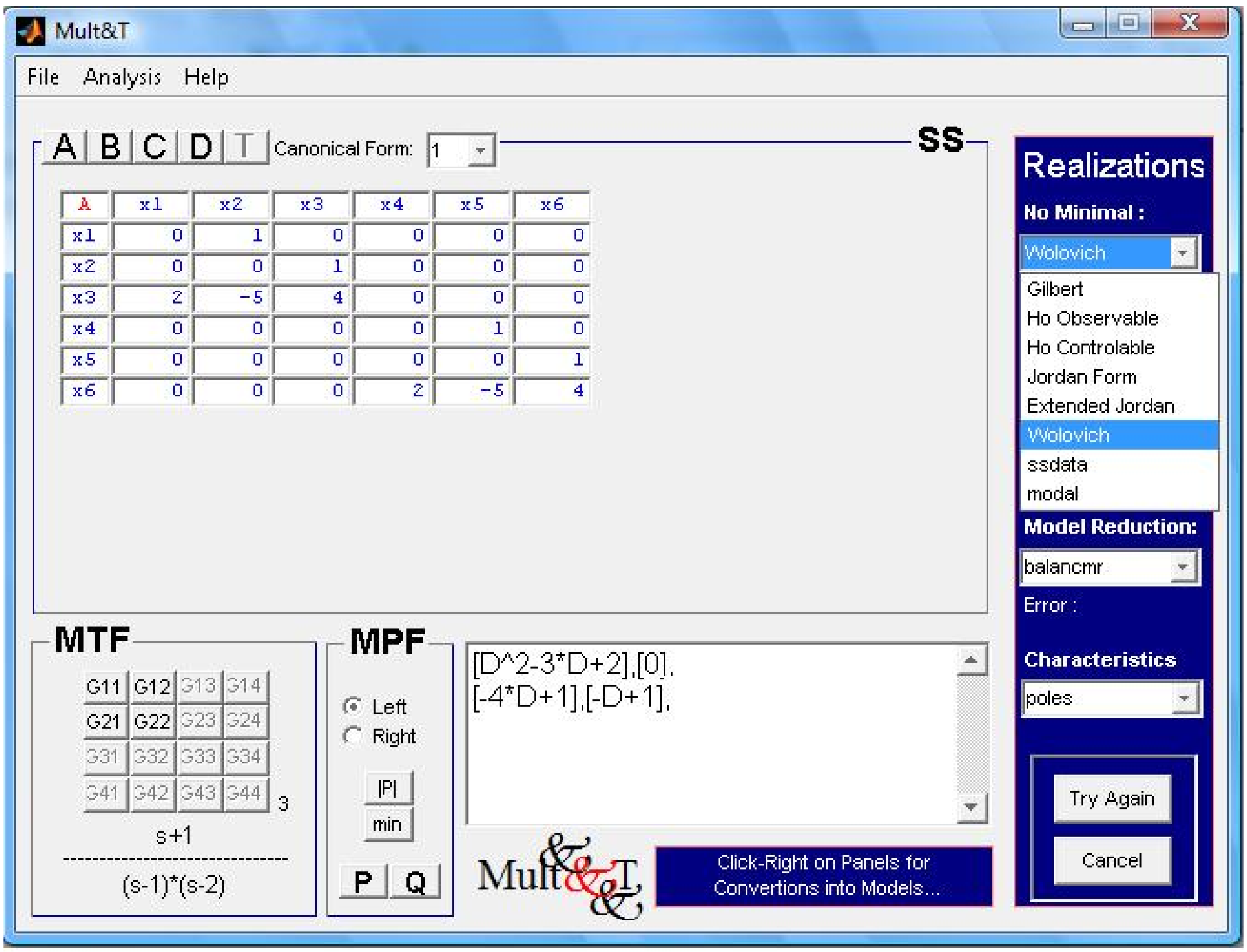

Figure A.1: Mult&T

Sintax

```
multitool
```

Example

```
>>multitool
```

Description

`multitool` initialize Multi Design Tool is a graphical-user interface (GUI) that allow the handling of multivarible models *(SS,MTF,MPF)* also conversions and characteristic special of the models, minimum and not minimum realizations, similarity transformations, balanced realizations and order reductions in the systems MIMO.

A.5.50. decopK

Sintax

```
[sysc k H]=decopK(sys)
```

Description

`decopK` Desacoupling by state feedback.

Example

```
>>[A B C D]=state(47)
>>sys=ss(A,B,C,D)
>>[sysc K H]=decopK(sys)
```

>>Gc=ss2mtf(sysc)

$$K = \begin{bmatrix} 0 & -1 & 0 \\ 6 & 11 & 5 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \hat{G}_c = \begin{bmatrix} \frac{1}{s} & 0 \\ 0 & \frac{1}{s} \end{bmatrix} \tag{A.67}$$

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 \\ 6 & 11 & 6 \\ -6 & -11 & -6 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} r \tag{A.68}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} x \tag{A.69}$$

A.5.51. canonK

Sintax

```
[sysc k kb]=canonK(sys)
```

Description

`canonK` Canonical form method design state feedback.

Example

```
>>[A B C D]=state(47)
>>sys=ss(A,B,C,D)
>>[sysc K Kb]=canonK(sys,[-4 -3 -1])
>>p=eig(sysc.a)
```

$$K = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix} \quad K_b = \begin{bmatrix} 12 & 1 & -0.83333 \\ -6 & 0 & 1 \end{bmatrix} \quad p = [-3 \quad -4 \quad -1] \tag{A.70}$$

$$\dot{x} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 0 \\ -6 & -11 & -6 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} r \tag{A.71}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} x \tag{A.72}$$

A.5.52. lyapK

Sintax

```
[sysc k kg]=lyapK(sys,pd)
```

Description

`lyapK` Lyapunov method design state feedback.

Example

```
>>[A B C D]=state(47)
>>sys=ss(A,B,C,D)
>>[sysc K Kg]=lyapK(sys,[-0.5 -1.5 -2.5])
>>p=eig(sysc.a)
```

$$K = \begin{bmatrix} -0.7545 & -1.0595 & -0.3084 \\ -0.5693 & -0.7455 & -0.2282 \end{bmatrix} \quad K_g = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad p = [-0.5 \quad -1.5 \quad -2.5] \tag{A.73}$$

$$\dot{x} = \begin{bmatrix} 0.7545 & 2.0595 & 0.3084 \\ 0.5693 & 0.7455 & 1.2282 \\ -6 & -11 & -6 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} r \tag{A.74}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} x \tag{A.75}$$

A.5.53. cyclicK

Sintax

```
[sysc k v]=cyclicK(sys,pd)
```

Description

`cyclicK` Cyclic method design state feedback.

Example

```
>>[A B C D]=state(12)
>>sys=ss(A,B,C,D)
>>[sysc K v]=cyclicK(sys,[-4 -3 -1])
>>p=eig(sysc.a)
```

$$\dot{x} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} r \tag{A.76}$$

$$y = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} x \tag{A.77}$$

$$K = \begin{bmatrix} 0 & -6 & 1 \\ 0 & -6 & 1 \end{bmatrix} \quad v = \begin{bmatrix} 1 & 1 \end{bmatrix}' \quad p = [-3 \quad -4 \quad -1] \tag{A.78}$$

$$\dot{x} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -7 & 1 \\ 0 & -12 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} r \tag{A.79}$$

$$y = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} x \tag{A.80}$$

Bibliography

[1] Chi-Tsong Chen. *Introduction to Linear System Theory*. Holt, Rinehart and Winston,INC, 1970.

[2] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, 1999.

[3] J.C.Basilio and B.Kouvaritakis. An algorithm for coprime matrix fraction description using silvester matrices. *Oxford University, Departament of Engineering Science*, PII: S0024-3795(96)00636-2:93–106, 2008.

[4] Vivero Oskar. Matlab function and a gui. Descarga de Archivos en: `http://www.mathworks.de/matlabcentral/fileexchange/loadFile.do?objectId=10816\&objectType=FILE`, 2003.

[5] Alain Gauthier Sellier. Realizaciones. Curso de Sistemas Lineales y M'ultiples Variables, Presentaci'on, I/2008.

[6] B. De Shutter. Minimal state-space realization in linear system theory: An overview. *Jornal of Computational and Applied Mathematics, special Issue on Numerical Analysis in the 20th Century*, 121:331–354, Sep. 2000.

[7] Lofti Zadhe and Charles Deseoer. *Linear System Theory, The state space approach*. McGraw Hill, 1963.