TR R140–93
November 1993

RASP
Model Order Reduction Programs

A. Varga

Deutsche Forschungsanstalt für Luft- und Raumfahrt e.V.
Institut für Robotik & Systemdynamik
Entwurfsorientierte Regelungstechnik (Prof. G. Grübel)
D–82234 Oberpfaffenhofen

CONTENTS

1. Balancing and Model Reduction of Stable State-Space Systems

The subroutines of this chapter can be used to compute reduced order models of stable state-space systems. Both additive and relative model reduction methods are provided. All implemented methods are based on the square-root accuracy enhancing technique. For several methods, the implemented versions are also balancing-free methods. All routines can be used for computing reduced models of both continuous-time and discrete-time systems. Moreover, all routines can be used to reliably compute minimal state-space realizations from given non-minimal models.

1. Additive model reduction methods

If G and G_r are the transfer-function matrices of the original and reduced order systems, respectively, then the additive methods try to compute G_r such that the norm of the additive modelling error

$$\Delta_a \;=\; G - G_r$$

is minimized. The implemented methods are square-root versions of the Balance and Truncate (B & T) method /1/, Singular Perturbation Approximation (SPA) method /2/ and the Hankel-Norm Approximation (HNA) method /3/. For the B & T and SPA methods, balancing-free versions, based on algorithms proposed in /4/ and /5/, are also provided. All methods possess the same infinity-norm error bounds for an r-th order reduced order model G_r of an n-th order system G:

$$\sigma_{r+1} \;\leq\; \|G - G_r\|_\infty \;\leq\; 2 \sum_{i=r+1}^{n} \sigma_i \;,$$

where σ_i, $i \;=\; 1,\; 2,\; \ldots,\; n$ are the Hankel-singular values of the system. The Hankel-singular values are the positive square-roots of the eigenvalues of the product of systems gramians.

The following routines are available for additive model reduction:

RPMRIB computes a reduced order model by using the square-root version of the B & T method /6/. This subroutine can also be used for computing balanced minimal state-space realizations of continuous-time or discrete-time systems.

RPMRBT computes a reduced order model by using the square-root and balancing-free version of the B & T method /4/.

RPMRSP computes a reduced order model by using the square-root and balancing-free version of the SPA method /5/.

RPMROH computes a reduced order model by using the optimal HNA method of /3/ based on the square-root balancing method.

2. Relative model reduction methods

If G and G_r are the transfer-function matrices of the original and of the reduced order systems, respectively, then the relative methods try to compute G_r such that the norm of the relative modelling error Δ_r defined by the expression

$$G_r \;=\; G(I - \Delta_r)$$

is minimized. The implemented method is a square-root version of the Balanced Stochastic Truncation (BST) method /8/. A balancing-free version of this method, based on the algorithm proposed in /9/, is also provided. The BST method possesses the following bound for the relative error for an r-th order reduced order model G_r of an n-th order system G:

$$\sigma_{r+1} \;\leq\; \|\Delta_r\|_\infty \;\leq\; 2 \sum_{i=r+1}^{n} \frac{\sigma_i}{1 - \sigma_i} \;,$$

where σ_i, $i \;=\; 1,\; 2,\; \ldots,\; n$ are the Hankel-singular values of an all-pass phase matrix.

The following routines are available for relative model reduction:

RPMRSB computes a reduced order model by using the square-root version of the BST method /8/. This subroutine can also be used for computing stochastically balanced minimal state-space realizations of continuous-time or discrete-time systems.

RPMRST computes a reduced order model by using the square-root and balancing-free version of the BST method /9/.

For both subroutines, a parameter α can be used as a weight between the absolute and relative errors. For $\alpha \;>\; 0$, the BST method is performed on a modified system with the transfer-function matrix $[G \;\; \alpha I]$. A zero value of α means a pure relative error minimization. Large positive values of α produce approximations which minimize the absolute approximation error. When α tends to infinite, the BST method produces identical results with the B & T method.

3. Auxiliary tools

Two subroutines are provided to be used in conjunction with the model reduction subroutines for stable systems:

RPMRSR computes a reduced order model by using state residualization (singular perturbation) formulas. The reduced model has the same steady-state gain as the original system.

RPMRDC performs a multivariable two-parameters bilinear transform. This routine is mainly used for continuous-to-discrete or discrete-to-continuous mapping of linear systems.

Literature:

/1/ Moore B.C.
Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction.
IEEE Trans. Autom. Control, vol. AC-26, pp. 17-31, 1981.

/2/ Liu Y. and Anderson B.D.O.
Singular Perturbation Approximation of Balanced Systems,
Int. J. Control, Vol. 50, pp. 1379-1405, 1989.

/3/ Glover, K.
All Optimal Hankel Norm Approximation of Linear Multivariable Systems and Their L-Infinity Error Bounds,
Int. J. Control, Vol. 36, pp. 1145-1193, 1984.

/4/ Varga A.
Efficient Minimal Realization Procedure Based on Balancing.
Proc. of IMACS/IFAC Symp. MCTS, Lille, France, May 7-10, 1991, Eds. A. El Moudui, P. Borne, S.G. Tzafestas, Vol. 2, pp. 42-46.

/5/ Varga A.
Balancing-Free Square-Root Algorithm for Computing Singular Perturbation Approximations. Proc. 30-th IEEE CDC, Brighton, Dec. 11-13, 1991, Vol. 2, pp. 1062-1065.

/6/ Tombs M.S. and Postlethwaite I.
Truncated Balanced Realization of Stable, Non-minimal State-Space Systems.
Int. J. Control, Vol. 46, pp. 1319-1330, 1987.

/7/ Desai U.B. and Pal D.
A Transformation Approach to Stochastic Model Reduction.
IEEE Trans. Autom. Control, vol. AC-29, pp. 1097-1100, 1984.

/8/ Safonov M.G. and Chiang R.Y.
Model Reduction for Robust Control: a Schur Relative-Error Method.
Int. J. Adapt. Contr. & Sign. Proc., vol.2, pp.259-272, 1988.

/9/ Varga A. and Fasol K.H.
A New Square-Root Balancing-Free Stochastic Truncation Model Reduction Algorithm.
Prepr. 12th IFAC Congress, Sydney, vol. 7, pp. 153-156, 1993.

SUBROUTINE RPMRIB

Square-root Balance & Truncate Model Reduction

Procedure purpose:

Given the matrices A, B and C of an original stable model, this subroutine computes the corresponding matrices Ar, Br and Cr of a reduced order model by using the square-root balance and truncate method of /1/. The routine can be also used for computing balanced minimal state-space realizations.

Usage:

```
CALL RPMRIB(A, N, B, M, C, P, DISCR, TOL, FIXORD, NR, HSV,
            RWORK, LWORK, *)
```

```
A     : IN, OUT, DOUBLE (N,N)
        On input  : system state matrix A of the original model
                    (column dense)
        On output : the NR*NR matrix Ar of the reduced model
                    (column dense)
N     : IN, INTEGER
        dimension of state vector
B     : IN, OUT, DOUBLE (N,M)
        On input  : system input matrix B of the original model
                    (column dense)
        On output : the NR*M matrix Br of the reduced model
                    (column dense)
M     : IN, INTEGER
        dimension of input vector
C     : IN, OUT, DOUBLE (P,N)
        On input  : system output matrix C of the original model
                    (column dense)
        On output : the P*NR matrix Cr of the reduced model
                    (column dense)
P     : IN, INTEGER
        dimension of output vector
DISCR : IN, LOGICAL
        specifies the type of the system:
        continuous-time system, if DISCR = .FALSE., or
        discrete-time system, if DISCR = .TRUE.
TOL   : IN, DOUBLE
        absolute tolerance used for determining the order of the
        reduced model (if FIXORD = 0). When TOL .LE. 0, an internally
        computed default value TOL = N*EPS*HNORM(A,B,C) is used,
        where EPS is the machine precision (see the LAPACK Library
        Routine DLAMCH) and HNORM(A,B,C) is the Hankel-norm of the
        original model (computed in HSV(1)).
```

FIXORD: IN, INTEGER
desired order of the reduced model. If FIXORD > 0, the order NR is set to FIXORD. If FIXORD = 0, the order NR is chosen in accordance with the specified tolerance TOL.

NR : OUT, INTEGER
resulted order of the reduced model. If FIXORD > 0, NR is set equal to FIXORD. If FIXORD = 0, NR is set equal to the number of Hankel singular values greater than TOL.

HSV : OUT, DOUBLE (N)
Hankel singular values of the system (the square roots of eigenvalues of the product of gramians) ordered decreasingly.

RWORK : OUT, DOUBLE (LWORK)
working array.

LWORK : IN, INTEGER
dimension of working array RWORK.
The value of LWORK must be at least N*(3*N+MAX(N,M,P)+4).

* : RETURN 1, target label in case of error (e.g. *1111)

File input/ output:

none

Method:

Literature

/1/ Tombs M.S. and Postlethwaite I.
Truncated balanced realization of stable, non-minimal state-space systems, Int. J. Control, Vol. 46, pp. 1319-1330, 1987.

Remarks:

- If HSV(NR) > HSV(NR+1), the resulting reduced model (Ar,Br,Cr) is stable and minimal.
- In the continuous-time case, the reduced model is balanced.
- If G and Gr are the transfer-function matrices of the systems (A,B,C) and (Ar,Br,Cr), respectively, then the approximation error G-Gr satisfies the inequalities

  HSV(NR+1) <= INFNORM(G-Gr) <= 2*(HSV(NR+1) + ... + HSV(N)),

  where INFNORM(G) is the infinity-norm of G.
- The reduced model is computed after the system is reduced to a state coordinate form in which A is in a real Schur form. The matrices of the reduced model are then computed as

  Ar = Ti*A*T, Br = Ti*B, Cr = C*T,

  where T and Ti are N*NR and NR*N projection matrices, respectively. These matrices are available, in column dense forms, in the working array RWORK beginning with elements RWORK(1) and RWORK(N*N+1), respectively.
- The accuracy loss which can be induced in the reduced model by

applying these projections or, if T and Ti are invertible, by transforming the given system to the balanced coordinate form, can be estimated by determining the condition numbers of the projection matrices T and Ti.

Life cycle:
1992 MAY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
Given the continuous-time system (A,B,C) with the following matrices:

$$A = \begin{pmatrix} -.04165 & 0 & 4.92 & -4.92 & 0 & 0 & 0 \\ -5.21 & -12.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.33 & -3.33 & 0 & 0 & 0 & 0 \\ .545 & 0 & 0 & 0 & -.545 & 0 & 0 \\ 0 & 0 & 0 & 4.92 & -.04165 & 0 & 4.92 \\ 0 & 0 & 0 & 0 & -5.21 & -12.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.33 & -3.33 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 \\ 12.5 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 12.5 \\ 0 & 0 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

a balanced state-space realization can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .FALSE.
      TOL    = 0.0D0
      FIXORD = 0
      LWORK  = 4*(N+1)*N
      CALL RPMRIB(A, N, B, M, C, P, DISCR, TOL, FIXORD, NR, HSV,
     *            RWORK, LWORK, *1111)
```

```
 The matrices of balanced system are:

      (  -.3967  3.0279       0       0   .9557    .4547        0 )
      ( -3.0871  -.2524       0       0  -.2601   -.3651        0 )
      (       0       0  -.5849 -2.2166       0        0   -.6258 )
Ar =  (       0       0  2.2166 -1.8927       0        0  -1.8629 )
      (  -.9800  -.6079       0       0 -1.7248  -1.9937        0 )
      (   .4594   .4108       0       0  2.1655 -13.4977        0 )
      (       0       0  -.6258  1.8629       0        0 -13.3940 )

      (  -.9987     .9987 )
      (  -.7254     .7254 )
      ( -1.0591   -1.0591 )
Br =  (  1.2046    1.2046 )
      (  -.9716     .9716 )
      (   .5840    -.5840 )
      (  -.5738    -.5738 )

      ( -.9922  .6335 -1.0591 -1.2046  .8978  .5838 -.5738 )
Cr =  (  .1612  .4997       0       0 -.5253  .0224       0 )
      (  .9922 -.6335 -1.0591 -1.2046 -.8978 -.5838 -.5738 )

 The computed Hankel singular values are:

   2.5139   2.0846   1.9178   0.7666   0.5473   0.0253   0.0246
```

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
The system is not stable.
-4-
Computation of Hankel singular values failed.
-5-
Selected order larger than the order of minimal realization.
-6-
Not enough working storage. It should be at least //LENG//.
```

SUBROUTINE RPMRBT

Square-Root Balancing-Free Balance & Truncate Model Reduction

Procedure purpose:

Given the matrices A, B and C of an original stable model, this subroutine computes the corresponding matrices Ar, Br and Cr of a reduced order model by using the square-root balancing-free version of the balance and truncate (B & T) method /1/. The routine can be also used for computing minimal realizations of state-space systems.

Usage:

```
CALL RPMRBT(A, N, B, M, C, P, DISCR, TOL, FIXORD, NR, HSV, RWORK,
            LWORK, IWORK, *)
```

A : IN, OUT, DOUBLE (N,N)
On input : system state matrix A of the original model (column dense)
On output : the NR*NR matrix Ar of the reduced model (column dense)

N : IN, INTEGER
dimension of state vector

B : IN, OUT, DOUBLE (N,M)
On input : system input matrix B of the original model (column dense)
On output : the NR*M matrix Br of the reduced model (column dense)

M : IN, INTEGER
dimension of input vector

C : IN, OUT, DOUBLE (P,N)
On input : system output matrix C of the original model (column dense)
On output : the P*NR matrix Cr of the reduced model (column dense)

P : IN, INTEGER
dimension of output vector

DISCR : IN, LOGICAL
specifies the type of the system:
continuous-time system, if DISCR = .FALSE., or
discrete-time system, if DISCR = .TRUE.

TOL : IN, DOUBLE
absolute tolerance used for determining the order of the reduced model (if FIXORD = 0). When TOL .LE. 0, an internally computed default value TOL = N*EPS*HNORM(A,B,C) is used, where EPS is the machine precision (see the LAPACK Library Routine DLAMCH) and HNORM(A,B,C) is the Hankel-norm of the original model (computed in HSV(1)).

FIXORD: IN, INTEGER

desired order of the reduced model. If FIXORD > 0, the order NR is set to FIXORD. If FIXORD = 0, the order NR is chosen in accordance with the specified tolerance TOL.

NR : OUT, INTEGER
resulted order of the reduced model. If FIXORD > 0, NR is set equal to FIXORD. If FIXORD = 0, NR is set equal to the number of Hankel singular values greater than TOL.

HSV : OUT, DOUBLE (N)
Hankel singular values of the system (the square roots of eigenvalues of the product of gramians) ordered decreasingly.

RWORK : OUT, DOUBLE (LWORK)
working array.

LWORK : IN, INTEGER
dimension of working array RWORK.
The value of LWORK must be at least N*(3*N+MAX(N,M,P)+4).

IWORK : OUT, INTEGER (N)

* : RETURN 1, target label in case of error (e.g. *1111)

File input/ output:

none

Method:

Literature

/1/ Varga A.
Efficient minimal realization procedure based on balancing
Proc. of IMACS/IFAC Symp. MCTS, Lille, France, May 7-10, 1991,
Eds. A. El Moudui, P. Borne, S.G. Tzafestas, Vol. 2, pp. 42-46.

Remarks:

- If HSV(NR) > HSV(NR+1), the resulting reduced model (Ar,Br,Cr) is stable and minimal, but generally not balanced.
- If G and Gr are the transfer-function matrices of the systems (A,B,C) and (Ar,Br,Cr), respectively, then the approximation error G-Gr satisfies the inequalities

  HSV(NR+1) <= INFNORM(G-Gr) <= 2*(HSV(NR+1) + ... + HSV(N)),

  where INFNORM(G) is the infinity-norm of G.
- The reduced model is computed after the system is reduced to a state coordinate form in which A is in a real Schur form. The matrices of the reduced model are then computed as

  Ar = Ti*A*T, Br = Ti*B, Cr = C*T,

  where T and Ti are N*NR and NR*N projection matrices, respectively. These matrices are available, in column dense forms, in the working array RWORK beginning with elements RWORK(1) and RWORK(N*N+1), respectively. In the implemented version, T always results with orthonormal columns.
- The accuracy loss which can be induced in the reduced model by applying these projections can be estimated by determining the

condition number of the projection matrix Ti.

Copyright:
1992 - DLR Institut fuer Dynamik der Flugsysteme

Life cycle:
1992 MAY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
Given the continuous-time system example (A,B,C) used by the subroutine RPMRIB, with the Hankel singular values

2.5139 2.0846 1.9178 0.7666 0.5473 0.0253 0.0246 ,

a fifth order approximate model can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .FALSE.
      TOL    = 0.1D0
      FIXORD = 0
      LWORK  = 4*(N+1)*N
      CALL RPMRBT(A, N, B, M, C, P, DISCR, TOL, FIXORD, NR, HSV,
     *            RWORK, LWORK, IWORK, *1111)
```

The matrices of reduced system are:

$$Ar = \begin{pmatrix} 1.3451 & -5.0399 & 0 & 0 & -4.5315 \\ 4.0214 & -3.6604 & 0 & 0 & -.9056 \\ 0 & 0 & .5124 & -1.7910 & 0 \\ 0 & 0 & 4.2167 & -2.9900 & 0 \\ -1.2402 & 1.6416 & 0 & 0 & -.0586 \end{pmatrix}$$

$$Br = \begin{pmatrix} -.3857 & .3857 \\ 3.1753 & -3.1753 \\ .7447 & .7447 \\ -3.6872 & -3.6872 \\ -1.8197 & 1.8197 \end{pmatrix}$$

$$Cr = \begin{pmatrix} -.6704 & -.1828 & .6582 & .2222 & .0104 \\ .1089 & -.4867 & 0 & 0 & -.8651 \\ .6704 & .1828 & .6582 & .2222 & -.0104 \end{pmatrix}$$

Error Messages:

-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
The system is not stable.
-4-
Computation of Hankel singular values failed.
-5-
Selected order larger than the order of minimal realization.
-6-
Not enough working storage. It should be at least //LENG//.

SUBROUTINE RPMRSP

Square-Root Balancing-Free Singular Perturbation Approximation

Procedure purpose:

Given the matrices A, B, C and D of an original stable model, this subroutine computes the corresponding matrices Ar, Br, Cr and Dr of a reduced order model by using the square-root balancing-free version of the singular perturbation approximation (SPA) method /1/.

Usage:

```
CALL RPMRSP(A, N, B, M, C, P, D, DISCR, TOL1, TOL2, FIXORD, BAL, NR,
            HSV, RWORK, LWORK, IWORK, *)
```

A : IN, OUT, DOUBLE (N,N)
On input : system state matrix A of the original model (column dense)
On output : the NR*NR matrix Ar of the reduced model (column dense)

N : IN, INTEGER
dimension of state vector

B : IN, OUT, DOUBLE (N,M)
On input : system input matrix B of the original model (column dense)
On output : the NR*M matrix Br of the reduced model (column dense)

M : IN, INTEGER
dimension of input vector

C : IN, OUT, DOUBLE (P,N)
On input : system output matrix C of the original model (column dense)
On output : the P*NR matrix Cr of the reduced model (column dense)

P : IN, INTEGER
dimension of output vector

D : IN, OUT, DOUBLE (P,M)
On input : system feedthrough matrix D of the original model (column dense)
On output : the P*M matrix Dr of the reduced model (column dense)

DISCR : IN, LOGICAL
specifies the type of the system:
continuous-time system, if DISCR = .FALSE., or
discrete-time system, if DISCR = .TRUE.

```
TOL1   :  IN, DOUBLE
          absolute tolerance used for determining the order of the
          reduced model (if FIXORD = 0). When TOL1 .LE. 0, an internally
          computed default value TOL1 = SQRT(N*EPS)*HNORM(A,B,C) is used,
          where EPS is the machine precision (see the LAPACK Library
          Routine DLAMCH) and HNORM(A,B,C) is the Hankel-norm of the
          original model (computed in HSV(1)).
TOL2   :  IN, DOUBLE
          absolute tolerance used for determining the order of the
          minimal realization. When TOL2 .LE. 0, an internally
          computed default value TOL2 = N*EPS*HNORM(A,B,C) is used.
FIXORD:   IN, INTEGER
          desired order of the reduced model. If FIXORD > 0, the order
          NR is set to FIXORD. If FIXORD = 0, the order NR is chosen in
          accordance with the specified tolerance TOL1.
BAL    :  IN, LOGICAL, SELECTION PARAMETER
          .TRUE.  : determine a balanced reduced model
          .FALSE. : determine an unbalanced reduced model
                    (recommended option)
NR     :  OUT, INTEGER
          resulted order of the reduced model. If FIXORD > 0, NR is set
          equal to FIXORD. If FIXORD = 0, NR is set equal to the number
          of Hankel singular values greater than TOL1.
HSV    :  OUT, DOUBLE (N)
          Hankel singular values of the system (the square roots of
          eigenvalues of the product of gramians) ordered decreasingly.
RWORK  :  OUT, DOUBLE (LWORK)
          working array.
LWORK  :  IN, INTEGER
          dimension of working array RWORK.
          The value of LWORK must be at least  N*(3*N+MAX(N,M,P)+4).
IWORK  :  OUT, INTEGER (N)
          working array. On normal exit, IWORK(1) contains NMIN, the
          order of the computed minimal realization.
*      :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

Literature

/1/ Varga A.
Balancing-free square-root algorithm for computing singular perturbation approximations. Proc. 30-th IEEE CDC, Brighton, Dec. 11-13, 1991, Vol. 2, pp. 1062-1065.

Remarks:

- If HSV(NR) > HSV(NR+1), the resulting reduced model (Ar,Br,Cr,Dr) is stable, minimal and balanced if BAL = .TRUE. . It has the same steady-state gain as the original system.
- If G and Gr are the transfer-function matrices of the systems

(A,B,C,D) and (Ar,Br,Cr,Dr), respectively, then the approximation error G-Gr satisfies the inequalities

HSV(NR+1) <= INFNORM(G-Gr) <= 2*(HSV(NR+1) + ... + HSV(N)),

where INFNORM(G) is the infinity-norm of G.

- The reduced model is computed after the system is reduced to a state coordinate form in which A is in a real Schur form. The matrices of a minimal realization are computed as

Am = Ti*A*T, Bm = Ti*B, Cm = C*T , (1)

where T and Ti are N*NMIN and NMIN*N projection matrices, respectively. These matrices are available, in column dense forms, in the working array RWORK beginning with elements RWORK(1) and RWORK(N*N+1), respectively, while NMIN is contained in the element IWORK(1). The matrices of the reduced model are computed from the minimal system (Am,Bm,Cm,D) by using singular perturbation formulas.
- The accuracy loss which can be induced in the reduced model by applying the projections (1) can be estimated by determining the condition numbers of the projection matrices T and Ti.
- By choosing TOL1 = TOL2 > 0, this subroutine can be also used to compute reduced models by using either the balanced B & T method, if BAL = .TRUE. , or the balancing-free B & T method, if BAL = .FALSE..

Copyright:

Life cycle:

1992 MAY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the continuous-time system example (A,B,C,D) used by the subroutine RPMRIB (D = 0), with the Hankel singular values

2.5139 2.0846 1.9178 0.7666 0.5473 0.0253 0.0246 ,

a fifth order approximate model can be computed with the following sequence of statements:

```
N      = 7
M      = 2
P      = 3
DISCR  = .FALSE.
TOL1   = 0.1D0
TOL2   = 1.0D-14
FIXORD = 0
BAL    = .FALSE.
```

```
      LWORK  =  4*(N+1)*N
      CALL RPMRSP(A, N, B, M, C, P, D, DISCR, TOL1, TOL2, FIXORD,
     *            BAL, NR, HSV, RWORK, LWORK, IWORK, *1111)
```

The matrices of reduced system are:

$$Ar = \begin{pmatrix} 1.3960 & -5.1248 & 0 & 0 & -4.4331 \\ 4.1411 & -3.8605 & 0 & 0 & -.6738 \\ 0 & 0 & .5847 & -1.9230 & 0 \\ 0 & 0 & 4.3823 & -3.2922 & 0 \\ -1.3261 & 1.7851 & 0 & 0 & -.2249 \end{pmatrix}$$

$$Br = \begin{pmatrix} -.2901 & .2901 \\ 3.4004 & -3.4004 \\ .6379 & .6379 \\ -3.9315 & -3.9315 \\ -1.9813 & 1.9813 \end{pmatrix}$$

$$Cr = \begin{pmatrix} -.6570 & -.2053 & .6416 & .2526 & .0364 \\ .1094 & -.4875 & 0 & 0 & -.8641 \\ .6570 & .2053 & .6416 & .2526 & -.0364 \end{pmatrix}$$

$$Dr = \begin{pmatrix} .0498 & -.0007 \\ .0010 & -.0010 \\ -.0007 & .0498 \end{pmatrix}$$

Error Messages:

-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
The system is not stable.
-4-
Computation of Hankel singular values failed.
-5-
Selected order larger than the order of minimal realization.
-6-
Singular matrix in the singular perturbation formulas.
-7-
Not enough working storage. It should be at least //LENG//.

SUBROUTINE RPMROH

Optimal Hankel-Norm Approximation with Square-Root Balancing

Procedure purpose:

Given the matrices A, B, C and D of an original stable model, this subroutine computes the corresponding matrices Ar, Br, Cr and Dr of a reduced order model by using the optimal Hankel-norm approximation method /1/ in conjunction with square-root balancing /2/.

Usage:

```
CALL RPMROH(A, N, B, M, C, P, D, DISCR, TOL1, TOL2, FIXORD, NR, HSV,
            RWORK, LWORK, IWORK, *)
```

A : IN, OUT, DOUBLE (N,N)
On input : system state matrix A of the original model (column dense)
On output : the NR*NR matrix Ar of the reduced model. (column dense)

N : IN, INTEGER
dimension of state vector

B : IN, OUT, DOUBLE (N,M)
On input : system input matrix B of the original model (column dense)
On output : the NR*M matrix Br of the reduced model (column dense)

M : IN, INTEGER
dimension of input vector

C : IN, OUT, DOUBLE (P,N)
On input : system output matrix C of the original model (column dense)
On output : the P*NR matrix Cr of the reduced model (column dense)

P : IN, INTEGER
dimension of output vector

D : IN, OUT, DOUBLE (P,M)
On input : system feedthrough matrix D of the original model (column dense)
On output : the P*M matrix Dr of the reduced model (column dense)

DISCR : IN, LOGICAL
specifies the type of the system:
continuous-time system, if DISCR = .FALSE., or
discrete-time system, if DISCR = .TRUE.

TOL1 : IN, DOUBLE
absolute tolerance used for determining the order of the reduced model (if FIXORD < 0). When TOL1 .LE. 0, an internally computed default value TOL1 = SQRT(N*EPS)*HNORM(A,B,C) is used, where EPS is the machine precision (see the LAPACK Library

```
          Routine DLAMCH) and HNORM(A,B,C) is the Hankel-norm of the
          original model (computed in HSV(1)).
TOL2   :  IN, DOUBLE
          absolute tolerance used for determining the order of the
          minimal realization. When TOL2 .LE. 0, an internally
          computed default value TOL2 = N*EPS*HNORM(A,B,C) is used.
FIXORD:   IN, INTEGER
          desired order of the reduced model.  If FIXORD < 0, the order
          is chosen in accordance with the specified tolerance TOL1.
NR     :  OUT, INTEGER
          resulted order of the reduced model. If FIXORD >= 0, the order
          NR is set to max(0,FIXORD-KR+1), where KR is the multiplicity
          of Hankel singular value HSV(FIXORD+1). If FIXORD < 0, the
          order NR is set equal to the number of Hankel singular values
          greater than TOL1.
HSV    :  OUT, DOUBLE (N)
          Hankel singular values of the system (the square roots of
          eigenvalues of the product of gramians) ordered decreasingly.
RWORK  :  OUT, DOUBLE (LWORK)
          working array.
LWORK  :  IN, INTEGER
          dimension of working array RWORK.
          The value of LWORK must be at least  N*(3*N+MAX(N,M,P)+4).
IWORK  :  OUT, INTEGER (N)
          working array. On normal exit, IWORK(1) contains NMIN, the
          order of the computed minimal realization.
*      :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

Literature

/1/ Glover, K.
All optimal Hankel norm approximation of linear multivariable systems and their L-infinity error bounds,
Int. J. Control, Vol. 36, pp. 1145-1193, 1984.

/2/ Tombs M.S. and Postlethwaite I.
Truncated balanced realization of stable, non-minimal state-space systems,
Int. J. Control, Vol. 46, pp. 1319-1330, 1987.

Remarks:

- The resulting reduced model (Ar,Br,Cr,Dr) is stable and minimal.
- If G and Gr are the transfer-function matrices of the systems (A,B,C,D) and (Ar,Br,Cr,Dr), respectively, then the approximation error G-Gr satisfies the inequalities

```
HSV(NR+1) <= INFNORM(G-Gr) <= 2*( HSV(NR+1) + ... + HSV(N) ),
```

where INFNORM(G) is the infinity-norm of G. Moreover, the computed

reduced system is optimal for the Hankel-norm, that is, the approximation error G-Gr satisfies

HNORM(G-Gr) = HSV(NR+1),

where HNORM(G) is the Hankel-norm of G.

- The reduced model is computed after the system is reduced to a balanced minimal state-space realization by using the projection formulas

Am = Ti*A*T, Bm = Ti*B, Cm = C*T , (1)

where T and Ti are N*NMIN and NMIN*N projection matrices, respectively. The matrix T is available, in a column dense form, in the working array RWORK beginning with element RWORK(1), while NMIN is contained in the element IWORK(1).
- The accuracy loss which can be induced in the reduced model by applying the balancing projections (1) can be estimated by determining the condition number of the projection matrix T.
- By choosing TOL1 = TOL2 > 0, this subroutine can be also used to compute a balanced minimal realization by using the square-root B & T method.

Copyright:

Life cycle:

1992 JUNE A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the continuous-time system example (A,B,C,D) used by the subroutine RPMRIB (D = 0), with the Hankel singular values

2.5139 2.0846 1.9178 0.7666 0.5473 0.0253 0.0246 ,

a fifth order approximate model can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .FALSE.
      TOL1   = 0.1D0
      TOL2   = 1.0D-14
      FIXORD = -1
      LWORK  =  4*(N+1)*N
      CALL RPMROH(A, N, B, M, C, P, D, DISCR, TOL1, TOL2, FIXORD,
     *            NR, HSV, RWORK, LWORK, IWORK, *1111)
```

```
 The matrices of reduced system are:

      ( -.5038  -1.8355  -2.6289        0        0 )
      ( 5.3070   -.5038   3.2250        0        0 )
 Ar = (      0        0  -1.5171        0        0 )
      (      0        0        0  -1.2925   9.0718 )
      (      0        0        0   -.5047  -1.2925 )

      (  -.3614    .3614 )
      (  1.5343  -1.5343 )
 Br = ( -1.1096   1.1096 )
      ( -4.5325  -4.5325 )
      (   .7396    .7396 )

      ( -.3055  -1.8971  -2.1124    .4421   2.1023 )
 Cr = ( 1.1112    .0394   -.3119        0        0 )
      (  .3055   1.8971   2.1124    .4421   2.1023 )

      (  .0126   -.0126 )
 Dr = (  .0005   -.0005 )
      ( -.0126    .0126 )
```

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
The system is not stable.
-4-
Computation of Hankel singular values failed.
-5-
Selected order larger than the order of minimal realization.
-6-
The computation of stable projection failed.
-7-
Not enough working storage. It should be at least //LENG//.
```

SUBROUTINE RPMRSB

Square-root Balanced Stochastic Truncation Method

Procedure purpose:

Given the matrices A, B, C and D of an original stable model,
this subroutine computes the corresponding matrices Ar, Br, Cr and Dr
of a reduced order model by using the square-root version of the
balanced stochastic truncation (BST) method of /1/.
For the applicability of the BST method, the given system must have
the number of systems outputs less than or equal to the number of
systems inputs and the transfer-function matrix of the system G,
must have no zeros on the imaginary axis for a continuous-time system
or on the unit circle for a discrete-time system.
In particular, the feedthrough matrix D must have maximal row rank.
If D has maximal column rank, the BST method can be employed
on the dual system with the transfer-function matrix G'.

A parameter ALPHA can be used as a weight between the absolute and
relative errors. For ALPHA <= 0, the BST method is performed on the
original system. If ALPHA > 0, the BST method is performed on a
modified system with the transfer-function matrix (G ALPHA*I).
This is the recommended approach to be used when the conditions on
the number inputs and outputs and/or on the rank of D are not fulfilled
by a given system.

Usage:

```
CALL RPMRSB(A, N, B, M, C, P, D, DISCR, ALPHA, TOL, FIXORD,
            NR, HSV, RCOND, RWORK, LWORK, IWORK, *)
```

```
A     :  IN, OUT, DOUBLE (N,N)
         On input  : system state matrix A of the original model
                     (column dense)
         On output : the NR*NR matrix Ar of the reduced model
                     (column dense)
N     :  IN, INTEGER
         dimension of state vector
B     :  IN, OUT, DOUBLE (N,M)
         On input  : system input matrix B of the original model
                     (column dense)
         On output : the NR*M matrix Br of the reduced model
                     (column dense)
M     :  IN, INTEGER
         dimension of input vector
C     :  IN, OUT, DOUBLE (P,N)
         On input  : system output matrix C of the original model
                     (column dense)
         On output : the P*NR matrix Cr of the reduced model
                     (column dense)
```

P : IN, INTEGER
dimension of output vector

D : IN, OUT, DOUBLE (P,M)
On input : system feedthrough matrix D of the original model
(column dense)
On output : the P*M matrix Dr of the reduced model
(column dense)

DISCR : IN, LOGICAL
specifies the type of the system:
continuous-time system, if DISCR = .FALSE., or
discrete-time system, if DISCR = .TRUE.

ALPHA : IN, DOUBLE, SELECTION PARAMETER
ALPHA > 0 specifies the absolute/relative error weighting parameter. A large positive value of ALPHA favours the minimization of the absolute approximation error, while a small value of ALPHA is appropriate for the minimization of the relative error.
ALPHA <= 0 means pure relative error method and can be used only if rank(D) = P and P <= M.

TOL : IN, DOUBLE
absolute tolerance used for determining the order of the reduced model (if FIXORD = 0). When TOL .LE. 0, an internally computed default value TOL = N*EPS is used, where EPS is the machine precision (see the LAPACK Library Routine DLAMCH).

FIXORD: IN, INTEGER
desired order of the reduced model. If FIXORD > 0, the order NR is set to FIXORD. If FIXORD = 0, the order NR is chosen in accordance with the specified tolerance TOL.

NR : OUT, INTEGER
resulted order of the reduced model. If FIXORD > 0, NR is set equal to FIXORD. If FIXORD = 0, NR is set equal to the number of singular values (HSV(i), i = 1, N) greater than TOL.

HSV : OUT, DOUBLE (N)
The Hankel singular values of the systems phase matrix, ordered decreasingly. The singular values are less than or equal to 1.

RCOND : OUT, DOUBLE
indicator of conditioning of the Riccati equation satisfied by the observability Gramian. A small value of RCOND (RCOND <= 1.E-7) means possible accuracy loss due to an ill-conditioned Riccati equation. Usually, the conditioning can be improved by choosing a larger value for ALPHA.

RWORK : OUT, DOUBLE (LWORK)
working array.

LWORK : IN, INTEGER
dimension of working array RWORK.
The value of LWORK must be at least
P*MB + max(MB*P,8*N) + N*(2*MB+P+N+max(N,M,P)+4*max(P,2*N),
where MB = M+P if ALPHA > 0 and MB = M otherwise.

```
IWORK :  OUT, INTEGER (2*N)
         working array.
*     :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

Literature

/1/ Safonov M.G. and Chiang R.Y.
Model reduction for robust control: a Schur relative error method,
Int. J. Adapt. Contr. & Sign. Proc., vol.2, pp.259-272, 1988.

Remarks:

- If HSV(NR) > HSV(NR+1), the resulting reduced model (Ar,Br,Cr,Dr) is stable and minimal.
- In the continuous-time case, the reduced model is stochastically balanced.
- If G and Gr are the transfer-function matrices of the systems (A,B,C,D) and (Ar,Br,Cr,Dr), respectively, then the relative approximation error satisfies the inequalities

$$HSV(NR+1) \le INFNORM(relerr) \le 2 * \sum_{i=NR+1}^{N} HSV(i)/(1-HSV(i))$$

  where INFNORM(G) is the infinity-norm of G. The relative error relerr is defined by the expression

  Gr = G*(I-relerr)

- The reduced model is computed after the system is reduced to a state coordinate form in which A is in a real Schur form. The matrices of the reduced model are then computed as

  Ar = Ti*A*T, Br = Ti*B, Cr = C*T,

  where T and Ti are N*NR and NR*N projection matrices, respectively. These matrices are available, in column dense forms, in the working array RWORK beginning with elements RWORK(1) and RWORK(N*N+1), respectively.
- The accuracy loss which can be induced in the reduced model by applying these projections or, if T and Ti are invertible, by transforming the given system to the balanced coordinate form, can be estimated by determining the condition numbers of the projection matrices T and Ti.

Copyright:

Life cycle:

1992 JUNE A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
Given the continuous-time system example (A,B,C,D) used by the subroutine RPMRIB (D = 0), a stochastically balanced model for the system (G I) can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .FALSE.
      TOL    = 0.0D0
      FIXORD = 0
      ALPHA  = 1.0D0
      LWORK  = P*(M+P) + N*(10*N+2*M+3*P+8)
      CALL RPMRSB(A, N, B, M, C, P, D, DISCR, ALPHA, TOL, FIXORD,
     *            NR, HSV, RCOND, RWORK, LWORK, IWORK, *1111)
```

The matrices corresponding to the stochastically balanced system (G I) are:

$$
Ar = \begin{pmatrix}
-.1996 & 0 & -2.8560 & 0 & 1.4913 & -.5003 & 0 \\
0 & -.2774 & 0 & -2.3059 & 0 & 0 & -.5931 \\
2.9100 & 0 & -.1252 & 0 & -.0070 & -.3191 & 0 \\
0 & 2.3059 & 0 & -2.0184 & 0 & 0 & -2.3648 \\
-1.5039 & 0 & .6922 & 0 & -1.9158 & 2.4031 & 0 \\
-.5023 & 0 & .4349 & 0 & -2.5019 & -13.6311 & 0 \\
0 & -.5931 & 0 & 2.3648 & 0 & 0 & -13.5759
\end{pmatrix}
$$

$$
Br = \begin{pmatrix}
.4191 & -.4191 \\
-.4857 & -.4857 \\
-.3173 & .3173 \\
.9524 & .9524 \\
.8724 & -.8724 \\
.5402 & -.5402 \\
-.5320 & -.5320
\end{pmatrix}
$$

$$
Cr = \begin{pmatrix}
1.8572 & -1.7570 & .6352 & -1.1934 & -.9958 & .5404 & -.5323 \\
-.1816 & 0 & .8944 & 0 & .4040 & .0137 & 0 \\
-1.8572 & -1.7570 & -.6352 & -1.1934 & .9958 & -.5404 & -.5323
\end{pmatrix}
$$

The computed singular values are:

.8803 .8506 .8038 .4494 .3973 .0214 .0209

The resulted value of RCOND is 0.51984 .

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
A matrix reduction to RSF form failed.
-3-
The system is not stable.
-4-
Computation of singular values failed.
-5-
Selected order larger than the order of minimal realization.
-6-
The matrix D has no maximal row rank.
-7-
Failure of eigenvalues reordering of the Hamiltonian matrix.
-8-
The Hamiltonian matrix is not dichotomic.
-9-
Singular matrix during solution of Riccati equation.
-10-
Not enough working storage. It should be at least //LENG//.
```

SUBROUTINE RPMRST

Square-Root Balancing-Free Stochastic Truncation Method

Procedure purpose:

Given the matrices A, B, C and D of an original stable model,
this subroutine computes the corresponding matrices Ar, Br, Cr and Dr
of a reduced order model by using the square-root balancing-free
version of the balanced stochastic truncation (BST) method of /1/.
For the applicability of the BST method, the given system must have
the number of systems outputs less than or equal to the number of
systems inputs and the transfer-function matrix of the system G,
must have no zeros on the imaginary axis for a continuous-time system
or on the unit circle for a discrete-time system.
In particular, the feedthrough matrix D must have maximal row rank.
If D has maximal column rank, the BST method can be employed
on the dual system with the transfer-function matrix G'.

A parameter ALPHA can be used as a weight between the absolute and
relative errors. For ALPHA <= 0, the BST method is performed on the
original system. If ALPHA > 0, the BST method is performed on a
modified system with the transfer-function matrix (G ALPHA*I).
This is the recommended approach to be used when the conditions on
the number inputs and outputs and/or on the rank of D are not fulfilled
by a given system.

Usage:

```
CALL RPMRST(A, N, B, M, C, P, D, DISCR, ALPHA, TOL, FIXORD,
            NR, HSV, RCOND, RWORK, LWORK, IWORK, *)

A     :  IN, OUT, DOUBLE (N,N)
         On input  : system state matrix A of the original model
                     (column dense)
         On output : the NR*NR matrix Ar of the reduced model
                     (column dense)
N     :  IN, INTEGER
         dimension of state vector
B     :  IN, OUT, DOUBLE (N,M)
         On input  : system input matrix B of the original model
                     (column dense)
         On output : the NR*M matrix Br of the reduced model
                     (column dense)
M     :  IN, INTEGER
         dimension of input vector
C     :  IN, OUT, DOUBLE (P,N)
         On input  : system output matrix C of the original model
                     (column dense)
         On output : the P*NR matrix Cr of the reduced model
                     (column dense)
```

```
P      :  IN, INTEGER
          dimension of output vector
D      :  IN, OUT, DOUBLE (P,M)
          On input  : system feedthrough matrix D of the original model
                      (column dense)
          On output : the P*M matrix Dr of the reduced model
                      (column dense)
DISCR  :  IN, LOGICAL
          specifies the type of the system:
          continuous-time system, if DISCR = .FALSE., or
          discrete-time system, if DISCR = .TRUE.
ALPHA  :  IN, DOUBLE, SELECTION PARAMETER
          ALPHA > 0 specifies the absolute/relative error weighting
          parameter. A large positive value of ALPHA favours the
          minimization of the absolute approximation error, while a
          small value of ALPHA is appropriate for the minimization
          of the relative error.
          ALPHA <= 0 means pure relative error method and can be
          used only if rank(D) = P and P <= M.
TOL    :  IN, DOUBLE
          absolute tolerance used for determining the order of the
          reduced model (if FIXORD = 0). When TOL .LE. 0, an internally
          computed default value TOL = N*EPS is used, where EPS is the
          machine precision (see the LAPACK Library Routine DLAMCH).
FIXORD:   IN, INTEGER
          desired order of the reduced model. If FIXORD > 0, the order NR
          is set to FIXORD. If FIXORD = 0, the order NR is chosen in
          accordance with the specified tolerance TOL.
NR     :  OUT, INTEGER
          resulted order of the reduced model. If FIXORD > 0, NR is set
          equal to FIXORD. If FIXORD = 0, NR is set equal to the number
          of singular values (HSV(i), i = 1, N) greater than TOL.
HSV    :  OUT, DOUBLE (N)
          The Hankel singular values of the systems phase matrix,
          ordered decreasingly. The singular values are less than or
          equal to 1.
RCOND  :  OUT, DOUBLE
          indicator of conditioning of the Riccati equation satisfied
          by the observability Gramian. A small value of RCOND
          (RCOND <= 1.E-7 ) means possible accuracy loss due to an
          ill-conditioned Riccati equation. Usually, the conditioning
          can be improved by choosing a larger value for ALPHA.
RWORK  :  OUT, DOUBLE (LWORK)
          working array.
LWORK  :  IN, INTEGER
          dimension of working array RWORK.
          The value of LWORK must be at least
          P*MB + max(MB*P,8*N) + N*(2*MB+P+N+max(N,M,P)+4*max(P,2*N),
          where MB = M+P if ALPHA > 0 and MB = M otherwise.
```

```
IWORK :  OUT, INTEGER (2*N)
         working array.
*     :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

Literature

/1/ Varga A. and Fasol K.H.
A New Square-Root Balancing-Free Stochastic Truncation Model Reduction Algorithm.
Prepr. 12th IFAC Congress, Sydney, vol. 7, pp. 153-156, 1993.

Remarks:

- If HSV(NR) > HSV(NR+1), the resulting reduced model (Ar,Br,Cr,Dr) is stable and minimal, but not stochastically balanced.
- If G and Gr are the transfer-function matrices of the systems (A,B,C,D) and (Ar,Br,Cr,Dr), respectively, then the relative approximation error satisfies the inequalities

$$HSV(NR+1) \leq INFNORM(relerr) \leq 2 * \sum_{i=NR+1}^{N} HSV(i)/(1-HSV(i))$$

  where INFNORM(G) is the infinity-norm of G. The relative error relerr is defined by the expression

```
Gr = G*(I-relerr)
```

- The reduced model is computed after the system is reduced to a state coordinate form in which A is in a real Schur form. The matrices of the reduced model are then computed as

```
Ar = Ti*A*T, Br = Ti*B, Cr = C*T,
```

  where T and Ti are N*NR and NR*N projection matrices, respectively. These matrices are available, in column dense forms, in the working array RWORK beginning with elements RWORK(1) and RWORK(N*N+1), respectively. In the implemented version, T always results with orthonormal columns.
- The accuracy loss which can be induced in the reduced model by applying these projections can be estimated by determining the condition number of the projection matrix Ti.

Copyright:

Life cycle:

1992 JUNE A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the continuous-time system example (A,B,C,D) used by the subroutine RPMRIB (D = 0), a fifth order BST approximation for the system (G I) can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .FALSE.
      TOL    = 0.1D0
      FIXORD = 0
      ALPHA  = 1.0D0
      LWORK  = P*(M+P) + N*(10*N+2*M+3*P+8)
      CALL RPMRST(A, N, B, M, C, P, D, DISCR, ALPHA, TOL, FIXORD,
     *            NR, HSV, RCOND, RWORK, LWORK, IWORK, *1111)
```

The matrices of reduced system are:

$$Ar = \begin{pmatrix} 1.2729 & 0 & -6.5947 & 0 & -3.4229 \\ 0 & .8169 & 0 & 2.4821 & 0 \\ 2.9889 & 0 & -2.9028 & 0 & .3692 \\ 0 & -3.3921 & 0 & -3.1126 & 0 \\ -1.4767 & 0 & 2.0339 & 0 & -.6107 \end{pmatrix}$$

$$Br = \begin{pmatrix} .1331 & -.1331 \\ -.0862 & -.0862 \\ 2.6777 & -2.6777 \\ -3.5767 & -3.5767 \\ -2.3033 & 2.3033 \end{pmatrix}$$

$$Cr = \begin{pmatrix} -.6907 & -.6882 & -.0779 & .0958 & -.0038 \\ .0676 & 0 & -.6532 & 0 & -.7522 \\ .6907 & -.6882 & .0779 & .0958 & .0038 \end{pmatrix}$$

The computed singular values are:

.8803 .8506 .8038 .4494 .3973 .0214 .0209

The resulted value of RCOND is 0.51984 .

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
A matrix reduction to RSF form failed.
-3-
The system is not stable.
-4-
Computation of singular values failed.
-5-
Selected order larger than the order of minimal realization.
-6-
The matrix D has no maximal row rank.
-7-
Failure of eigenvalues reordering of the Hamiltonian matrix.
-8-
The Hamiltonian matrix is not dichotomic.
-9-
Singular matrix during solution of Riccati equation.
-10-
Not enough working storage. It should be at least //LENG//.
```

SUBROUTINE RPMRSR

Singular Perturbation Reduction Formulas

Procedure purpose:

For the stable system (A,B,C,D) with matrices A, B and C partitioned as

$$A = \begin{pmatrix} A11 & A12 \\ A21 & A22 \end{pmatrix}, \quad B = \begin{pmatrix} B1 \\ B2 \end{pmatrix}, \quad C = \begin{pmatrix} C1 & C2 \end{pmatrix},$$

this subroutine computes the matrices of the reduced order system (Ar,Br,Cr,Dr) by using the singular perturbation approximation formulas:

$$Ar = A11 + A12*(g*I-A22)^{-1}*A21, \quad Br = B1 + A12*(g*I-A22)^{-1}*B2$$

$$Cr = C1 + C2*(g*I-A22)^{-1}*A21, \quad Dr = D + C2*(g*I-A22)^{-1}*B2$$

where g = 0 for a continuous-time system and g = 1 for a discrete-time system.

Usage:

```
CALL RPMRSR(A, N, B, M, C, P, D, DISCR, NR, IWORK, *)
```

```
A     :  IN, OUT, DOUBLE (N,N)
         On input  : system state matrix A of the original model
                     (column dense)
         On output : the NR*NR matrix Ar of the reduced model
                     (column dense)
N     :  IN, INTEGER
         dimension of state vector
B     :  IN, OUT, DOUBLE (N,M)
         On input  : system input matrix B of the original model
                     (column dense)
         On output : the NR*M matrix Br of the reduced model
                     (column dense)
M     :  IN, INTEGER
         dimension of input vector
C     :  IN, OUT, DOUBLE (P,N)
         On input  : system output matrix C of the original model
                     (column dense)
         On output : the P*NR matrix Cr of the reduced model
                     (column dense)
P     :  IN, INTEGER
         dimension of output vector

D     :  IN, OUT, DOUBLE (P,M)
```

```
          On input  : system feedthrough matrix D of the original model
                      (column dense)
          On output : the P*M matrix Dr of the reduced model
                      (column dense)
 DISCR :  IN, LOGICAL
          specifies the type of the system:
          continuous-time system, if DISCR = .FALSE., or
          discrete-time system, if DISCR = .TRUE.
 NR    :  IN, INTEGER
          desired order of the reduced model.
 IWORK :  OUT, INTEGER (N-NR)
          working array.
 *     :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

Literature

/1/ Liu Y. and Anderson B.D.O.
Singular perturbation approximation of balanced systems,
Int. J. Control, Vol. 50, pp. 1379-1405, 1989.

Remarks:

- The resulting system (Ar,Br,Cr,Dr) has the same steady-state gain as the given full order system.
- This routine usually follows RPMRIB for computing balanced singular perturbation approximations /1/.
- Used with NR = 0, this routine evaluates the steady-state gain of a stable system.

Life cycle:

1992 MAY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the continuous-time system (A,B,C,D) with the matrices A, B and C as in example for the subroutine RPMRIB and D set to a null matrix, a balanced singular perturbation approximation of order 5 can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .FALSE.
      TOL    = 0.0D0
```

```
      FIXORD = 0
      LWORK  = 4*(N+1)*N
      NSPA   = 5
C     Compute a balanced minimal realization.
      CALL RPMRIB(A, N, B, M, C, P, DISCR, TOL, FIXORD, NR, HSV,
     *            RWORK, LWORK, *1111)
C     Perform a balanced singular perturbation model reduction.
      CALL RPMRSR(A, NR, B, M, C, P, D, DISCR, NSPA, IWORK, *1111)
```

The matrices of balanced singular perturbation approximation are:

$$Ar = \begin{pmatrix} -.3813 & 3.0418 & 0 & 0 & 1.0286 \\ -3.0996 & -.2635 & 0 & 0 & -.3187 \\ 0 & 0 & -.5557 & -2.3036 & 0 \\ 0 & 0 & 2.3036 & -2.1518 & 0 \\ -1.0479 & -.6685 & 0 & 0 & -2.0447 \end{pmatrix}$$

$$Br = \begin{pmatrix} -.9790 & .9790 \\ -.7412 & .7412 \\ -1.0323 & -1.0323 \\ 1.2844 & 1.2844 \\ -1.0578 & 1.0578 \end{pmatrix}$$

$$Cr = \begin{pmatrix} -.9723 & .6513 & -1.0323 & -1.2844 & .9914 \\ .1620 & .5004 & 0 & 0 & -.5217 \\ .9723 & -.6513 & -1.0323 & -1.2844 & -.9914 \end{pmatrix}$$

$$Dr = \begin{pmatrix} .0498 & -.0007 \\ .0010 & -.0010 \\ -.0007 & .0498 \end{pmatrix}$$

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
Matrix A22 is singular.
```

SUBROUTINE RPMRDC

Bilinear Transformation of Transfer-Function Matrices

Procedure purpose:

This subroutine performs a transformation on the parameters (A,B,C,D) of a system, which is equivalent to a bilinear transformation of the corresponding transfer function matrix.

For a continuous-time system, the resulting matrices $(\bar{A},\bar{B},\bar{C},\bar{D})$ correspond to the continuous-to-discrete bilinear transformation

$$s \rightarrow z = \text{alpha}\,\frac{\text{beta} + s}{\text{beta} - s}\,.$$

For a discrete-time system, the resulting matrices $(\bar{A},\bar{B},\bar{C},\bar{D})$ correspond to the discrete-to-continuous bilinear transformation

$$z \rightarrow s = \text{beta}\,\frac{z - \text{alpha}}{z + \text{alpha}}\,.$$

Usage:

```
CALL RPMRDC(A, N, B, M, C, P, D, DISCR, ALPHA, BETA,
            RWORK, LWORK, IWORK, *)
```

A : IN, OUT, DOUBLE (N,N)
On input : system state matrix A of the original system (column dense)

On output : the N*N matrix $\bar{A}$ of the transformed system (column dense)

N : IN, INTEGER
dimension of state vector

B : IN, OUT, DOUBLE (N,M)
On input : system input matrix B of the original system (column dense)

On output : the N*M matrix $\bar{B}$ of the transformed system (column dense)
If M = 0, this matrix is not referenced.

M : IN, INTEGER
dimension of input vector

C : IN, OUT, DOUBLE (P,N)
On input : system output matrix C of the original system (column dense)

On output : the P*N matrix $\bar{C}$ of the transformed system

```
                    (column dense)
          If P = 0, this matrix is not referenced.
P      :  IN, INTEGER
          dimension of output vector
D      :  IN, OUT, DOUBLE (P,M)
          On input  : system feedthrough matrix D of the original system
                      (column dense)

                                  -
          On output : the P*M matrix D of the transformed system
                      (column dense)
          If M = 0 or P = 0, this matrix is not referenced.
DISCR  :  IN, LOGICAL
          specifies the type of the original system:
          continuous-time system, if DISCR = .FALSE., or
          discrete-time system, if DISCR = .TRUE.
ALPHA  :  IN, DOUBLE
          parameter alpha of the bilinear transformation.
          It should be non-zero.
BETA   :  IN, DOUBLE
          parameter beta of the bilinear transformation.
          It should be non-zero.
RWORK  :  OUT, DOUBLE (LWORK)
          working array.
LWORK  :  IN, INTEGER
          dimension of working array RWORK.
          The value of LWORK must be at least  N.
IWORK  :  OUT, INTEGER (N)
          working array.
*      :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

Literature

/1/ Al-Saggaf U.M. and Franklin G.F.
Model reduction via balanced realizations: a extension and frequency weighting techniques, IEEE Trans. Autom. Contr., Vol.33, pp. 687-692, 1988.

Remarks:

- For stable systems, the recommended values of the parameters alpha and beta are: alpha = 1 and beta = 1. By using these values, to left half-plane poles of continuous-time systems correspond poles of the resulting discrete-time systems inside the unit circle and vice versa.
- For an unstable continuous-time system, beta should be not a pole of the original system.
- For an unstable discrete-time system, -alpha should be not a pole of the original system.

Copyright:
1992 - DLR Institut fuer Dynamik der Flugsysteme

Life cycle:
1992 JUNE A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
Given the discrete-time system (A,B,C,D) with the following matrices:

```
    (  .972774  .049696  .414321 -.485309   .013280  .000135  .002015 )
    ( -.293823  .279304 -.077753  .087312  -.001739 -.000012 -.000211 )
    ( -.052626  .155605  .707804  .009770  -.000143        0 -.000013 )
A = (  .053758  .001022  .011947  .973372  -.053758 -.001022 -.011947 )
    (  .013280  .000135  .002015  .485309   .972774  .049696  .414321 )
    ( -.001739 -.000012 -.000211 -.087312  -.293823  .279304 -.077753 )
    ( -.000143        0 -.000013 -.009770  -.052626  .155605  .707804 )

    (  .023476  .000035 )
    (  .710910 -.000002 )
    (  .126805        0 )        ( 1  0  0  0  0  0  0 )       ( 0  0 )
B = (  .000344 -.000344 )    C = ( 0  0  0  1  0  0  0 )   D = ( 0  0 )
    (  .000035  .023476 )        ( 0  0  0  0  1  0  0 )       ( 0  0 )
    ( -.000002  .710910 )
    (        0  .126805 )
```

a balanced truncated approximation of order 5 can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .TRUE.
      ALPHA  = 1.0D0
      BETA   = 1.0D0
      TOL    = 0.0D0
      FIXORD = 5
      LWORK  = 4*(N+1)*N
      NSPA   = 5
C     Perform the bilinear transformation.
      CALL RPMRDC(A, N, B, M, C, P, D, DISCR, ALPHA, BETA,
     *            RWORK, LWORK, IWORK, *1111)
C     Perform model reduction.
      CALL RPMRIB(A, N, B, M, C, P, .NOT.DISCR, TOL, FIXORD,
     *            NR, HSV, RWORK, LWORK, *1111)
C     Perform the inverse bilinear transformation.
      CALL RPMRDC(A, NR, B, M, C, P, D, .NOT.DISCR, ALPHA,
     *            BETA, RWORK, LWORK, IWORK, *1111)
```

The matrices of balanced singular perturbation approximation are:

$$Ar = \begin{pmatrix} .9187 & -.2877 & 0 & 0 & .0762 \\ .2939 & .9260 & 0 & 0 & .0408 \\ 0 & 0 & .9296 & -.1951 & 0 \\ 0 & 0 & .1951 & .8060 & 0 \\ -.0732 & .0718 & 0 & 0 & .8478 \end{pmatrix}$$

$$Br = \begin{pmatrix} -.3325 & .3325 \\ .1977 & -.1977 \\ .3436 & .3436 \\ -.3293 & -.3293 \\ -.2662 & .2662 \end{pmatrix}$$

$$Cr = \begin{pmatrix} -.3256 & -.1731 & .3436 & .3293 & .2465 \\ .0414 & -.1593 & 0 & 0 & -.1626 \\ .3256 & .1731 & .3436 & .3293 & -.2465 \end{pmatrix}$$

$$Dr = \begin{pmatrix} -.0166 & .0008 \\ -.0013 & .0013 \\ .0008 & -.0166 \end{pmatrix}$$

Note that the resulting model is the same as that which would be obtained by using the balanced singular perturbation approximation approach on the original discrete system.

Error Messages:

-1-
Invalid parameter value on entry.
-2-
The matrix (ALPHA*I + A) is exactly singular.
-3-
The matrix (BETA*I - A) is exactly singular.

2. Order Reduction of Unstable State-Space Systems

The subroutines of this chapter can be used to compute reduced order approximations of unstable state-space systems. The main intended application of these subroutines is for controller order reduction. The provided subroutines represent useful tools by which the powerful model reduction methods for stable systems can be efficiently employed to compute reduced order approximations of unstable systems. All routines can be used for computing reduced models of both continuous-time and discrete-time systems. Moreover, all routines can be used to reliably compute minimal state-space realizations from given non-minimal unstable models.

Two alternative basic approaches can be used for reducing unstable models:

1. Reduction of stable projections

If G is the transfer-function matrix of a n-th order (not necessarily stable) system, then the following procedure can be used to reduce the order of G:

1) Decompose additively G as

$$G = G_1 + G_2$$

such that G_1 has only stable poles and G_2 has only unstable poles.

2) Determine G_{1r}, a reduced order approximation of the stable part G_1.

3) Assemble the reduced model G_r as

$$G_r = G_{1r} + G_2$$

For the model reduction at step 2 any of methods available for stable systems can be used. The following routines are provided to perform the computations at steps 1 and 3.

RPMRSD computes the terms G_1 and G_2 of an additive spectral decomposition of a transfer-function matrix G with respect to a specified region of the complex plane. G_1 and G_2 are determined such that G_1 has only poles in that region and G_2 has exclusively poles outside that region. This subroutine can be used to compute stable and anti-stable projections or fast and slow modes decompositions of the given transfer-function matrix. The computation of additive decompositions is based on an algorithm explained in /1/.

RPMRAS adds or subtracts two systems in state-space form.

2. Reduction of stable coprime factors

The following procedure can be used to compute an r-th order approximation G_r of an n-th order (not necessarily stable) system G:

1) Compute a left coprime factorization of the transfer-function matrix G in the form

$$G = R^{-1}Q ,$$

where R and Q are stable transfer-function matrices of degree n.

2) Approximate the transfer-function matrix $[Q\ R]$ of degree n with $[Q_r\ R_r]$ of degree r by using a model reduction method for stable systems.

3) Form the r-th order approximation of G as

$$G_r = R_r^{-1}Q_r.$$

A similar procedure can be given for a right coprime factorization of G in the form

$$G = QR^{-1}.$$

For the model reduction at step 2 any of methods available for stable systems can be used. The following routines are provided to perform the computations at steps 1 and 3:

RPMRLF computes the state-space representations for the factors of a left coprime factorization of a transfer-function matrix with prescribed stability degree /2/.

RPMRRF computes the state-space representations for the factors of a right coprime factorization of a transfer-function matrix with prescribed stability degree /2/.

RPMRLI computes the state-space representations for the factors of a left coprime factorization with (co)inner denominator of a transfer-function matrix /3/.

RPMRRI computes the state-space representations for the factors of a right coprime factorization with inner denominator of a transfer-function matrix /3/.

RPMRLB computes the state-space representation corresponding to a left coprime factorization of a transfer-function matrix.

RPMRRB computes the state-space representation corresponding to a right coprime factorization of a transfer-function matrix.

Literature:

/1/ Safonov, M.G., Jonckheere, E.A., Verma, M. and Limebeer, D.J.
Synthesis of Positive Real Multivariable Feedback Systems,
Int. J. Control, Vol. 45, pp. 817-842, 1987.

/2/ Varga A.
Coprime Factors Model Reduction Based on Accuracy
Enhancing Techniques,
Systems Analysis, Modelling and Simulation, vol. 11,
pp. 303-311, 1993.

/3/ Varga A.
A Schur Method for Computing Coprime Factorizations
with Inner Denominators and Applications in Model Reduction,
Proc. 1993 ACC, San Francisco, CA, pp. 2130-2131, 1993.

SUBROUTINE RPMRSD

Additive Spectral Decomposition of Linear Systems

Procedure purpose:

For the system G = (A,B,C,D) with the transfer-function matrix G, a similarity transformation matrix T is determined such that the transformed system

$$(\bar{A},\bar{B},\bar{C},D) = (T^{-1}AT,\ T^{-1}B,\ CT,\ D) \qquad (1)$$

has the state-matrix $\bar{A}$ in a block diagonal form. If we partition the transformed systems matrices conformally with the structure of the state-matrix

$$\bar{A} = \begin{pmatrix} A1 & 0 \\ 0 & A2 \end{pmatrix},\quad \bar{B} = \begin{pmatrix} B1 \\ B2 \end{pmatrix},\quad \bar{C} = \begin{pmatrix} C1 & C2 \end{pmatrix}, \qquad (2)$$

then A1 has eigenvalues in a region of interest and A2 outside of that region. The decomposition (2) corresponds to an additive decomposition of the transfer-function matrix G as

$$G = G1 + G2$$

where G1 = (A1,B1,C1,D1) has order N1, G2 = (A2,B2,C2,D2) has order N2, and D = D1+D2.

This subroutine can be used to compute stable and antistable projections in order to perform further order reduction of the stable part. It can be also used for determining fast and slow modes decompositions in order to remove the fast modes of a given system.

Usage:

```
CALL RPMRSD(A, N, B, M, C, P, D, WITHD, DISCR, ALPHA, STDOM,
            BDIAG, BETA, N1, N2, A2, B2, C2, D2, RWORK, LWORK, *)
```

```
A     :  IN, OUT, DOUBLE (N,N)
         On input  : system state-matrix A of the original model
                     (column dense)
         On output : If BDIAG = .TRUE., the N x N state matrix Ā
                     in the block diagonal form (2).
                     (column dense)
                     The leading N1 x N1 submatrix A1 of A has its
                     eigenvalues in the domain of interest defined
                     by ALPHA and STDOM, while the trailing N2 x N2
                     submatrix A2 has its eigenvalues outside the
                     domain of interest.
                     If BDIAG = .FALSE., the N1 x N1 submatrix A1
```

```
                    from (2)..
                    (column dense)
                    (see the description of parameter STDOM).
N      :  IN, INTEGER
          dimension of state vector
B      :  IN, OUT, DOUBLE (N,M)
          On input  : system input matrix B of the original model
                      (column dense)
          On output : If BDIAG = .TRUE., the N x M input matrix B
                      of the transformed system (2).
                      (column dense)
                      If BDIAG = .FALSE., the N1 x M submatrix B1
                      from (2).
                      (column dense)
          If M = 0, this matrix is not referenced.
M      :  IN, INTEGER
          dimension of input vector
C      :  IN, OUT, DOUBLE (P,N)
          On input  : system output matrix C of the original model
                      (column dense)
          On output : If BDIAG = .TRUE., the P x N output matrix C
                      of the transformed system (2).
                      (column dense)
                      If BDIAG = .FALSE., the P x N1 submatrix C1
                      from (2).
          On output : the P x N1 submatrix C1 from (2)
                      (column dense)
          If P = 0, this matrix is not referenced.
P      :  IN, INTEGER
          dimension of output vector
D      :  IN, OUT, DOUBLE (P,M)
          On input  : If WITHD = .TRUE., the P*M system feedthrough
                      matrix D of the original system.
                      (column dense)
          On output : If WITHD = .TRUE. and BDIAG = .FALSE.,
                      the matrix BETA*D.
                      (column dense)
          If WITHD = .FALSE, or BDIAG = .TRUE., or M = 0, or P = 0,
          D is not referenced.
WITHD  :  IN, LOGICAL
          specifies whether or not a given feedthrough matrix D is used
          in computing the additive decomposition:
          WITHD = .TRUE.  means a given feedthrough matrix D is used in
                          determining G1 and G2.
          WITHD = .FALSE. means D is assumed a null matrix.
DISCR  :  IN, LOGICAL
          specifies the type of the system:
          continuous-time system, if DISCR = .FALSE., or
          discrete-time system, if DISCR = .TRUE.
ALPHA  :  IN, DOUBLE
          specifies the boundary parameter for the domain of interest
```

```
          for eigenvalues (see the description of parameter STDOM).
STDOM :   IN, LOGICAL
          specifies whether the domain of interest is of stability
          type (left half plane or inside of a circle) or of
          instability type (right half plane or outside of a circle).
          STDOM and ALPHA define the domain of interest for
          lambda(A), the eigenvalues of A, as follows:
          For a continuous-time system (DISCR =.FALSE.)
              Real(lambda(A)) <  ALPHA if STDOM =.TRUE. ;
              Real(lambda(A)) >= ALPHA if STDOM =.FALSE.;
          For a discrete-time system (DISCR =.TRUE.)
              Abs(lambda(A)) <  ALPHA if STDOM =.TRUE. ;
              Abs(lambda(A)) >= ALPHA if STDOM =.FALSE..
BDIAG :   IN, LOGICAL
          specifies whether the block diagonal decomposition (2) or
          the additive spectral decomposition is to be computed:
          BDIAG = .TRUE.,  the block diagonal decomposition (2) is
                           to be computed
          BDIAG = .FALSE., the matrices of the additive spectral
                           decomposition G = G1 + G2 resulting from
                           (2) are to be computed.
BETA  :   IN, DOUBLE
          The scaling factor used to compute D1 and D2.
N1    :   OUT, INTEGER.
          number of eigenvalues of the matrix A lying inside of
          the domain of interest. N1 is also the dimension of
          the invariant subspace of A corresponding to the eigenvalues
          of A1.
N2    :   OUT, INTEGER.
          number of eigenvalues of the matrix A lying outside of
          the domain of interest. N2 is also the dimension of the
          reducing subspace of A corresponding to the eigenvalues
          of A2.  Notice that N = N1 + N2.
A2    :   OUT, DOUBLE (N2,N2)
          If BDIAG = .FALSE., the N2 x N2 matrix A2 from (2).
          (column dense)
          This matrix has its eigenvalues outside of the domain of
          interest defined by ALPHA and STDOM.
          (see the description of parameter STDOM).
          If BDIAG = .TRUE., this matrix is not referenced.
B2    :   OUT, DOUBLE (N2,M)
          the N2 x M matrix B2 from (2).
          (column dense)
          If BDIAG = .TRUE. or M = 0, this matrix is not referenced.
C2    :   OUT, DOUBLE (P,N2)
          the P x N2 matrix C2 from (2)
          (column dense)
          If BDIAG = .TRUE. or P = 0, this matrix is not referenced.
          Note: If A2, B2 and C2 are not of interest, then these
                arrays can overlap the working array RWORK.
D2    :   OUT, DOUBLE
```

```
          the P x M matrix D2 = (1-BETA)*D.
          (column dense)
          This matrix is not referenced if D is not referenced.
 RWORK :  OUT, DOUBLE (LWORK).
          working array. On normal return, the leading N x N part of
          RWORK, contains the transformation matrix T used in (1)
          to reduce the state-matrix A to the block-diagonal form (2).
 LWORK :  IN, INTEGER.
          dimension of working array RWORK.
          The value of LWORK must be at least  N*(N+5).
 *      :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

Literature

/1/ Safonov, M.G., Jonckheere, E.A., Verma, M. and Limebeer, D.J.
Synthesis of positive real multivariable feedback systems,
Int. J. Control, Vol. 45, pp. 817-842, 1987.

Remarks:

- The accuracy loss which can be induced in the decomposed model by applying the transformation (1) can be estimated by determining the condition number of the transformation matrix T, returned in RWORK.

Copyright:

Life cycle:

1992 JUNE A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the unstable continuous-time system (A,B,C) with the following matrices:

$$A = \begin{pmatrix} -.04165 & 0 & 4.92 & .492 & 0 & 0 & 0 \\ -5.21 & -12.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.33 & -3.33 & 0 & 0 & 0 & 0 \\ .545 & 0 & 0 & 0 & .0545 & 0 & 0 \\ 0 & 0 & 0 & -.492 & .004165 & 0 & 4.92 \\ 0 & 0 & 0 & 0 & .521 & -12.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.33 & -3.33 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 \\ 12.5 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 12.5 \\ 0 & 0 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

a slow/fast decomposition can be computed in order to directly remove the fast modes from the system. The following sequence of statements can be used to separate the modes with respect to the boundary value -4 for the real parts of eigenvalues:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .FALSE.
      ALPHA  = -4.0D0
      STDOM  = .FALSE.
      BDIAG  = .FALSE.
      WITHD  = .FALSE.
      BETA   = 1.0D0
      LWORK  =  N*(N+5)
      CALL RPMRSD(A, N, B, M, C, P, D, WITHD, DISCR, ALPHA, STDOM,
     *            BDIAG, BETA, N1, N2, A2, B2, C2, D, RWORK, LWORK,
     *            *1111)
```

The matrices of the slow subsystem are:

$$A1 = \begin{pmatrix} -1.4178 & -5.1682 & .5481 & 2.5944 & .0695 \\ .9109 & -1.4178 & .1460 & .4718 & .3044 \\ 0 & 0 & .1605 & -.4948 & 4.7654 \\ 0 & 0 & .0474 & .1605 & -.3769 \\ 0 & 0 & 0 & 0 & -3.5957 \end{pmatrix}$$

$$B1 = \begin{pmatrix} 2.8493 & .0351 \\ 2.9533 & .0993 \\ -.3203 & 1.6526 \\ -1.0775 & -.1609 \\ .0089 & -4.7125 \end{pmatrix}$$

$$C1 = \begin{pmatrix} -.8659 & .2787 & -.0185 & -.2005 & -.0002 \\ .0797 & -.3951 & -.0427 & -.9141 & .0068 \\ -.0165 & -.0645 & -.9935 & .0732 & .0376 \end{pmatrix}$$

```
 The matrices of the fast subsystem are:

 A2 = ( -13.1627         0 )        B2 = ( -11.4205   -.0015 )
      (        0 -12.4245 )             (   -.0020  12.4858 )

      ( -.1245         0 )
 C2 = (  .0052   -.0006 )
      (  .0002    .1472 )

 The slow subsystem of order N1 = 5 can be used as a dominant modes
 approximation of the original system.
```

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
The reduction of A to the real Schur form failed.
-3-
The reordering of eigenvalues failed.
-4-
Separation failure due to very close eigenvalues.
-5-
Not enough working storage. It should be at least //LENG//.
```

SUBROUTINE RPMRAS

Sum or Difference of Two Systems in State-Space Form

Procedure purpose:

If (A1,B1,C1,D1) and (A2,B2,C2,D2) are the state-space models of the given two systems having the transfer-function matrices G1 and G2, respectively, this subroutine constructs the state-space model (A,B,C,D) which corresponds to the transfer-function matrix G = G1 + G2 or G = G1 - G2.

Usage:

```
CALL RPMRAS(A1, N1, B1, M, C1, P, D1, OP, A2, N2, B2, C2, D2,
            A, N, B, C, D, *)
```

```
A1    : IN, DOUBLE (N1,N1)
        system state-matrix A1 of the first system
        (column dense)
        If N1 = 0, this matrix is not referenced.
N1    : IN, INTEGER
        dimension of state vector of first system
B1    : IN, DOUBLE (N1,M)
        system input matrix B1 of the first system
        (column dense)
        If N1 = 0, this matrix is not referenced.
M     : IN, INTEGER
        dimension of the common input vector of both systems
C1    : IN, DOUBLE (P,N1)
        system output matrix C1 of the first system
        (column dense)
        If N1 = 0, this matrix is not referenced.
P     : IN, INTEGER
        dimension of output vector
D1    : IN, DOUBLE (P,M)
        system feedthrough matrix D1 of the first system
        (column dense)
OP    : IN, INTEGER, SELECTION PARAMETER
        specifies the operation to be performed.
        If OP > 0, the sum of systems is computed.
        IF OP < 0, the difference of systems is computed.
A2    : IN, DOUBLE (N2,N2)
        system state-matrix A2 of the second system
        (column dense)
        If N2 = 0, this matrix is not referenced.
N2    : IN, INTEGER
        dimension of state vector of second system
B2    : IN, DOUBLE (N2,M)
        system input matrix B2 of the second system
        (column dense)
```

```
          If N2 = 0, this matrix is not referenced.
C2    :   IN, DOUBLE (P,N2)
          system output matrix C2 of the second system
          (column dense)
          If N2 = 0, this matrix is not referenced.
D2    :   IN, DOUBLE (P,M)
          system feedthrough matrix D2 of the second system
          (column dense)
A     :   OUT, DOUBLE (N,N)
          the N x N state matrix A of the resulting system.
          (column dense)
          This matrix matrix can be overwritten on A1, if A1 provides
          sufficient storage space.
          If N = 0, this matrix is not referenced.
N     :   OUT, INTEGER
          set to N1 + N2, the dimension of state vector of the
          resulting system
B     :   OUT, DOUBLE (N,M)
          the N x M input matrix B of the resulting system.
          (column dense)
          This matrix matrix can be overwritten on B1, if B1 provides
          sufficient storage space.
          If N = 0, this matrix is not referenced.
C     :   OUT, DOUBLE (P,N)
          the P x N output matrix C of the resulting system
          (column dense)
          This matrix matrix can be overwritten on C1, if C1 provides
          sufficient storage space.
          If N = 0, this matrix is not referenced.
D     :   OUT, DOUBLE (P,M).
          the P x M feedthrough matrix D of the resulting system
          (column dense)
          This matrix matrix can be overwritten on D1.
*     :   RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The matrices of the resulting systems are determined as:

$$A = \begin{pmatrix} A1 & 0 \\ 0 & A2 \end{pmatrix}, \quad B = \begin{pmatrix} B1 \\ B2 \end{pmatrix}$$

$$C = \begin{pmatrix} C1 & S*C2 \end{pmatrix}, \quad D = D1 + S*D2$$

where S = 1 if OP > 0 and S = -1 if OP < 0.

Remarks:
none

Copyright:
1992 - DLR Institut fuer Dynamik der Flugsysteme

Life cycle:
1992 JUNE A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
The difference of two systems (A1,B1,C1,D1) and (A2,B2,C2,D2) can be computed in the place of the first system, with the following sequence of statements:

```
      OP   = -1
      CALL RPMRAS(A1, N1, B1, M, C1, P, D1, OP, A2, N2, B2, C2, D2,
     *            A1, N, B1, C1, D1, *1111)
```

Error Messages:

-1-
Invalid parameter value on entry.

SUBROUTINE RPMRLF

Left Coprime Factorization with Prescribed Stability Degree

Procedure purpose:

Given the matrices A, B, C and D of a state-space representation of a transfer-function matrix G, this subroutine computes the matrices AQR, BQR, CQR, DQR of a state-space representation of the transfer-function matrix $\begin{pmatrix} Q \\ R \end{pmatrix}$ such that $G = R^{-1} * Q$ is a stable rational left coprime factorization of G. The state matrix AQR can be determined such that its eigenvalues are inside a specified stability region of the complex plane. The computation of the factorization is based on a pole assignment method described in /1/.

Usage:

```
CALL RPMRLF(A, N, B, M, C, P, D, WITHD, DISCR, ALPHA, BETA, TOL,
            NQ, NR, RWORK, LWORK, *)
```

A : IN, OUT, DOUBLE (N,N)
  On input : system state matrix A of the given state-space representation of G (column dense)
  On output : the NQ*NQ state matrix AQR in real Schur form of the resulting state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

N : IN, INTEGER
  dimension of state vector

B : IN, OUT, DOUBLE (N,M+P)
  On input : the N*M system input matrix B of the given state-space representation of G (column dense)
  On output : the NQ*(M+P) input matrix BQR of the resulting state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

M : IN, INTEGER
  dimension of input vector

C : IN, OUT, DOUBLE (P,N)
  On input : the P*N system output matrix C of the given state-space representation of G (column dense)
  On output : the P*NQ output matrix CQR of the resulting state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

P : IN, INTEGER
  dimension of output vector

D : IN, OUT, DOUBLE (P,M+P)

On input : If WITHD = .TRUE., the P*M system feedthrough matrix D of the given state-space representation of G. If WITHD = .FALSE, D is implicitly assumed a P x M null matrix in computing the matrices of the left coprime factorization. (column dense)

On output : the P*(M+P) feedthrough matrix DQR of the resulting state-space representation of (Q R). (column dense)

WITHD : IN, LOGICAL
specifies whether or not a given feedthrough matrix D is used in computing the factor Q of the coprime factorization:
WITHD = .TRUE. means a given feedthrough matrix D is used in determining Q.
WITHD = .FALSE. means D is assumed a null matrix in determining Q.

DISCR : IN, LOGICAL
specifies the type of the system:
continuous-time system, if DISCR = .FALSE., or
discrete-time system, if DISCR = .TRUE.

ALPHA : IN, DOUBLE
specifies the boundary of the stability region for the eigenvalues of A. For a continuous-time system (DISCR = .FALSE.), ALPHA is the maximum admissible value for the real parts of eigenvalues, while for a discrete-time system (DISCR = .TRUE.), ALPHA represents the maximum admissible value for the moduli of eigenvalues. The eigenvalues inside the stability region will not be modified by the pole assignment algorithm.
ALPHA >= 0 if DISCR = .TRUE. .

BETA : IN, DOUBLE
specifies the desired stability degree to be assigned for the eigenvalues of A outside the stability region defined by ALPHA. The eigenvalues outside the stability region will be assigned to have the real parts equal to BETA and unmodified imaginary parts for a continuous-time system (DISCR = .FALSE.) or moduli equal to BETA for a discrete-time system (DISCR = .TRUE.).

TOL : IN, DOUBLE
absolute tolerance level below which the elements of C are considered zero (used for observability tests). Recommended value is TOL = N * EPS * NORM(C), where NORM(C) denotes the infinity-norm of C and EPS is the machine precision (see LAPACK Library Routine DLAMCH).

NQ : OUT, INTEGER
the order of the resulting factors Q and R.
Generally NQ = N - NS, where NS is the number of undetectable eigenvalues outside the stability region.

```
NR    :  OUT, INTEGER
         the order of the minimal realization of the factor R.
         Generally NR is the number of observable eigenvalues
         of A outside the stability region (the number of modified
         eigenvalues).
RWORK :  OUT, DOUBLE (LWORK)
         working array.
LWORK  : IN, INTEGER
         dimension of working array RWORK.
         The value of LWORK must be at least
         max(N*M, N*P, M*P, (N*N + 4*P + max(6*N,4*M,4*P)) ).
*     :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The state-space representations of the factors Q and R are constructed in the forms

$$Q = (Z'*(A+H*C)*Z,\ Z'*(B+H*D),\ C*Z,\ D)$$

and

$$R = (Z'*(A+H*B)*Z,\ Z'*H,\ C*Z,\ I)$$

where Z is an N x NQ matrix with orthonormal columns and H is an N x P output injection matrix assigning the eigenvalues of A in the specified stability region. If the given state-space representation is detectable, the order NQ of the resulting state-space representation of (Q R) is equal to N. If the given state-space representation is not detectable, the undetectable part of the original system is automatically deflated and the resulting NQ is less than N. The matrices AQR, BQR, CQR and DQR are computed as:

$$AQR = Z'*(A+H*C)*Z\ ,\quad BQR = (\ Z'*B+Z'*H*D\ \ Z'*H\ ),$$
$$CQR = C*Z\ ,\qquad\qquad DQR = (\ D\ I\ ).$$

The comprime factorization can be computed with (WITHD = .TRUE.) or without (WITHD = .FALSE.) an explicitly given feedthrough matrix D. In the second case, D = 0 is used for constructing the above matrices.

Literature

/1/ Varga A.
Coprime Factors Model Reduction Based on Accuracy Enhancing Techniques,
Systems Analysis, Modelling and Simulation, vol. 11, pp. 303-311, 1993.

Remarks:

- An NR order minimal state-space representation (AR,BR,CR,DR) of the factor R can be recovered from the computed matrices AQR, BQR, CQR and DQR as follows:

AR is the NR x NR submatrix formed by the rows 1, ..., NR and the columns 1, ..., NR of AQR (contained in A);
BR is the NR x M submatrix formed by the rows 1, ..., NR and columns M+1, ..., M+P of BQR (contained in B);
CR is the P x NR submatrix formed from the columns 1, ..., NR of CQR (contained in C);
DR is the P x P submatrix formed from the columns M+1, ..., M+P of DQR (contained in D).

- This routine is usually followed by one of the routines provided for performing model reduction of the stable systems. The resulting reduced model of the original system can be computed by using the subroutine RPMRLB to compute the transfer-function matrix corresponding to a left coprime factorization.

Copyright:

Life cycle:

1992 JUNE A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the continuous-time unstable system example (A,B,C,D) used by the subroutine RPMRSD (D = 0), a stable left coprime factorization can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      WITHD  = .FALSE.
      DISCR  = .FALSE.
      ALPHA  = 0.0D0
      BETA   = -1.0D0
      TOL    = 1.0D-10
      LWORK  =  N*N+4*P+6*N
      CALL RPMRLF(A, N, B, M, C, P, D, WITHD, DISCR, ALPHA, BETA,
     *            TOL, NQ, NR, RWORK, LWORK, *1111)
```

The matrices of the computed left coprime factorization are:

$$AQR = \begin{pmatrix} -1.0000 & -.4465 & 4.8212 & .2260 & .0062 & -.1813 & -.0895 \\ .0526 & -1.0000 & -.3364 & -.0166 & .4199 & -.2408 & -1.7274 \\ 0 & 0 & -3.5957 & -3.5463 & -.0163 & .0175 & .0592 \\ 0 & 0 & 0 & -12.4245 & .0004 & -.0344 & -.0180 \\ 0 & 0 & 0 & 0 & -13.1627 & 1.9835 & 3.6182 \\ 0 & 0 & 0 & 0 & 0 & -1.4178 & 5.6218 \\ 0 & 0 & 0 & 0 & 0 & -.8374 & -1.4178 \end{pmatrix}$$

```
      (    .0631   -.5122    .0155    .0753   1.1676 )
      (   1.1544    .0159    .2623   1.1297   -.0763 )
      (   -.0476    .3029        0        0        0 )
BQR = (    .0130  12.4858        0        0        0 )
      ( -11.7198    .0038        0        0        0 )
      (  -2.8173    .0308        0        0        0 )
      (   3.1018   -.0009        0        0        0 )

      ( -.0132   -.2238    .0079   -.0026   -.1279    .8797    .3994 )
CQR = ( -.0643   -.9639    .0040   -.0009   -.0305   -.2562    .0122 )
      ( -.9962    .0660    .0377   -.0419        0    .0022   -.0017 )

      ( 0  0  1  0  0 )
DQR = ( 0  0  0  1  0 )
      ( 0  0  0  0  1 )
```

For this example NQ = 7 and NR = 2.

Error Messages:

-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
The ordering of the real Schur failed.

SUBROUTINE RPMRRF

Right Coprime Factorization with Prescribed Stability Degree

Procedure purpose:

Given the matrices A, B, C and D of a state-space representation of a transfer-function matrix G, this subroutine computes the matrices AQR, BQR, CQR, DQR of a state-space representation of the transfer-function matrix $\begin{pmatrix} Q \\ R \end{pmatrix}$ such that $G = Q * R^{-1}$ is a stable rational right coprime factorization of G. The state matrix AQR can be determined such that its eigenvalues are inside a specified stability region of the complex plane. The computation of the factorization is based on a pole assignment method described in /1/.

Usage:

```
CALL RPMRRF(A, N, B, M, C, P, D, WITHD, DISCR, ALPHA, BETA, TOL,
            NQ, NR, RWORK, LWORK, *)
```

A : IN, OUT, DOUBLE (N,N)
On input : system state matrix A of the given state-space representation of G (column dense)
On output : the NQ*NQ state matrix AQR in real Schur form of $\begin{pmatrix} Q \\ R \end{pmatrix}$ the resulting state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

N : IN, INTEGER
dimension of state vector

B : IN, OUT, DOUBLE (N,M)
On input : system input matrix B of the given state-space representation of G (column dense)
On output : the NQ*M input matrix BQR of the resulting $\begin{pmatrix} Q \\ R \end{pmatrix}$ state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

M : IN, INTEGER
dimension of input vector

C : IN, OUT, DOUBLE (P+M,N)
On input : the P*N system output matrix C of the given state-space representation of G (column dense)
On output : the (P+M)*NQ output matrix CQR of the resulting $\begin{pmatrix} Q \\ R \end{pmatrix}$ state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

P : IN, INTEGER
dimension of output vector

D : IN, OUT, DOUBLE (P+M,M)
On input : If WITHD = .TRUE., the P*M system feedthrough matrix D of the given state-space representation of G. If WITHD = .FALSE, D is implicitly assumed a P x M null matrix in computing the matrices of the right coprime factorization. (column dense)
On output : the (P+M)*M feedthrough matrix DQR of the resulting state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

WITHD : IN, LOGICAL
specifies whether or not a given feedthrough matrix D is used in computing the factor Q of the coprime factorization:
WITHD = .TRUE. means a given feedthrough matrix D is used in determining Q.
WITHD = .FALSE. means D is assumed a null matrix in determining Q.

DISCR : IN, LOGICAL
specifies the type of the system:
continuous-time system, if DISCR = .FALSE., or
discrete-time system, if DISCR = .TRUE.

ALPHA : IN, DOUBLE
specifies the boundary of the stability region for the eigenvalues of A. For a continuous-time system (DISCR = .FALSE.), ALPHA is the maximum admissible value for the real parts of eigenvalues, while for a discrete-time system (DISCR = .TRUE.), ALPHA represents the maximum admissible value for the moduli of eigenvalues. The eigenvalues inside the stability region will not be modified by the pole assignment algorithm.
ALPHA >= 0 if DISCR = .TRUE. .

BETA : IN, DOUBLE
specifies the desired stability degree to be assigned for the eigenvalues of A outside the stability region defined by ALPHA. The eigenvalues outside the stability region will be assigned to have the real parts equal to BETA and unmodified imaginary parts for a continuous-time system (DISCR = .FALSE.) or moduli equal to BETA for a discrete-time system (DISCR = .TRUE.).

TOL : IN, DOUBLE
absolute tolerance level below which the elements of B are considered zero (used for controllability tests). Recommended value is TOL = N * EPS * NORM(B), where NORM(B) denotes the 1-norm of B and EPS is the machine precision (see LAPACK Library Routine DLAMCH).

NQ : OUT, INTEGER
the order of the resulting factors Q and R.
Generally NQ = N - NS, where NS is the number of uncontrollable eigenvalues outside the stability region.

NR : OUT, INTEGER

```
          the order of the minimal realization of the factor R.
          Generally NR is the number of controllable eigenvalues
          of A outside the stability region (the number of modified
          eigenvalues).
 RWORK :  OUT, DOUBLE (LWORK)
          working array.
 LWORK  : IN, INTEGER
          dimension of working array RWORK.
          The value of LWORK must be at least
          N*N+4*M+max(6*N,4*M,4*P).
 *      : RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The state-space representations of the factors Q and R are constructed in the forms

$$Q = (Z'*(A+B*F)*Z,\ Z'*B,\ (C+D*F)*Z,\ D)$$

and

$$R = (Z'*(A+B*F)*Z,\ Z'*B,\ F*Z,\ I)$$

where Z is an N x NQ matrix with orthonormal columns and F is an M x N state-feedback matrix assigning the eigenvalues of A in the specified stability region. If the given state-space representation is stabilizable, the order NQ of the resulting state-space representation $\begin{pmatrix} Q \\ R \end{pmatrix}$ is equal to N. If the given state-space representation is not stabilizable, the unstabilizable part of the original system is automatically deflated and the resulting NQ is less than N. The matrices AQR, BQR, CQR and DQR are computed as:

$$AQR = Z'*(A+B*F)*Z,\ BQR = Z'*B,\ CQR = \begin{pmatrix} C*Z+D*F*Z \\ F*Z \end{pmatrix},\ DQR = \begin{pmatrix} D \\ I \end{pmatrix}.$$

The comprime factorization can be computed with (WITHD = .TRUE.) or without (WITHD = .FALSE.) an explicitly given feedthrough matrix D. In the second case, D = 0 is used for constructing the above matrices.

Literature

/1/ Varga A.
Coprime Factors Model Reduction Based on Accuracy Enhancing Techniques,
Systems Analysis, Modelling and Simulation, vol. 11,
pp. 303-311, 1993.

Remarks:

- An NR order minimal state-space representation (AR,BR,CR,DR) of the factor R can be recovered from the computed matrices AQR, BQR, CQR and DQR as follows:

AR is the NR x NR submatrix formed by the rows NQ-NR+1, ..., NQ and the columns NQ-NR+1, ..., NQ of AQR (contained in A);
BR is the NR x M submatrix formed by the rows NQ-NR+1, ..., NQ of BQR (contained in B);
CR is the M x NR submatrix formed from the rows P+1, ..., P+M and the columns NQ-NR+1. ..., NQ of CQR (contained in C);
DR is the M x M submatrix formed from the rows P+1, ..., P+M of DQR (contained in D).

- This routine is usually followed by one of the routines provided for performing model reduction of the stable systems. The resulting reduced model of the original system can be computed by using the subroutine RPMRRB to compute the transfer-function matrix corresponding to a right coprime factorization.

Copyright:

Life cycle:

1992 JUNE A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the continuous-time unstable system example (A,B,C,D) used by the subroutine RPMRSD (D = 0), a right stable coprime factorization can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      WITHD  = .FALSE.
      DISCR  = .FALSE.
      ALPHA  = 0.0D0
      BETA   = -1.0D0
      TOL    = 1.0D-10
      LWORK  =  N*N+4*M+6*N
      CALL RPMRRF(A, N, B, M, C, P, D, WITHD, DISCR, ALPHA, BETA,
     *            TOL, NQ, NR, RWORK, LWORK, *1111)
```

The matrices of the computed right coprime factorization are:

$$AQR = \begin{pmatrix} -1.4178 & -5.1682 & 3.2450 & .0241 & .2232 & 5.4114 & .0201 \\ .9109 & -1.4178 & -2.1262 & -.1188 & -.0867 & .5383 & .2376 \\ 0 & 0 & -13.1627 & -.0011 & -.0646 & -6.8045 & .0315 \\ 0 & 0 & 0 & -12.4245 & 3.3373 & -33.5354 & -6.4031 \\ 0 & 0 & 0 & 0 & -3.5957 & 10.6960 & 6.6253 \\ 0 & 0 & 0 & 0 & 0 & -1.0000 & .0030 \\ 0 & 0 & 0 & 0 & 0 & -7.7653 & -1.0000 \end{pmatrix}$$

```
       (   5.0302   -.0063 )
       (    .7078   -.0409 )
       ( -11.3663    .0051 )
BQR =  (   -.0375 -11.6309 )
       (   -.1740   3.7681 )
       (  -1.1040   -.1956 )
       (   -.0472  -2.5948 )

       ( -.8659    .2787   -.3432   -.0007   -.0019   -.2335    .0152 )
       (  .0797   -.3951    .0976    .0045    .0295   -.9043    .0968 )
CQR =  ( -.0165   -.0645    .0097   -.1341   -.8080   -.0599   -.5666 )
       (      0        0        0        0        0    .4915   -.0036 )
       (      0        0        0        0        0   2.8707    .4690 )

       ( 0  0 )
       ( 0  0 )
DQR =  ( 0  0 )
       ( 1  0 )
       ( 0  1 )
```

For this example NQ = 7 and NR = 2.

Error Messages:

-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
The ordering of the real Schur failed.

SUBROUTINE RPMRLI

Left Coprime Factorization with Inner Denominator

Procedure purpose:

Given the matrices A, B, C and D of a state-space representation of a transfer-function matrix G, this subroutine computes the matrices AQR, BQR, CQR, DQR of a state-space representation of the transfer-function matrix $(Q\ R)$ such that $G = R^{-1} * Q$ is a stable rational left coprime factorization of G with a co-inner denominator R. The computation of the factorization is based on a dual version of the method proposed in /1/. G must not have poles on the imaginary axis for a continuous-time system or on the unit circle for a discrete-time system.

Usage:

```
CALL RPMRLI(A, N, B, M, C, P, D, WITHD, DISCR, TOL, NQ, NR,
            RWORK, LWORK, *)
```

A : IN, OUT, DOUBLE (N,N)
  On input : system state matrix A of the given state-space representation of G. A must not have controllable eigenvalues on the imaginary axis if DISCR = .FALSE. or on the unit circle if DISCR = .TRUE..
  (column dense)
  On output : the NQ*NQ state matrix AQR in real Schur form of the resulting state-space representation of (Q R).
  (column dense)

N : IN, INTEGER
  dimension of state vector

B : IN, OUT, DOUBLE (N,M+P)
  On input : the N*M system input matrix B of the given state-space representation of G
  (column dense)
  On output : the NQ*(M+P) input matrix BQR of the resulting state-space representation of (Q R).
  (column dense)

M : IN, INTEGER
  dimension of input vector

C : IN, OUT, DOUBLE (P,N)
  On input : the P*N system output matrix C of the given state-space representation of G
  (column dense)
  On output : the P*NQ output matrix CQR of the resulting state-space representation of (Q R).
  (column dense)

P : IN, INTEGER

```
          dimension of output vector
D      :  IN, OUT, DOUBLE (P,M+P)
          On input  : If WITHD = .TRUE., the P*M system feedthrough
                      matrix D of the given state-space representation
                      of G. If WITHD = .FALSE, D is implicitly assumed
                      a P x M null matrix in computing the matrices
                      of the left coprime factorization.
                      (column dense)
          On output : the P*(M+P) feedthrough matrix DQR of the
                      resulting state-space representation of ( Q R ).
                      (column dense)
WITHD  :  IN, LOGICAL
          specifies whether or not a given feedthrough matrix D is used
          in computing the factor Q of the coprime factorization:
          WITHD = .TRUE.  means a given feedthrough matrix D is used in
                          determining Q.
          WITHD = .FALSE. means D is assumed a null matrix in
                          determining Q.
DISCR  :  IN, LOGICAL
          specifies the type of the system:
          continuous-time system, if DISCR = .FALSE., or
          discrete-time system, if DISCR = .TRUE.
TOL    :  IN, DOUBLE
          absolute tolerance level below which the elements of
          C are considered zero (used for observability tests).
          Recommended value is TOL = N * EPS * NORM(C), where
          NORM(C) denotes the infinity-norm of C and EPS is the machine
          precision (see LAPACK Library Routine DLAMCH).
NQ     :  OUT, INTEGER
          the order of the resulting factors Q and R.
          Generally NQ = N - NS, where NS is the number of
          undetectable eigenvalues outside the stability region.
NR     :  OUT, INTEGER
          the order of the minimal realization of the factor R.
          Generally NR is the number of observable eigenvalues
          of A outside the stable region (the number of modified
          eigenvalues).
RWORK  :  OUT, DOUBLE (LWORK)
          working array.
LWORK  :  IN, INTEGER
          dimension of working array RWORK.
          The value of LWORK must be at least
          max(N*M, N*P, M*P, (N*N + P*(P+2) + max(6*N,4*M,4*P)) ).
*      :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The state-space representations of the factors Q and R are constructed in the forms

```
     Q = (Z'*(A+H*C)*Z, Z'*(B+H*D), V*C*Z, V*D)
and
     R = (Z'*(A+H*B)*Z, Z'*H, V*C*Z, V)

where Z is an N x NQ matrix with orthonormal columns, H is an N x P
output injection matrix reflecting the unstable eigenvalues of A in
the stability region, and V is a P x P gain matrix. The resulting R
is generally non-minimal and co-inner, that is R(s)*R'(-s) = I in the
continuous-time case or R(z)*R'(1/z) = I in the discrete-time case.
If the given state-space representation is detectable, the order
NQ of the resulting state-space representation of ( Q R ) is equal to
N. If the given state-space representation is not detectable, the
undetectable part of the original system is automatically deflated and
the resulting NQ is less than N. The matrices AQR, BQR, CQR and DQR
are computed as:

AQR = Z'*(A+H*C)*Z , BQR = ( Z'*B+Z'*H*D  Z'*H ),
CQR = V*C*Z ,        DQR = ( V*D  V ).

The comprime factorization can be computed with (WITHD = .TRUE.) or
without (WITHD = .FALSE.) an explicitly given feedthrough matrix D.
In the second case, D = 0 is used for constructing the above matrices.

Literature
/1/ Varga A.
    A Schur method for computing coprime factorizations
    with inner denominators and applications in model reduction,
    Proc. 1993 ACC, San Francisco, CA, pp. 2130-2131, 1993.
```

Remarks:

- No explicit checks are performed to detect eigenvalues of A which are near to the imaginary axis if DISCR = .FALSE. or to the unit circle if DISCR = .TRUE.. For such systems, the problem is ill-conditioned and the computed results may be inaccurate.
- An NR order minimal state-space representation (AR,BR,CR,DR) of the factor R which is simultaneously inner and co-inner can be recovered from the computed matrices AQR, BQR, CQR and DQR as follows:
  AR is the NR x NR submatrix formed by the rows 1, ..., NR and the columns 1, ..., NR of AQR (contained in A);
  BR is the NR x M submatrix formed by the rows 1, ..., NR and columns M+1, ..., M+P of BQR (contained in B);
  CR is the P x NR submatrix formed from the columns 1, ..., NR of CQR (contained in C);
  DR is the P x P submatrix formed from the columns M+1, ..., M+P of DQR (contained in D).
- This routine is usually followed by one of the routines provided for performing model reduction of the stable systems. The resulting reduced model of the original system can be computed by using the subroutine RPMRLB to compute the transfer-function matrix corresponding to a left coprime factorization.

- The (co)innerness property of the denominator is preserved when using either the Balance & Truncate, or the Singular Perturbation Approximation, or the Balanced Stochastic Truncation methods, to compute a reduced order approximation of the factors of a coprime factorization with inner denominator of a continuous transfer-function matrix /1/. In the discrete-time case, the (co)innerness of the denominator is preserved only by the Singular Perturbation Approximation method /1/.

Life cycle:
1992 JULY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
Given the continuous-time unstable system example (A,B,C,D) used by the subroutine RPMRSD (D = 0), a stable left coprime factorization with inner denominator can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      WITHD  = .FALSE.
      DISCR  = .FALSE.
      TOL    = 1.0D-10
      LWORK  =  N*N+P*(P+2)+6*N
      CALL RPMRLI(A, N, B, M, C, P, D, WITHD, DISCR, TOL, NQ, NR,
     *            RWORK, LWORK, *1111)
```

The matrices of the computed left coprime factorization are:

$$AQR = \begin{pmatrix} -.1605 & -.4489 & 4.2621 & .2229 & -.2394 & -.0491 & .8740 \\ .0523 & -.1605 & 2.2250 & .1217 & .4166 & -.2518 & -1.6140 \\ 0 & 0 & -3.5957 & -3.5463 & -.0163 & .0175 & .0592 \\ 0 & 0 & 0 & -12.4245 & .0004 & -.0344 & -.0180 \\ 0 & 0 & 0 & 0 & -13.1627 & 1.9835 & 3.6182 \\ 0 & 0 & 0 & 0 & 0 & -1.4178 & 5.6218 \\ 0 & 0 & 0 & 0 & 0 & -.8374 & -1.4178 \end{pmatrix}$$

```
      (   -.5523   -.4443   -.0306   -.1281    .4984 )
      (   1.0157   -.2554    .0158    .0692    .1688 )
      (   -.0476    .3029        0        0        0 )
BQR = (    .0130  12.4858        0        0        0 )
      ( -11.7198    .0038        0        0        0 )
      (  -2.8173    .0308        0        0        0 )
      (   3.1018   -.0009        0        0        0 )

      (  .1063   -.1975    .0079   -.0026   -.1279    .8797    .3994 )
CQR = (  .4513   -.8541    .0040   -.0009   -.0305   -.2562    .0122 )
      ( -.8826   -.4668    .0377   -.0419        0    .0022   -.0017 )

      ( 0   0   1   0   0 )
DQR = ( 0   0   0   1   0 )
      ( 0   0   0   0   1 )
```

The minimal state-space realization of the inner denominator R has the following matrices:

```
AR = ( -.1605   -.4489 )   BR = ( -.0306   -.1281    .4984 )
     (  .0523   -.1605 )        (  .0158    .0692    .1688 )

     (  .1063   -.1975 )        ( 1   0   0 )
CR = (  .4513   -.8541 )   DR = ( 0   1   0 )
     ( -.8826   -.4668 )        ( 0   0   1 )
```

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
The ordering of the real Schur failed.
-4-
The system (A,B,C,D) is not factorable.
```

SUBROUTINE RPMRRI

Right Coprime Factorization with Inner Denominator

Procedure purpose:

Given the matrices A, B, C and D of a state-space representation of a transfer-function matrix G, this subroutine computes the matrices AQR, BQR, CQR, DQR of a state-space representation of the

$$\begin{pmatrix} Q \\ R \end{pmatrix}$$

transfer-function matrix such that $G = Q * R^{-1}$ is a stable rational right coprime factorization with inner denominator of G. The computation of the factorization is based on the method described in /1/. G must not have poles on the imaginary axis for a continuous-time system or on the unit circle for a discrete-time system.

Usage:

```
CALL RPMRRI(A, N, B, M, C, P, D, WITHD, DISCR, TOL, NQ, NR,
            RWORK, LWORK, *)
```

A : IN, OUT, DOUBLE (N,N)
- On input : system state matrix A of the given state-space representation of G. A must not have controllable eigenvalues on the imaginary axis if DISCR = .FALSE. or on the unit circle if DISCR = .TRUE.. (column dense)
- On output : the NQ*NQ state matrix AQR in real Schur form of $\begin{pmatrix} Q \\ R \end{pmatrix}$ the resulting state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

N : IN, INTEGER
dimension of state vector

B : IN, OUT, DOUBLE (N,M)
- On input : the N*M system input matrix B of the given state-space representation of G (column dense)
- On output : the NQ*M input matrix BQR of the resulting $\begin{pmatrix} Q \\ R \end{pmatrix}$ state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

M : IN, INTEGER
dimension of input vector

C : IN, OUT, DOUBLE (P+M,N)
- On input : the P*N system output matrix C of the given state-space representation of G (column dense)
- On output : the (P+M)*NQ output matrix CQR of the resulting $\begin{pmatrix} Q \\ R \end{pmatrix}$ state-space representation of $\begin{pmatrix} Q \\ R \end{pmatrix}$. (column dense)

```
P     : IN, INTEGER
        dimension of output vector
D     : IN, OUT, DOUBLE (P+M,M)
        On input  : If WITHD = .TRUE., the P*M system feedthrough
                    matrix D of the given state-space representation
                    of G. If WITHD = .FALSE, D is implicitly assumed
                    a P x M null matrix in computing the matrices
                    of the right coprime factorization.
                    (column dense)
        On output : the (P+M)*M feedthrough matrix DQR of the
                                                          ( Q )
                    resulting state-space representation of ( R ).
                    (column dense)
WITHD : IN, LOGICAL
        specifies whether or not a given feedthrough matrix D is used
        in computing the factor Q of the coprime factorization:
        WITHD = .TRUE.  means a given feedthrough matrix D is used in
                        determining Q.
        WITHD = .FALSE. means D is assumed a null matrix in
                        determining Q.
DISCR : IN, LOGICAL
        specifies the type of the system:
        continuous-time system, if DISCR = .FALSE., or
        discrete-time system, if DISCR = .TRUE.
TOL   : IN, DOUBLE
        absolute tolerance level below which the elements of
        B are considered zero (used for controllability tests).
        Recommended value is TOL = N * EPS * NORM(B), where
        NORM(B) denotes the 1-norm of B and EPS is the machine
        precision (see LAPACK Library Routine DLAMCH).
NQ    : OUT, INTEGER
        the order of the resulting factors Q and R.
        Generally NQ = N - NS, where NS is the number of
        uncontrollable eigenvalues outside the stability region.
NR    : OUT, INTEGER
        the order of the minimal realization of the factor R.
        Generally NR is the number of controllable eigenvalues
        of A outside the stable region (the number of modified
        eigenvalues).
RWORK : OUT, DOUBLE (LWORK)
        working array.
LWORK : IN, INTEGER
        dimension of working array RWORK.
        The value of LWORK must be at least
        N*N+M*(M+2)+max(6*N,4*M,4*P).
*     : RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The state-space representations of the factors Q and R are constructed in the forms

$$Q = (Z'*(A+B*F)*Z,\ Z'*B*V,\ (C+D*F)*Z,\ D*V)$$

and

$$R = (Z'*(A+B*F)*Z,\ Z'*B*V,\ F*Z,\ V)$$

where Z is an N x NQ matrix with orthonormal columns, F is an M x N state-feedback matrix reflecting the unstable eigenvalues of A in the stability region and V is a P x P gain matrix. The resulting R is inner, that is R'(-s)*R(s) = I in the continuous-time case or R'(1/z)*R(z) = I in the discrete-time case.
If the given state-space representation is stabilizable, the order NQ of the resulting state-space representation

$$\begin{pmatrix} Q \\ R \end{pmatrix}$$

is equal to N. If the given state-space representation is not stabilizable, the unstabilizable part of the original system is automatically deflated and the resulting NQ is less than N.
The matrices AQR, BQR, CQR and DQR are computed as:

$$AQR = Z'*(A+B*F)*Z,\ BQR = Z'*B*V,\ CQR = \begin{pmatrix} C*Z+D*F*Z \\ F*Z \end{pmatrix},\ DQR = \begin{pmatrix} D*V \\ V \end{pmatrix}.$$

The comprime factorization can be computed with (WITHD = .TRUE.) or without (WITHD = .FALSE.) an explicitly given feedthrough matrix D. In the second case, D = 0 is used for constructing the above matrices.

Literature

/1/ Varga A.
A Schur method for computing coprime factorizations with inner denominators and applications in model reduction, Proc. 1993 ACC, San Francisco, CA, pp. 2130-2131, 1993.

Remarks:

- No explicit checks are performed to detect eigenvalues of A which are near to the imaginary axis if DISCR = .FALSE. or to the unit circle if DISCR = .TRUE.. For such systems, the problem is ill-conditioned and the computed results may be inaccurate.
- An NR order minimal state-space representation (AR,BR,CR,DR) of the factor R can be recovered from the computed matrices AQR, BQR, CQR and DQR as follows:
  AR is the NR x NR submatrix formed by the rows NQ-NR+1, ..., NQ and the columns NQ-NR+1, ..., NQ of AQR (contained in A);
  BR is the NR x M submatrix formed by the rows NQ-NR+1, ..., NQ of BQR (contained in B);
  CR is the M x NR submatrix formed from the rows P+1, ..., P+M and the columns NQ-NR+1. ..., NQ of CQR (contained in C);
  DR is the M x M submatrix formed from the rows P+1, ..., P+M of DQR (contained in D).
- This routine is usually followed by one of the routines provided for performing model reduction of the stable systems. The resulting

reduced model of the original system can be computed by using the subroutine RPMRRB to compute the transfer-function matrix corresponding to a right coprime factorization.

- The innerness property of the denominator is preserved when using either the Balance & Truncate, or the Singular Perturbation Approximation, or the Balanced Stochastic Truncation methods, to compute a reduced order approximation of the factors of a coprime factorization with inner denominator of a continuous transfer-function matrix /1/. In the discrete-time case, the innerness of the denominator is preserved only by the Singular Perturbation Approximation method /1/.

Copyright:

1992 - DLR Institut fuer Dynamik der Flugsysteme

Life cycle:

1992 JULY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the continuous-time unstable system example (A,B,C,D) used by the subroutine RPMRSD (D = 0), a right stable coprime factorization can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      WITHD  = .FALSE.
      DISCR  = .FALSE.
      TOL    = 1.0D-10
      LWORK  =  N*N+M*(M+4)+6*N
      CALL RPMRRI(A, N, B, M, C, P, D, WITHD, DISCR, TOL, NQ, NR,
     *            RWORK, LWORK, *1111)
```

The matrices of the computed right coprime factorization are:

```
      ( -1.4178  -5.1682   3.2450    .0241    .2232   4.1066   -.2336 )
      (   .9109  -1.4178  -2.1262   -.1188   -.0867    .4816    .2196 )
      (       0        0 -13.1627   -.0011   -.0646  -3.8320    .3429 )
AQR = (       0        0       0 -12.4245   3.3373   -.2642  -2.6816 )
      (       0        0       0        0  -3.5957    .1871   5.4221 )
      (       0        0       0        0        0   -.1605    .0772 )
      (       0        0       0        0        0   -.3040   -.1605 )
```

```
       (   5.0302   -.0063 )
       (    .7078   -.0409 )
       ( -11.3663    .0051 )
BQR =  (   -.0375 -11.6309 )
       (   -.1740   3.7681 )
       (  -1.1050   -.3215 )
       (    .0066  -2.5822 )

       ( -.8659    .2787   -.3432   -.0007   -.0019   -.2325    .0265 )
       (  .0797   -.3951    .0976    .0045    .0295   -.8985    .1406 )
CQR =  ( -.0165   -.0645    .0097   -.1341   -.8080   -.0874   -.5630 )
       (      0        0        0        0        0    .2288   -.0259 )
       (      0        0        0        0        0    .0070    .1497 )

       ( 0  0 )
       ( 0  0 )
DQR =  ( 0  0 )
       ( 1  0 )
       ( 0  1 )
```

The minimal state-space realization of the inner denominator R has the following matrices:

```
AR =  ( -.1605    .0772 )   BR = ( -1.1050   -.3215 )
      ( -.3040   -.1605 )        (   .0066  -2.5822 )

CR =  ( .2288   -.0259 )    DR = ( 1   0 )
      ( .0070    .1497 )         ( 0   1 )
```

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
The ordering of the real Schur failed.
-4-
The system (A,B,C,D) is not factorable.
```

SUBROUTINE RPMRLB

State-Space Representation of a Left Coprime Factorization

Procedure purpose:

Constructs the state-space representation for the system G = (A,B,C,D) from the factors Q = (AQR,BQ,CQR,DQ) and R = (AQR,BR,CQR,DR) of its left coprime factorization

$$G = R^{-1} * Q$$

where G, Q and R are the corresponding transfer-function matrices.

Usage:

```
CALL RPMRLB(A, N, B, M, C, P, D, IWORK, *)
```

```
A     :  IN, OUT, DOUBLE (N,N)
         On input  : the N*N state matrix AQR of the state-space
                     representation of ( Q R ).
                     (column dense)
         On output : the state matrix A of the state-space
                     representation of G
                     (column dense)
N     :  IN, INTEGER
         dimension of state vector
B     :  IN, OUT, DOUBLE (N,M+P)
         On input  : the N*(M+P) input matrix ( BQ BR ) of the
                     state-space representation of ( Q R ).
                     (column dense)
         On output : the N*M input matrix B of the state-space
                     representation of G
                     (column dense)
M     :  IN, INTEGER
         dimension of input vector
C     :  IN, OUT, DOUBLE (P,N)
         On input  : the P*N output matrix ( CQR ) of the state-space
                     representation of ( Q R ) .
                     (column dense)
         On output : the P*N output matrix C of the state-space
                     representation of G
                     (column dense)
P     :  IN, INTEGER
         dimension of output vector
D     :  IN, OUT, DOUBLE (P,M+P)
         On input  : the P*(M+P) feedthrough matrix ( DQ DR ) of the
                     state-space representation of ( Q R ) .
                     (column dense)
         On output : the P*M output matrix D of the state-space
                     representation of G
```

```
                    (column dense)
IWORK :  OUT, INTEGER (P)
         working array.
*     :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The subroutine computes the matrices of the state-space representation G = (A,B,C,D) by using the formulas:

$$A = AQR - BR * DR^{-1} * CQR, \quad B = BQ - BR * DR^{-1} * DQ,$$

$$C = DR^{-1} * CQR, \quad D = DR^{-1} * DQ.$$

Literature

/1/ Varga A.
Coprime Factors Model Reduction Method based on Square-Root Balancing-Free Techniques,
Proc. 4-th IMACS Symp. on Systems Analysis and Simulation,
Berlin, Aug. 25-28, 1992.

Remarks:

- This routine is thought to be used in conjunction with one of the routines for computing left coprime factorizations of transfer-function matrices and therefore, the input matrices of this subroutine have identical structures with the structures of the matrices resulting from the factorization routines.

Copyright:

Life cycle:

1992 JULY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the continuous-time unstable system example (A,B,C,D) used by the subroutine RPMRSD (D = 0), a fifth order approximate model can be computed with the following sequence of statements implementing the coprime factors model reduction approach based on the square-root balancing-free B & T method:

```
N      = 7
M      = 2
P      = 3
WITHD  = .FALSE.
```

```
      DISCR  = .FALSE.
      ALPHA  = 0.0D0
      BETA   = -1.0D0
      TOL    = 1.0D-10
      LWORK  =  MAX(N*N+4*M+6*N, 4*(N+1)*N)
C     Compute the left coprime factorization of G.
      CALL RPMRLF(A, N, B, M, C, P, D, WITHD, DISCR, ALPHA, BETA,
     *            TOL, NQ, NR, RWORK, LWORK, *1111)
C     Compute the reduced order approximation of the coprime factors.
      TOL    = 0.0D0
      FIXORD = 5
      CALL RPMRBT(A, NQ, B, M, C, P+M, DISCR, TOL, FIXORD, NR, HSV,
     *            RWORK, LWORK, IWORK, *1111)
C     Compute the reduced order approximation of G.
      CALL RPMRLB(A, NR, B, M, C, P, D, IWORK, *1111)
```

The matrices AQR, BQR, CQR and DQR of the computed left coprime factorization (Q R) are those computed in the example presented for the subroutine RPMRLF.

The matrices AQRr, BQRr and CQRr of the reduced order approximation (Qr Rr) of (Q R) computed with the subroutine RPMRBT are:

```
       (  -.2054   -.1837   -.0522   -.9910    .4825 )
       (   .0475    .5169   1.8402   -.2473   -.2658 )
AQRr = (   .0197  -4.0359  -3.2528    .8956    .8612 )
       (  1.5121   -.5515   -.4983  -2.4071   1.2362 )
       ( -1.2927   -.4066   -.5290   1.3998  -1.9715 )

       (  .0704   2.1236    .0072    .0237   -.9778 )
       (  .9558    .0428   -.0845   -.3647   -.0567 )
BQRr = ( 5.3424    .2182    .0351    .1518    .0544 )
       (  .7745  -4.2105   -.1703   -.7373   -.4817 )
       (  .5534   3.8290   -.1925   -.8260    .4379 )

       ( -.0555    .9341   -.2686   -.0784   -.0500 )
CQRr = ( -.0096    .1070   -.1771    .6179    .6916 )
       (  .8856    .0512   -.0368    .3138   -.2840 )
```

The Hankel singular values of the system (Q R) are:

```
2.8349   1.9641    .7710    .5583    .4518    .0237    .0201
```

The matrices Ar, Br and Cr of the reduced order approximation of the original system are:

$$Ar = \begin{pmatrix} .6612 & -.1428 & -.0821 & -.6983 & .1887 \\ .0895 & .6377 & 1.7508 & -.0108 & -.0339 \\ -.0251 & -4.0878 & -3.2145 & .7875 & .7735 \\ 1.9222 & -.2888 & -.6924 & -1.8137 & 1.6008 \\ -1.6991 & -.1609 & -.7109 & 1.7577 & -1.2855 \end{pmatrix}$$

$$Br = \begin{pmatrix} .0704 & 2.1236 \\ .9558 & .0428 \\ 5.3424 & .2182 \\ .7745 & -4.2105 \\ .5534 & 3.8290 \end{pmatrix}$$

$$Cr = \begin{pmatrix} -.0555 & .9341 & -.2686 & -.0784 & -.0500 \\ -.0096 & .1070 & -.1771 & .6179 & .6916 \\ .8856 & .0512 & -.0368 & .3138 & -.2840 \end{pmatrix}$$

Error Messages:

-1-
Invalid parameter value on entry.
-2-
Singular denominator feedthrough matrix.

SUBROUTINE RPMRRB

State-Space Representation of a Right Coprime Factorization

Procedure purpose:

Constructs the state-space representation for the system G = (A,B,C,D) from the factors Q = (AQR,BQR,CQ,DQ) and R = (AQR,BQR,CR,DR) of its right coprime factorization

$$G = Q * R^{-1}$$

where G, Q and R are the corresponding transfer-function matrices.

Usage:

```
CALL RPMRRB(A, N, B, M, C, P, D, RWORK, LWORK, IWORK, *)
```

```
A     :  IN, OUT, DOUBLE (N,N)
         On input  : the N*N state matrix AQR of the state-space
                                          ( Q )
                     representation of    ( R ).
                     (column dense)
         On output : the N*N state matrix A of the state-space
                     representation of G
                     (column dense)
N     :  IN, INTEGER
         dimension of state vector
B     :  IN, OUT, DOUBLE (N,M)
         On input  : the N*M input matrix BQR of the state-space
                                          ( Q )
                     representation of    ( R ).
                     (column dense)
         On output : the N*M input matrix B of the state-space
                     representation of G
                     (column dense)
M     :  IN, INTEGER
         dimension of input vector
C     :  IN, OUT, DOUBLE (P+M,N)
                                                ( CQ )
         On input  : the (P+M)*N output matrix  ( CR ) of the state-space
                                          ( Q )
                     representation of    ( R ) .
                     (column dense)
         On output : the P*N output matrix C of the state-space
                     representation of G
                     (column dense)
P     :  IN, INTEGER
         dimension of output vector

D     :  IN, OUT, DOUBLE (P+M,M)
```

```
                                                        ( DQ )
          On input  : the (P+M)*M feedthrough matrix    ( DR ) of the
                                                        ( Q )
                      state-space representation of     ( R ) .
                      (column dense)
          On output : the P*M output matrix D of the state-space
                      representation of G
                      (column dense)
 RWORK :  OUT, DOUBLE (LWORK)
          working array.
 LWORK  : IN, INTEGER
          dimension of working array RWORK.
          The value of LWORK must be at least M.
 IWORK :  OUT, INTEGER (M)
          working array.
 *      : RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The subroutine computes the matrices of the state-space representation G = (A,B,C,D) by using the formulas:

$$A = AQR - BQR * DR^{-1} * CR, \quad B = BQR * DR^{-1},$$
$$C = CQ - DQ * DR^{-1} * CR, \quad D = DQ * DR^{-1}.$$

Literature

/1/ Varga A.
Coprime Factors Model Reduction Method based on Square-Root Balancing-Free Techniques,
Proc. 4-th IMACS Symp. on Systems Analysis and Simulation, Berlin, Aug. 25-28, 1992.

Remarks:

- This routine is thought to be used in conjunction with one of the routines for computing right coprime factorizations of transfer-function matrices and therefore, the input matrices of this subroutine have identical structures with the structures of the matrices resulting from the factorization routines.

Copyright:

Life cycle:

1992 JULY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Given the continuous-time unstable system example (A,B,C,D) used by the subroutine RPMRSD (D = 0), a fifth order approximate model can be computed with the following sequence of statements implementing the coprime factors model reduction approach based on the square-root balancing-free B & T method:

```
      N      = 7
      M      = 2
      P      = 3
      WITHD  = .FALSE.
      DISCR  = .FALSE.
      ALPHA  = 0.0D0
      BETA   = -1.0D0
      TOL    = 1.0D-10
      LWORK  =  MAX(N*N+4*M+6*N, 4*(N+1)*N)
C     Compute the right coprime factorization of G.
      CALL RPMRRF(A, N, B, M, C, P, D, WITHD, DISCR, ALPHA, BETA,
     *            TOL, NQ, NR, RWORK, LWORK, *1111)
C     Compute the reduced order approximation of the coprime factors.
      TOL    = 0.0D0
      FIXORD = 5
      CALL RPMRBT(A, NQ, B, M, C, P+M, DISCR, TOL, FIXORD, NR, HSV,
     *            RWORK, LWORK, IWORK, *1111)
C     Compute the reduced order approximation of G.
      CALL RPMRRB(A, NR, B, M, C, P, D, RWORK, LWORK, IWORK, *1111)
```

The matrices AQR, BQR, CQR and DQR of the computed right coprime factorization $\begin{pmatrix} Q \\ R \end{pmatrix}$ are those computed in the example presented for the subroutine RPMRRF.

The matrices AQRr, BQRr and CQRr of the reduced order approximation $\begin{pmatrix} Qr \\ Rr \end{pmatrix}$ of $\begin{pmatrix} Q \\ R \end{pmatrix}$ computed with the subroutine RPMRBT are:

$$AQRr = \begin{pmatrix} .4395 & -2.1330 & .0840 & .5701 & 1.2834 \\ .4486 & .0632 & -2.6514 & 4.4916 & 2.8834 \\ .3124 & -2.2932 & -2.1209 & -.6210 & -2.2299 \\ .4282 & -3.6434 & -1.5460 & -3.1359 & -3.4620 \\ -.1693 & -.0617 & 6.0877 & -4.3216 & -2.4482 \end{pmatrix}$$

$$BQRr = \begin{pmatrix} -.1752 & .0288 \\ -2.0423 & 3.3941 \\ -3.1532 & .2849 \\ -4.7751 & -1.2608 \\ .5936 & -3.8651 \end{pmatrix}$$

```
        (  .1886   -.6605    .2856    .1344   -.6076 )
        (  .1436   -.0582    .6008   -.3669    .1148 )
CQRr =  ( -.9659   -.1880    .1632    .0212   -.0313 )
        ( -.0523    .1432   -.1820    .1834    .0739 )
        (  .0060    .6962  -1.0518   1.1370    .6083 )

                                              ( Q )
The Hankel singular values of the system      ( R )  are:

9.8484   3.1500    .7672    .6939    .4023    .0224    .0194

The matrices Ar, Br and Cr of the reduced order approximation of
the original system are:

     (  .4302  -2.1280    .0824    .5695   1.2788 )
     (  .3214  -2.0071    .5469   1.0071    .9696 )
Ar = (  .1457  -2.0399  -2.3952   -.3667  -2.1703 )
     (  .1858  -2.0817  -3.7413   -.8267  -2.3424 )
     ( -.1151   2.5441   2.1302   -.0358   -.1407 )

     (  -.1752    .0288 )
     ( -2.0423   3.3941 )
Br = ( -3.1532    .2849 )
     ( -4.7751  -1.2608 )
     (   .5936  -3.8651 )

     (  .1886   -.6605    .2856    .1344   -.6076 )
Cr = (  .1436   -.0582    .6008   -.3669    .1148 )
     ( -.9659   -.1880    .1632    .0212   -.0313 )
```

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
Singular denominator feedthrough matrix.
```

3. Frequency-Weighted Order Reduction of Stable State-Space Systems

The subroutines of this chapter can be used to compute reduced order approximations of stable state-space systems by using the frequency-weighting approach. The provided subroutines represent tools by which weighting factors can be added to the given system model or can be removed from it. The routines can be used for both continuous-time and discrete-time systems.

If G is a given p x m stable transfer-function matrix of degree n, and W_1 and W_2 are p x p and m x m stable, invertible and minimum-phase transfer-function matrices of degrees n_1 and n_2, respectively, then the following procedure can be used to compute an r-th order approximation G_r of G by employing the frequency-weighting approach proposed in /1/:

1) Compute G_1, the n-th order stable projection of $(W_1^*)^{-1}G(W_2^*)^{-1}$.

2) Determine G_{1r}, an r-th degree approximation of G_1 by using a model reduction method for stable systems.

3) Compute G_r as the r-th order stable projection of $W_1^*G_{1r}W_2^*$.

Note: In the above expressions W^* represents either $W^T(-s)$ for a continuous-time system or $W^T(1/z)$ for a discrete-time system.

The frequency-weighting technique was originally proposed to be used in conjunction with the optimal Hankel-norm approximation method of /2/. In this case, the optimal weighted approximation error satisfies

$$\| \ (W_1^*)^{-1}(G - G_r)(W_2^*)^{-1}\|_H \ = \ \sigma_{r+1} \ ,$$

where σ_{r+1} is the $(r+1)$-th Hankel singular values of G_1, the transfer-function matrix computed at step 1 of the above procedure, and $\|G\|_H$ is the Hankel-norm of G.

For the model reduction at step 2, any of methods available for stable systems can be used. The following routines are provided to perform the computations at steps 1 and 3:

RPMRFL constructs for either $(W_1^*)^{-1}G$ or W_1^*G an n-th order state-space realization of its stable projection by using the explicit formulas derived in /3/.

RPMRFR constructs for either $G(W_1^*)^{-1}$ or GW_1^* an n-th order state-space realization of its stable projection by using the explicit formulas derived in /3/.

Literature:

/1/ Latham, G.A., Anderson, B.D.O.
Frequency-weighted optimal Hankel-norm approximation of stable transfer functions,
Systems & Control Letters, Vol. 5, pp. 229-236, 1985.

/2/ Glover, K.
All Optimal Hankel Norm Approximation of Linear Multivariable Systems and Their L-Infinity Error Bounds,
Int. J. Control, Vol. 36, pp. 1145-1193, 1984.

/3/ Varga A.
Explicit formulas for an efficient implementation of the frequency-weighting model reduction approach,
Proc. 1993 ECC, Groningen, NL, pp. 693-696, 1993.

SUBROUTINE RPMRFL

Stable Projections for Output Frequency-Weighted Model Reduction

Procedure purpose:

Constructs a state-space representation (AS,BS,CS,DS) of the
stable projection of either (W^{*-1})*G or W*G from the state-space representations (A,B,C,D) and (AW,BW,CW,DW) of the transfer-function matrices G and W, respectively. G should be stable. For the premultiplication with the conjugate of W, the system W used as frequency weight, should be stable and in the discrete-time case, W should have no poles in the origine. For the premultiplication with the inverse of the conjugate of W, the system W should be invertible and minimum-phase, and for the discrete-time case, W should have no zeros in the origine.

W* denotes the conjugate of W given by either W'(-s) for a continuous-time system or W'(1/z) for a discrete-time system.

Usage:

```
CALL RPMRFL(A, N, B, M, C, P, D, WITHD, DISCR, AW, NW,
   *        BW, CW, DW, INVW, RWORK, LWORK, IWORK, *)
```

```
A     :  IN, OUT, DOUBLE (N,N)
         On input  : system state matrix A of the system G
                     (column dense)
         On output : system state-matrix AS of the stable projection
                          * -1                          *
                     of (W )  *G if INVW = .TRUE. or of W *G if
                     INVW = .FALSE.. The resulting AS is in a real
                     Schur form.
                     (column dense)
N     :  IN, INTEGER, N > 0
         dimension of state vector of the system G
B     :  IN, OUT, DOUBLE (N,M)
         On input  : system input matrix B of the system G
                     (column dense)
         On output : system input matrix BS of the stable projection
                          * -1                          *
                     of (W )  *G if INVW = .TRUE. or of W *G if
                     INVW = .FALSE..
                     (column dense)
M     :  IN, INTEGER, M > 0
         dimension of input vector of the system G.
C     :  IN, OUT, DOUBLE (P,N)
         On input  : system output matrix C of the system G
                     (column dense)
         On output : system output matrix CS of the stable projection
```

```
                        * -1                                  *
                    of (W )  *G if INVW = .TRUE. or of W *G if
                    INVW = .FALSE..
                    (column dense)
P       :   IN, INTEGER, P > 0
            dimension of output vectors of systems G and W and of the
            input vector of the system W.
D       :   IN, OUT, DOUBLE (P,M)
            On input  : if WITHD = .TRUE., the system feedthrough matrix
                        D of the system G
                        (column dense)
            On output : if WITHD = .TRUE., the system output matrix DS
                        of the stable projection
                        * -1                                  *
                    of (W )  *G if INVW = .TRUE. or of W *G if
                    INVW = .FALSE..
                    (column dense)
            If WITHD = .FALSE., this matrix is not referenced.
WITHD   :   IN, LOGICAL
            specifies whether or not the feedthrough matrix D is null:
            WITHD = .TRUE.,  a non-zero D is assumed.
            WITHD = .FALSE., a zero D matrix is assumed.
DISCR   :   IN, LOGICAL
            specifies the type of the systems G and W:
            continuous-time systems, if DISCR = .FALSE., or
            discrete-time systems, if DISCR = .TRUE.
AW      :   IN, DOUBLE (NW,NW)
            system state-matrix AW of the system W
            (column dense)
            If NW = 0, this matrix is not referenced.
NW      :   IN, INTEGER, NW >= 0
            dimension of state vector of the system W
BW      :   IN, DOUBLE (NW,P)
            system input matrix BW of the system W
            (column dense)
            If NW = 0, this matrix is not referenced.
CW      :   IN, DOUBLE (P,NW)
            system output matrix CW of the system W
            (column dense)
            If NW = 0, this matrix is not referenced.
DW      :   IN, DOUBLE (P,P)
            system feedthrough matrix DW of the system W.
            (column dense)
INVW    :   IN, LOGICAL
            specifies whether the conjugate or the inverse of the
            conjugate of W is used to compute the stable projection:
            INVW = .TRUE., the inverse of the conjugate of W is used.
            INVW = .FALSE., the conjugate of W is used.
RWORK   :   OUT, DOUBLE (LWORK)
            working array.
LWORK   :   IN, INTEGER
```

```
          dimension of working array RWORK.
          The value of LWORK must be at least
          max(N,NW)*(max(N,NW)+2)+max(3,P)*N+NW*(NW+2*P+3)+P*P.
 IWORK :  OUT, INTEGER (P)
 *     :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The matrices of the stable projection of W *G are computed using the explicit formulas established in /1/.

For continuous-time systems, the matrices AS, BS, CS and DS of the stable projection are computed as

```
  AS = U'*A*U,  BS = U'*B, CS = (BW'*X+DW'*C)*U, DS = DW'*D,
```

where U is an orthogonal matrix which reduces A to a real Schur form and X satisfies the continuous-time Sylvester equation

```
  AW'*X + X*A + CW'*C = 0.
```

For discrete-time systems, the matrices AS, BS, CS and DS of the stable projection are computed as

```
  AS = U'*A*U,  BS = U'*B, CS = (BW'*X*A+DW'*C)*U,
  DS = (DW'-BW'*inv(AW')*CW')*D,
```

where U is an orthogonal matrix which reduces A to a real Schur form and X satisfies the discrete-time Sylvester equation

```
  AW'*X*A + CW'*C = X.
```

For computing the stable projection of $(W^*)^{-1}*G$, the same formulas are used by replacing the system (AW,BW,CW,DW) by its inverse

$$(AW-BW*DW^{-1}*CW,\ -DW^{-1}*BW,\ CW*DW^{-1},\ DW^{-1}).$$

Literature

/1/ Varga A.
Explicit formulas for an efficient implementation of the frequency-weighting model reduction approach,
Proc. 1993 ECC, Groningen, NL, pp. 693-696, 1993.

Remarks:

- No explicit checks are performed for the singularity of the matrices to be inverted. The error messages 4, 8 or 9 are issued only if exact singularity is detected. If the

respective matrices are nearly singular, the computed results may be inaccurate.

Copyright:
1992 - DLR Institut fuer Dynamik der Flugsysteme

Life cycle:
1992 JULY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
The following sequence of statements can be used to compute a frequency-weighted Balance & Truncate approximation of a stable continuous-time system G = (A,B,C,D) by using W = (AW,BW,CW,DW) as outputs frequency-weighting factor:

```
      DISCR = .FALSE.
      WITHD = .FALSE.
      NA = MAX(N,NW)
      LWORK = NA*(NA+2) + MAX(3,P)*N + NW*(NW+P+P+3) + P*P
      LWORK = MAX(LWORK,4*(N+1)*N)
C                                              *-1
C     Compute G1, the stable projection of W   *G.
      INVW  = .TRUE.
      CALL RPMRFL(A, N, B, M, C, P, D, WITHD, DISCR, AW, NW,
     *           BW, CW, DW, INVW, RWORK, LWORK, IWORK, *1111)
C     Compute G1r, a Balance & Truncate approximation of G1.
      TOL    = 0.1D0
      FIXORD = 0
      CALL RPMRBT(A, N, B, M, C, P, DISCR, TOL, FIXORD, NR, HSV, RWORK,
     *            LWORK, IWORK, *1111)
C                                              *
C     Compute Gr, the stable projection of W *G1r.
      INVW  = .FALSE.
      IF (NR.GT.0) CALL RPMRFL(A, NR, B, M, C, P, D, WITHD, DISCR,
     *             AW, NW, BW, CW, DW, INVW, RWORK, LWORK, IWORK, *1111)
```

By using the above sequence of statements for the system matrices

$$A = \begin{pmatrix} -3.8637 & -7.4641 & -9.1416 & -7.4641 & -3.8637 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and the output frequency-weighting matrices

$$AW = \begin{pmatrix} -2 & -1 \\ 1 & 0 \end{pmatrix} \quad BW = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad CW = \begin{pmatrix} -1.8 & 0 \end{pmatrix} \quad DW = 1.$$

the following results have been obtained:

1) The matrices of resulting G1 = (A1,B1,C1) are:

$$A1 = \begin{pmatrix} -.2588 & 1.1498 & -.4160 & .4591 & 1.7533 & -7.7212 \\ -.8114 & -.2588 & -.0281 & .3703 & -.5606 & 1.1293 \\ 0 & 0 & -.7071 & -.5287 & -.8349 & 2.0301 \\ 0 & 0 & .9457 & -.7071 & -2.6789 & 9.9802 \\ 0 & 0 & 0 & 0 & -.9659 & 6.9494 \\ 0 & 0 & 0 & 0 & -.0096 & -.9659 \end{pmatrix}$$

$$B1 = \begin{pmatrix} -.5286 \\ .0829 \\ .1458 \\ .6908 \\ .4464 \\ -.1263 \end{pmatrix}$$

$$C1 = \begin{pmatrix} 1.8545 & .2907 & -.9917 & 1.5564 & 2.0202 & 8.0920 \end{pmatrix}$$

The Hankel singular values of this system are:

2.6790 2.1589 .8424 .1929 .0219 .0011

2) The matrices of resulting G1r = (A1r, B1r, C1r), the fourth order approximation of G1 obtained by using the Balance & Truncate method are:

$$A1r = \begin{pmatrix} .3561 & 1.1201 & -1.5444 & 2.9212 \\ -.7526 & .3622 & .8039 & -.4983 \\ -.1178 & -.2523 & 1.6688 & -3.8735 \\ -.3645 & -.1765 & 1.9355 & -3.7298 \end{pmatrix} \quad B1r = \begin{pmatrix} -.2287 \\ .0472 \\ .2308 \\ .3288 \end{pmatrix}$$

$$C1r = \begin{pmatrix} -3.1845 & 3.1611 & 4.1436 & -6.1542 \end{pmatrix}$$

3) The matrices of the reduced model Gr = (Ar,Br,Cr), the stable projection of $W^* G1r$, are:

$$Ar = \begin{pmatrix} -.2272 & -1.2419 & 1.8147 & 4.9573 \\ .7378 & -.2272 & .7266 & 1.9178 \\ 0 & 0 & -.4442 & -3.8626 \\ 0 & 0 & .0501 & -.4442 \end{pmatrix} \quad Br = \begin{pmatrix} -.3529 \\ -.1511 \\ .2441 \\ .0946 \end{pmatrix}$$

$$Cr = \begin{pmatrix} -.0057 & -.7088 & -.6474 & .0717 \end{pmatrix}$$

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
Reduction of A or AW to RSF form failed.
-3-
The system (A,B,C,D) is not stable.
-4-
The system (AW,BW,CW,DW) is not invertible.
-5-
The system (AW,BW,CW,DW) is not minimum-phase.
-6-
The system (AW,BW,CW,DW) is not stable.
-7-
The solution of the Sylvester equation failed.
-8-
The system (AW,BW,CW,DW) has a zero in origine.
-9-
The system (AW,BW,CW,DW) has a pole in origine.
-10-
Not enough working storage. It should be at least //LENG//.
```

*SUBROUTINE RPMRFR*a

Stable Projections for Input Frequency-Weighted Model Reduction

Procedure purpose:

Constructs a state-space representation (AS,BS,CS,DS) of the
stable projection of either $G*(W^{-1})^*$ or $G*W^*$ from the state-space representations (A,B,C,D) and (AW,BW,CW,DW) of the transfer-function matrices G and W, respectively. G should be stable. For the postmultiplication with the conjugate of W, the system W used as frequency weight, should be stable and in the discrete-time case, W should have no poles in the origine. For the postmultiplication with the inverse of the conjugate of W, the system W should be invertible and minimum-phase, and for the discrete-time case, W should have no zeros in the origine.

*

W^* denotes the conjugate of W given by either W'(-s) for a continuous-time system or W'(1/z) for a discrete-time system.

Usage:

```
CALL RPMRFR(A, N, B, M, C, P, D, WITHD, DISCR, AW, NW,
   *        BW, CW, DW, INVW, RWORK, LWORK, IWORK, *)
```

```
A     :  IN, OUT, DOUBLE (N,N)
         On input  : system state matrix A of the system G
                     (column dense)
         On output : system state-matrix AS of the stable projection
                            * -1                          *
                     of G*(W )   if INVW = .TRUE. or of G*W  if
                     INVW = .FALSE.. AS results in a real Schur form.
                     (column dense)
N     :  IN, INTEGER, N > 0
         dimension of state vector of the system G
B     :  IN, OUT, DOUBLE (N,M)
         On input  : system input matrix B of the system G
                     (column dense)
         On output : system input matrix BS of the stable projection
                            * -1                          *
                     of G*(W )   if INVW = .TRUE. or of G*W  if
                     INVW = .FALSE..
                     (column dense)
M     :  IN, INTEGER, M > 0
         dimension of input vectors of systems G and W and of the
         output vector of the system W.
C     :  IN, OUT, DOUBLE (P,N)
         On input  : system output matrix C of the system G
                     (column dense)
         On output : system output matrix CS of the stable projection
```

```
                          * -1                              *
                    of G*(W )   if INVW = .TRUE. or of G*W  if
                    INVW = .FALSE..
                    (column dense)
P      :  IN, INTEGER, P > 0
          dimension of output vector of the systems G.
D      :  IN, OUT, DOUBLE (P,M)
          On input  : if WITHD = .TRUE., the system feedthrough matrix
                      D of the system G
                      (column dense)
          On output : if WITHD = .TRUE., the system output matrix DS
                      of the stable projection
                          * -1                              *
                      of G*(W )   if INVW = .TRUE. or of G*W  if
                      INVW = .FALSE..
                      (column dense)
          If WITHD = .FALSE., this matrix is not referenced.
WITHD  :  IN, LOGICAL
          specifies whether or not the feedthrough matrix D is null:
          WITHD = .TRUE.,  a non-zero D is assumed.
          WITHD = .FALSE., a zero D matrix is assumed.
DISCR  :  IN, LOGICAL
          specifies the type of the systems G and W:
          continuous-time systems, if DISCR = .FALSE., or
          discrete-time systems, if DISCR = .TRUE.
AW     :  IN, DOUBLE (NW,NW)
          system state-matrix AW of the system W
          (column dense)
          If NW = 0, this matrix is not referenced.
NW     :  IN, INTEGER, NW >= 0
          dimension of state vector of the system W
BW     :  IN, DOUBLE (NW,M)
          system input matrix BW of the system W
          (column dense)
          If NW = 0, this matrix is not referenced.
CW     :  IN, DOUBLE (M,NW)
          system output matrix CW of the system W
          (column dense)
          If NW = 0, this matrix is not referenced.
DW     :  IN, DOUBLE (M,M)
          system feedthrough matrix DW of the system W.
          (column dense)
INVW   :  IN, LOGICAL
          specifies whether the conjugate or the inverse of the
          conjugate of W is used to compute the stable projection:
          INVW = .TRUE., the inverse of the conjugate of W is used.
          INVW = .FALSE., the conjugate of W is used.
RWORK  :  OUT, DOUBLE (LWORK)
          working array.
LWORK  :  IN, INTEGER
          dimension of working array RWORK.
```

```
          The value of LWORK must be at least
          max(N,NW)*(max(N,NW)+2)+max(3,M)*N+NW*(NW+2*M+3)+M*M.
 IWORK :  OUT, INTEGER (M)
 *     :  RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The matrices of the stable projection of G*W* are computed using the explicit formulas established in /1/.

For continuous-time systems, the matrices AS, BS, CS and DS of the stable projection are computed as

$$AS = U'*A*U,\quad BS = U'*(B*DW'+Y*CW'),\quad CS = C*U,\quad DS = D*DW',$$

where U is an orthogonal matrix which reduces A to a real Schur form and X satisfies the continuous-time Sylvester equation

$$A*Y + Y*AW' + B*BW' = 0.$$

For discrete-time systems, the matrices AS, BS, CS and DS of the stable projection are computed as

$$AS = U'*A*U,\quad BS = U'*(B*DW'+A*Y*CW'),\quad CS = C*U,$$
$$DS = D*(DW'-BW'*inv(AW')*CW'),$$

where U is an orthogonal matrix which reduces A to a real Schur form and X satisfies the discrete-time Sylvester equation

$$A*X*AW' + B*BW' = X.$$

For computing the stable projection of $G*(W^{-1})$, the same formulas are used by replacing the system (AW,BW,CW,DW) by its inverse

$$(AW-BW*DW^{-1}*CW,\ -DW^{-1}*BW,\ CW*DW^{-1},\ DW^{-1}).$$

Literature

/1/ Varga A.
Explicit formulas for an efficient implementation of the frequency-weighting model reduction approach,
Proc. 1993 ECC, Groningen, NL, pp. 693-696, 1993.

Remarks:

- No explicit checks are performed for the singularity of the matrices to be inverted. The error messages 4, 8 or 9 are issued only if exact singularity is detected. If the respective matrices are nearly singular, the computed results

may be inaccurate.

Copyright:
1992 - DLR Institut fuer Dynamik der Flugsysteme

Life cycle:
1992 JULY A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
The following sequence of statements can be used to compute a frequency-weighted Balance & Truncate approximation of a stable continuous-time system G = (A,B,C,D) by using W = (AW,BW,CW,DW) as inputs frequency-weighting factor:

```
      DISCR = .FALSE.
      WITHD = .FALSE.
      NA = MAX(N,NW)
      LWORK = NA*(NA+2) + MAX(3,M)*N + NW*(NW+M+M+3) + M*M
      LWORK = MAX(LWORK,4*(N+1)*N)
      INVW  = .TRUE.
C                                                  *-1
C     Compute G1, the stable projection of G*W    .
      CALL RPMRFR(A, N, B, M, C, P, D, WITHD, DISCR, AW, NW,
     *           BW, CW, DW, INVW, RWORK, LWORK, IWORK, *1111)
C     Compute G1r, a Balance & Truncate approximation of G1.
      TOL    = 0.1D0
      FIXORD = 0
      CALL RPMRBT(A, N, B, M, C, P, DISCR, TOL, FIXORD, NR, HSV, RWORK,
     *            LWORK, IWORK, *1111)
C                                                   *
C     Compute Gr, the stable projection of G1r*W .
      INVW  = .FALSE.
      IF (NR.GT.0) CALL RPMRFR(A, NR, B, M, C, P, D, WITHD, DISCR,
     *             AW, NW, BW, CW, DW, INVW, RWORK, LWORK, IWORK, *1111)
```

By using the above sequence of statements for the system matrices employed as example by the routine RPMRFL and the input frequency weighting matrices

```
AW = ( -2  -1 )     BW = ( 1 )      CW = ( -1.98  0 )     DW = 1.
     (  1   0 )          ( 0 )
```

the following results have been obtained:

1) The matrices of resulting G1 = (A1,B1,C1) are:

$$A1 = \begin{pmatrix} -.2588 & 1.1498 & -.4160 & .4591 & 1.7533 & -7.7212 \\ -.8114 & -.2588 & -.0281 & .3703 & -.5606 & 1.1293 \\ 0 & 0 & -.7071 & -.5287 & -.8349 & 2.0301 \\ 0 & 0 & .9457 & -.7071 & -2.6789 & 9.9802 \\ 0 & 0 & 0 & 0 & -.9659 & 6.9494 \\ 0 & 0 & 0 & 0 & -.0096 & -.9659 \end{pmatrix}$$

$$B1 = \begin{pmatrix} .9829 \\ -1.2926 \\ -.1802 \\ -.1386 \\ .8993 \\ -.2544 \end{pmatrix}$$

$$C1 = \begin{pmatrix} .5286 & .0829 & -.1458 & .6908 & -.4464 & -.1263 \end{pmatrix}$$

The Hankel singular values of this system are:

3.6669 2.7631 .9436 .2203 .0242 .0012

2) The matrices of resulting G1r = (A1r, B1r, C1r), the fourth order approximation of G1 obtained by using the Balance & Truncate method are:

$$A1r = \begin{pmatrix} .0319 & .9780 & -.3791 & 1.4260 \\ -1.0820 & -.1537 & .5603 & .1316 \\ .0471 & .0161 & .0203 & -.2535 \\ -.1054 & -.1199 & 1.7048 & -1.2521 \end{pmatrix} \quad B1r = \begin{pmatrix} .8019 \\ 2.3139 \\ -.2167 \\ .8140 \end{pmatrix}$$

$$C1r = \begin{pmatrix} -.5033 & .3702 & -.1618 & -.6910 \end{pmatrix}$$

3) The matrices of the reduced model Gr = (Ar,Br,Cr), the stable projection of $G1r*W^*$, are:

$$Ar = \begin{pmatrix} -.2320 & -.8461 & 1.3981 & 1.1418 \\ 1.0879 & -.2320 & -.3532 & .1965 \\ 0 & 0 & -.4448 & -1.8414 \\ 0 & 0 & .1086 & -.4448 \end{pmatrix} \quad Br = \begin{pmatrix} -.4294 \\ -.1232 \\ .3549 \\ .2182 \end{pmatrix}$$

$$Cr = \begin{pmatrix} -.4309 & -.4169 & -.7204 & -.1249 \end{pmatrix}$$

Error Messages:

```
-1-
Invalid parameter value on entry.
-2-
Reduction of A or AW to RSF form failed.
-3-
The system (A,B,C,D) is not stable.
-4-
The system (AW,BW,CW,DW) is not invertible.
-5-
The system (AW,BW,CW,DW) is not minimum-phase.
-6-
The system (AW,BW,CW,DW) is not stable.
-7-
The solution of the Sylvester equation failed.
-8-
The system (AW,BW,CW,DW) has a zero in origine.
-9-
The system (AW,BW,CW,DW) has a pole in origine.
-10-
Not enough working storage. It should be at least //LENG//.
```

4. Evaluation of the Norms of Transfer-Function Matrices

The functions of this chapter can be used to compute norms of transfer-function matrices. The main intended applications of these functions are for the evaluation of the norms of the model reduction approximation errors. All functions are applicable to both stable and unstable, minimal or non-minimal, continuous-time or discrete-time systems.

The following subroutines are provided to compute norms:

RPMRHN computes the Hankel-norm and Hankel-singular values of the stable projection of a transfer-function matrix by using the square-root method of /1/ combined with the stable/ antistable projection procedure described in /2/.

RPMRN2 computes the L2 or l2 norm of a transfer-function matrix by using the algorithms described in /3/ and /4/.

Literature:

/1/ Tombs M.S. and Postlethwaite I.
Truncated Balanced Realization of Stable, Non-minimal State-Space Systems.
Int. J. Control, Vol. 46, pp. 1319-1330, 1987.

/2/ Safonov, M.G., Jonckheere, E.A., Verma, M. and Limebeer, D.J.
Synthesis of Positive Real Multivariable Feedback Systems,
Int. J. Control, Vol. 45, pp. 817-842, 1987.

/3/ Varga A.
On computing 2-norms of transfer-function matrices.
Proc. of 1992 American Control Conference, Chicago, June 1992.

/4/ Varga A.
A Schur method for computing coprime factorizations with inner denominators and applications in model reduction,
Research Report 4.92, Dept. Mech. Eng., Ruhr-Univ. Bochum,
Proc. 1993 ACC, San Francisco, CA, pp. 2130-2131, 1993.

SUBROUTINE RPMRHN

Hankel-Norm and Hankel-Singular Values of a Transfer-Function Matrix

Procedure purpose:

Given the matrices A, B and C of a state-space model, this subroutine computes the Hankel-norm and the Hankel-singular values of the stable projection of the corresponding transfer-function matrix.

Usage:

```
CALL RPMRHN(A, N, B, M, C, P, DISCR, HANORM, HSV, NS, RWORK, LWORK, *)
```

```
A     : IN, OUT, DOUBLE (N,N)
        On input  : system state matrix A of the given model.
                    (column dense)
        On output : the state matrix A in a block diagonal real Schur
                    form with reordered eigenvalues. The leading
                    NS x NS part of A has eigenvalues in the
                    stability domain and the trailing (N-NS) x (N-NS)
                    part has eigenvalues outside the stability domain.
                    The stability domain is defined as either the
                    open left half complex plane for a continuous-
                    time system (DISCR = .FALSE.) or the interior of
                    the unit circle for a discrete-time system
                    (DISCR = .TRUE.).
                    (column dense)
N     : IN, INTEGER
        dimension of state vector
B     : IN, OUT, DOUBLE (N,M)
        On input  : system input matrix B of the given model
                    (column dense)
        On output : the input matrix B corresponding to the reduced
                    form of A
                    (column dense)
M     : IN, INTEGER
        dimension of input vector
C     : IN, OUT, DOUBLE (P,N)
        On input  : system output matrix C of the given model
                    (column dense)
        On output : the output matrix C corresponding to the reduced
                    form of A
                    (column dense)
P     : IN, INTEGER
        dimension of output vector
DISCR : IN, LOGICAL
        specifies the type of the system:
        continuous-time system, if DISCR = .FALSE., or
        discrete-time system, if DISCR = .TRUE.
```

```
HANORM:  OUT, DOUBLE
         the Hankel-norm of the stable projection
HSV    : OUT, DOUBLE (N)
         the leading NS elements contain the Hankel-singular values of
         the stable projection ordered decreasingly
NS     : OUT, INTEGER
         order of the stable projection
RWORK  : OUT, DOUBLE (LWORK)
         working array.
LWORK  : IN, INTEGER
         dimension of working array RWORK.
         The value of LWORK must be at least N*(N+5+MAX(P,M)).
*      : RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

If the transfer-function matrix G is unstable, then the stable projection G1 of G is first computed from the additive stable/unstable decomposition of G

$$G = G1 + G2$$

where G1 is the stable projection and G2 is defined as G2 = G - G1. If G is stable, then G1 = G and G2 = 0. The computation of the stable projection is based on the algorithm presented in /1/.

Let (A1,B1,C1) be the state-space representation of G1. Then, the Hankel-norm of G is computed as the maximum Hankel singular value of the system (A1,B1,C1). The computation of the Hankel singular values is performed by using the square-root method of /2/.

Literature

/1/ Safonov, M.G., Jonckheere, E.A., Verma, M. and Limebeer, D.J.
Synthesis of positive real multivariable feedback systems, Int. J. Control, Vol. 45, pp. 817-842, 1987.

/2/ Tombs M.S. and Postlethwaite I.
Truncated balanced realization of stable, non-minimal state-space systems, Int. J. Control, Vol. 46, pp. 1319-1330, 1987.

Remarks:

- The Hankel-norm is only a seminorm when the transfer-function matrix is not strictly proper or not causal.

Copyright:

Life cycle:
1992 AUGUST A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
Given the continuous-time unstable system (A,B,C) used by the subroutine RPMRSD, the Hankel-norm of its stable projection and the corresponding Hankel-singular values can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .FALSE.
      LWORK  = N*(N+5+MAX(P,M))
      CALL RPMRHN(A, N, B, M, C, P, DISCR, HANORM, HSV, NS,
     *            RWORK, LWORK, *1111)
```

The stable projection has order NS = 5.

The computed Hankel-norm of the stable projection is

```
   HANORM(G) = 1.8198
```

The computed Hankel-singular values of the stable projection are:

```
  1.8198    .8047    .7382    .0242    .0238
```

Error Messages:

-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
Reordering of the RSF of A failed.
-4-
Computation of gramians failed.
-5-
Computation of Hankel singular values failed.

SUBROUTINE RPMRN2

L_2- or l_2-Norm of a Transfer-Function Matrix.

Procedure purpose:
Given the matrices A, B, C and D of a state-space model, this subroutine computes the L2- or the l2-norm of the corresponding transfer-function matrix.

Usage:

```
CALL RPMRN2(A, N, B, M, C, P, D, WITHD, DISCR, TOL, L2NORM,
            RWORK, LWORK, *)
```

```
A      :  IN, DOUBLE (N,N)
          system state matrix A of the given model. A must not have
          eigenvalues on the imaginary axis if DISCR = .FALSE. or
          on the unit circle if DISCR = .TRUE. .
          On exit, the content of A is destroyed.
          (column dense)
N      :  IN, INTEGER
          dimension of state vector
B      :  IN, DOUBLE (N,M)
          system input matrix B of the given model.
          On exit, the content of B is destroyed.
          (column dense)
M      :  IN, INTEGER
          dimension of input vector
C      :  IN, OUT, DOUBLE (P,N)
          system output matrix C of the given model.
          On exit, the content of C is destroyed.
          (column dense)
P      :  IN, INTEGER
          dimension of output vector
D      :  IN, DOUBLE (P,M)
          If WITHD = .TRUE., the system feedthrough matrix D of
          the given model.
          If WITHD = .FALSE, D is assumed a P x M null matrix.
          (column dense)
          If WITHD = .FALSE. or DISCR = .FALSE., D is not referenced.
WITHD  :  IN, LOGICAL
          specifies whether or not a given feedthrough matrix D is used
          in computing the l2-norm:
          WITHD = .TRUE.  means a given feedthrough matrix D is used in
          computing the l2-norm.
          (This choice is permitted only if DISCR = .TRUE.).
          WITHD = .FALSE. means D is assumed a null matrix.
DISCR  :  IN, LOGICAL
          specifies the type of the system:
          continuous-time system, if DISCR = .FALSE., or
```

```
          discrete-time system, if DISCR = .TRUE.
 TOL    : IN, DOUBLE
          absolute tolerance level below which the elements of
          B are considered zero (used for controllability tests).
          Recommended value is TOL = N * EPS * NORM(B), where
          NORM(B) denotes the 1-norm of B and EPS is the machine
          precision (see LAPACK Library Routine DLAMCH).
 L2NORM:  OUT, DOUBLE
          the L2-norm if DISCR = .FALSE. or the l2-norm if
          DISCR = .TRUE. of the systems transfer-function matrix.
 RWORK :  OUT, DOUBLE (LWORK)
          working array.
 LWORK  : IN, INTEGER
          dimension of working array RWORK.
          The value of LWORK must be at least
          N*(N+M) + M*(2*M+2) + max(6*N, 4*M, 4*P)
 *      : RETURN 1, target label in case of error (e.g. *1111)
```

File input/ output:

none

Method:

The subroutine is based on the algorithms proposed in /1/ and /2/.

If the given transfer-function matrix G is unstable, then a right coprime factorization with inner denominator of G is first computed

$$G = Q*R^{-1}$$

where Q and R are stable transfer-function matrices and R is inner. If G is stable, then Q = G and R = I.
Let (AQ,BQ,CQ,DQ) be the resulting state-space representation of Q.

If DISCR = .FALSE., then the L2-norm of G is computed as

$$NORM2(G) = NORM2(Q) = SQRT(TRACE(BQ'*X*BQ))$$

where X satisfies the continuous-time Lyapunov equation

$$AQ'*X + X*AQ + CQ'*CQ = 0.$$

If DISCR = .TRUE., then the l2-norm of G is computed as

$$NORM2(G) = NORM2(Q) = SQRT(TRACE(BQ'*X*BQ+DQ'*DQ))$$

where X satisfies the discrete-time Lyapunov equation

$$AQ'*X*AQ - X + CQ'*CQ = 0.$$

Literature

/1/ Varga A.,
On computing 2-norms of transfer-function matrices,
Proc. 1992 American Control Conference, Chicago, June 1992.
/2/ Varga A.
A Schur method for computing coprime factorizations
with inner denominators and applications in model reduction,
Proc. 1993 ACC, San Francisco, CA, pp. 2130-2131, 1993.

Remarks:
none

Copyright:

Life cycle:
1992 AUGUST A. Varga, Ruhr-Universitaet Bochum: coded

Libraries required:
RASP, BLAS (1,2,3), LAPACK

Example:
Given the continuous-time unstable system (A,B,C) used by the subroutine RPMRSD, the L2-norm of its transfer-function matrix can be computed with the following sequence of statements:

```
      N      = 7
      M      = 2
      P      = 3
      DISCR  = .FALSE.
      WITHD = .FALSE.
      TOL = 1.D-14
      LWORK  = N*(N+M+6) + M*(2*M+2)
      CALL RPMRN2(A, N, B, M, C, P, D, WITHD, DISCR, TOL, L2NORM,
     *            RWORK, LWORK, *1111)
```

The computed L2-norm is L2NORM(G) = 7.9395

Error Messages:

-1-
Invalid parameter value on entry.
-2-
Reduction of A to RSF form failed.
-3-
Reordering of the RSF of A failed.
-4-
A has an eigenvalue on the unit circle.
-5-
A has an eigenvalue on the imaginary axis.
-6-
The solution of the Lyapunov equation failed.

Appendix 1. RASP Driver Routines

Model Reduction Tools for Stable Systems

RPMRIB computes reduced order models using the square-root Balance & Truncate model reduction method.
RASP interface to the subroutine SRBT.

RPMRBT computes reduced order models using the square-root balancing-free version of the Balance & Truncate model reduction method.
RASP interface to the subroutine SRBFT.

RPMRSP computes reduced order models using the square-root balancing-free singular perturbation model reduction method.
RASP interface to the subroutine SRBFSP.

RPMROH computes reduced order models using the optimal Hankel-norm approximation method based on square-root balancing.
RASP interface to the subroutine OHNAP.

RPMRSB computes reduced order models using the square-root version of the Balanced Stochastic Truncation model reduction method.
RASP interface to the subroutine SRST.

RPMRST computes reduced order models using the square-root balancing-free version of the Balanced Stochastic Truncation model reduction method.
RASP interface to the subroutine SRBFS.

RPMRSR computes reduced order models by using the singular perturbation approximation formulas.
RASP interface to the subroutine SRESID.

RPMRDC constructs the state-space representation of a system corresponding to a bilinear transformation of its transfer-function matrix.
RASP interface to the subroutine BILIN.

Model Reduction Tools for Unstable Systems

RPMRSD computes an additive spectral decomposition of the transfer-function matrix of a system.
RASP interface to the subroutine SADSDC.

RPMRAS adds or substracts two systems in state-space form.
RASP interface to the subroutine S1PMS2.

RPMRLF constructs a left coprime factorization of a transfer-function matrix with rational factors having a prescribed stability degree.
RASP interface to the subroutine LCFS.

RPMRRF constructs a right coprime factorization of a transfer-function matrix with rational factors having a prescribed stability degree.
RASP interface to the subroutine RCFS.

RPMRLI constructs the left coprime factorization with inner denominator of a transfer-function matrix.
RASP interface to the subroutine LCFID.

RPMRRI constructs the right coprime factorization with inner denominator of a transfer-function matrix.
RASP interface to the subroutine RCFID.

RPMRLB computes the state-space representation of a transfer-function matrix from its stable rational left coprime factorization.
RASP interface to the subroutine LCFI.

RPMRRB computes the state-space representation of a transfer-function matrix from its stable rational right coprime factorization.
RASP interface to the subroutine RCFI.

Frequency-Weighting Model Reduction Tools

RPMRFL constructs the stable projections for output frequency-weighted model reduction.
RASP interface to the subroutine SFRLW.

RPMRFR constructs the stable projections for input frequency-weighted model reduction.
RASP interface to the subroutine SFRRW.

Model Reduction Performance Evaluation Tools

RPMRHN computes the Hankel norm and the Hankel singular values of the stable projection of a tranfer-function matrix.
RASP interface to the subroutine SHANRM.

RPMRN2 compute the L2- or l2-norm of a transfer-function matrix.
RASP interface to the subroutine SL2NRM.

Appendix 2. SLICOT Compatible Driver Routines

BILIN constructs the state-space representation of a system corresponding to a bilinear transformation of its transfer-function matrix.

LCFI constructs the state-space representation of a system from the factors of its left coprime factorization.

LCFID constructs the left coprime factorization with inner denominator of a transfer-function matrix.

LCFS constructs a left coprime factorization of a transfer-function matrix with rational factors having a prescribed stability degree.

OHNAP computes reduced order models using the optimal Hankel-norm approximation method based on square-root balancing.

RCFI constructs the state-space representation of a system from the factors of its right coprime factorization.

RCFID constructs the right coprime factorization with inner denominator of a transfer-function matrix.

RCFS constructs a right coprime factorization of a transfer-function matrix with rational factors having a prescribed stability degree.

S1PMS2 adds or substracts two systems in state-space form.

SADSDC computes an additive spectral decomposition of the transfer-function matrix of a system.

SFRLW constructs the stable projections for output frequency-weighted model reduction.

SFRRW constructs the stable projections for input frequency-weighted model reduction.

SHANRM computes the Hankel norm and the Hankel singular values of the stable projection of a tranfer-function matrix.

SL2NRM compute the L2- or l2-norm of a transfer-function matrix.

SRBFS computes reduced order models using the square-root balancing-free Stochastic Balance & Truncate model reduction method.

SRBFSP computes reduced order models using the square-root balancing-free singular perturbation model reduction method.

SRBFT computes reduced order models using the square-root balancing-free version of the Balance & Truncate model reduction method.

SRBT computes reduced order models using the square-root Balance & Truncate model reduction method.

SRESID computes a reduced order model by using singular perturbation approximation formulas.

SRST computes reduced order models using the square-root Stochastic Balance & Truncate model reduction method.

Appendix 3. SLICOT Compatible Computational and Auxiliary Routines

DGEES1 computes the eigenvalues, the real Schur form and
the corresponding Schur vectors of a real non-symmetric matrix.
(Short version of the LAPACK driver routine DGEES).

DTRSVD computes the singular value decomposition of a real upper
triangular matrix.
(Short version of the LAPACK driver routine DGESVD).

DUTUPD performs a rank-one update of the upper-triangular factor of
a Cholesky decomposition.

OHNAP1 computes reduced order models using the optimal Hankel-norm
approximation method based on square-root balancing
(computational kernel called only by OHNAP).

PLYAPS solves for X = U'U either the stable non-negative definite
continuous-time Lyapunov equation

A'X + XA = -B'B

or the convergent non-negative definite discrete-time Lyapunov
equation

A'XA - X = -B'B

where A is an n by n matrix in real Schur form, B is an
m by n matrix and U is an n by n upper-triangular matrix.

PLYAP1 solves for X = U'*U either the stable non-negative definite
continuous-time Lyapunov equation

S'*X + X*S = -R'*R,

or the convergent non-negative definite discrete-time Lyapunov
equation

S'*X*S - X = -R'*R,

where S is a stable or convergent n by n convergent matrix in a
real Schur form, R is an n by n upper triangular matrix and
U is an n by n upper-triangular matrix.

PLYAP2 solves for X = U'*U either the continuous-time Lyapunov equation

S'*X + X*S = -ISGN*R'*R

or the discrete-time Lyapunov equation

S'*X*S - X = -ISGN*R'*R,

where S is a two by two matrix with complex conjugate
eigenvalues, R is a two by two upper triangular matrix,
ISGN = -1 or 1, and U is a two by two upper triangular matrix.

PLYAP4 solves for X either the continuous-time Sylvester equation
S'*X + X*A = C,
or the discrete-time Sylvester equation
S'*X*A - X = C,
where S is an n by n matrix in a real Schur form, A is an m by m matrix (m = 1 or m = 2) and C is an n by m matrix. (computational kernel called only by PLYAP1).

RCFS1 constructs the factors of a stable rational right coprime factorization of a transfer-function matrix. The computed factors have a prescribed stability degree. (computational kernel called only by LCFS and RCFS)

RCFID1 constructs the factors of a right coprime factorization with inner denominator of a transfer-function matrix. (computational kernel called only by LCFID and RCFID)

RCFID2 constructs the inner denominator of a right coprime factorization for a system of order at most two.

SALOC2 solves an n by n pole placement problem (n = 1 or 2).

SEIG computes the eigenvalues of an upper quasi-triangular matrix.

SEOR1 reorders the diagonal blocks of a principal submatrix of an upper quasi-triangular matrix together with its eigenvalues by constructing an orthogonal similarity transformation.

SPLITB computes the eigenvalues of a 2 by 2 diagonal block of an upper quasi-triangular matrix, reduces it to the standard form and splits the block in the case of real eigenvalues by constructing an orthogonal transformation.

SFRLW1 constructs the stable projections for input frequency-weighted model reduction. (computational kernel called only by SFRLW)

SFRRW1 constructs the stable projections for output frequency-weighted model reduction. (computational kernel called only by SFRRW)

SRBFP1 computes reduced order models using the square-root balancing-free singular perturbation model reduction method. (computational kernel called only by SRBFSP)

SRBFS1 computes reduced order models using the square-root balancing-free Stochastic Balance & Truncate model reduction method. (computational kernel called only by SRBFS)

SRBFT1 computes reduced order models using the square-root balancing-free version of the Balance & Truncate model reduction method. (computational kernel called only by SRBFT)

SRBT1 computes reduced order models using the square-root Balance & Truncate model reduction method. (computational kernel called only by SRBT)

SRGRO computes the Cholesky factor R of the observability Gramian Q = R'*R, satisfying a Riccati equation of the form

$$(A - BE^{-1}C)^T Q + Q(A - BE^{-1}C) + C^T E^{-1} C + QBE^{-1}B^T Q \;=\; 0$$

SRGRO1 (computational kernel called only by SRGRO)

SRSFDC reduces the state matrix of a system to an upper real Schur form by using an orthogonal similarity transformation and applies the accumulated transformation to the systems input and output matrices.

SRSFOD reduces the state matrix of a system to an upper real Schur form by using an orthogonal similarity transformation, reorders the diagonal blocks of the computed real Schur form and applies the accumulated transformation to the systems input and output matrices.

SRST1 computes reduced order models using the square-root Stochastic Balance & Truncate model reduction method. (computational kernel called only by SRST)

SYLVS solves for the n by m matrix X, the Sylvester matrix equation in either the continuous-time form

```
A*X + ISGN*X*B = C
```

or in the discrete-time form

```
ISGN*A*X*B + X = C;
```

where A and B are respectively n by n and m by m matrices in real Schur form, C is an n by m matrix and ISGN = 1 or -1.

SYLV2 solves for the n by m matrix X, 1 <= n,m <= 2, either the continuous-time Sylvster equation

```
TL*X + ISGN*X*TR = B,
```

or the discrete-time Sylvester equation

```
ISGN*TL*X*TR + X = B,
```

where TL is n by n, TR is m by m, B is n by m and ISGN = 1 or -1.

SYSINV computes the inverse of a given system.

Auxiliary Routines

DCROTG constructs a complex plane rotation.

DMNEG changes the sign of a two-dimensional matrix.

DMPTR pertransposes a band of a square two-dimensional matrix.

DMSCAL scales a two-dimensional matrix by using row and/or column scaling factors.

DMTRA transposes all or part of a two-dimensional matrix into another matrix.

HOUSH applies the Householder reflection computed by the LAPACK routine DLARFG.

PAP reverses the order of rows and/or columns of a two-dimensional matrix.

Appendix 4. Called LAPACK and BLAS Routines

LAPACK Routines

```
DBDSQR, DGEBAK, DGEBAL, DGEBD2, DGECON, DGEHD2, DGEHRD, DGELSX, DGEQPF,
DGEQR2, DGEQRF, DGETF2, DGETRF, DGETRI, DGETRS, DHSEQR, DORG2R, DORGBR,
DORGHR, DORGL2, DORGLQ, DORGQR, DORM2R, DRSCL,  DTREXC, DTRSYL, DTRTI2,
DTRTRI, DTZRQF

DLABAD, DLACON, DLACPY, DLADIV, DLAEXC, DLAHQR, DLAHRD, DLAIC1, DLALN2,
DLANGE, DLANHS, DLANV2, DLAPY2, DLAPY3, DLARF,  DLARFB, DLARFG, DLARFT,
DLARFX, DLARTG, DLAS2,  DLASCL, DLASET, DLASR,  DLASSQ, DLASV2, DLASWP,
DLASY2, DLATRS, DLATZM, DLAZRO

DLAMC1, DLAMC2, DLAMC3, DLAMC4, DLAMC5, DLAMCH, ILAENV, LSAME
```

BLAS 1 Routines

```
DASUM, DAXPY, DCOPY, DDOT, DNRM2, DROT, DROTG, DSCAL, DSWAP, IDAMAX
```

BLAS 2 Routines

```
DGEMV, DGER, DTRMV, DTRSV, XERBLA
```

BLAS 3 Routines

```
DGEMM, DSYRK, DTRMM, DTRSM
```