SLICOT Working Note 1998-2

Task II.A.1 - Selection of Model Reduction Routines [1]

Andras Varga[2]

June 1998

[1]This document presents research results of the European Community BRITE-EURAM III Thematic Networks Programme NICONET (contract number BRRT-CT97-5040) and is distributed by the Working Group on Software WGS. *WGS secretariat*: Mrs. Ida Tassens, ESAT - Katholieke Universiteit Leuven, K. Mercierlaan 94, 3001-Leuven-Heverlee, BELGIUM. This report is also available by anonymous ftp from `wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN1998-2.ps.Z`

[2]Deutsches Zentrum für Luft und Raumfahrt, Institut für Robotik und Systemdynamik, DLR Oberpfaffenhofen, Postfach 1116, D-82230 Wessling, Germany

Contents

1 Introduction

Model reduction is of fundamental importance in many modeling and control applications. The basic reduction algorithms discussed in this working note belong mostly to the class of methods based on or related to balancing techniques [15, 14, 7, 9] and are primarily intended for the reduction of linear, stable, continuous- or discrete-time systems. All methods rely on guaranteed error bounds and have particular features which recommend them for use in specific applications. The basic methods combined with coprime factorization or spectral decomposition techniques can be used to reduce unstable systems [13] or to perform frequency-weighted model reduction [12, 8].

The basis for standardization of the model reduction routines in SLICOT will form the collection of routines available in the RASP-MODRED library [23]. The underlying algorithms represent the latest developments of various procedures for solving computational problems appearing in the context of model reduction. Most algorithms possess desirable attributes as generality, numerical reliability, enhanced accuracy, and thus are completely satisfactory to serve as bases for robust software implementations. The implementations of routines in RASP-MODRED are based on the linear algebra standard package LAPACK [3] and are suitable for standardization in SLICOT. It is worth mentioning that the implemented algorithms in RASP-MODRED are generally superior to those implemented in the model reduction tools of commercial packages [6, 4, 2].

Note however that for some model reduction methods of potential interest in practical applications, as for instance the enhanced modal reduction approach [27] , the optimal L_2–norm model reduction [30] or the frequency-weighted reduction using Enns's approach [8], no reliable software in FORTRAN is available in this moment.

2 Overview of Balancing Related Model Reduction Algorithms

Consider the n-th order original state-space model $G := (A, B, C, D)$ with the *transfer-function matrix* (TFM) $G(\lambda) = C(\lambda I - A)^{-1}B + D$, and let $G_r := (A_r, B_r, C_r, D_r)$ be an r-th order approximation of the original model ($r < n$), with the TFM $G_r = C_r(\lambda I - A_r)^{-1}B_r + D_r$. A large class of model reduction methods can be interpreted as performing a similarity transformation Z yielding

$$\left[\begin{array}{c|c} Z^{-1}AZ & Z^{-1}B \\ \hline CZ & D \end{array}\right] := \left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & D \end{array}\right],$$

and then defining the reduced model (A_r, B_r, C_r, D_r) as the leading diagonal system $(A_{11},\ B_1,\ C_1,\ D)$. When writing $Z := [\,T\ U\,]$ and $Z^{-1} := [\,L^T\ V^T\,]^T$, then $\Pi = TL$ is a projector on T along L and $LT = I_r$. Thus the reduced system is $(A_r, B_r, C_r, D_r) = (LAT,\ LB,\ CT,\ D)$. Partitioned forms as above can be used to construct a so-called *singular perturbation approximation* (SPA).

The matrices of the reduced model in this case are given by

$$\begin{array}{rcl} A_r & = & A_{11} + A_{12}(\gamma I - A_{22})^{-1}A_{21}, \\ B_r & = & B_1 + A_{12}(\gamma I - A_{22})^{-1}B_2, \\ C_r & = & C_1 + C_2(\gamma I - A_{22})^{-1}A_{21}, \\ D_r & = & D + C_2(\gamma I - A_{22})^{-1}B_2. \end{array} \tag{1}$$

where $\gamma = 0$ for a continuous-time system and $\gamma = 1$ for a discrete-time system. Note that SPA formulas preserve the DC-gains of stable original systems.

Specific requirements for model reduction algorithms are formulated and discussed in [28]. Such requirements are: (1) applicability of methods regardless the original system is minimal or not; (2) emphasis on enhancing the numerical accuracy of computations; (3) relying on numerically reliable procedures.

The first requirement can be fulfilled by computing L and T directly, without determining Z or Z^{-1}. In particular, if the original system is not minimal, then L and T can be chosen to compute an *exact* minimal realization of the original system [21].

The emphasis on improving the accuracy of computations led to so-called algorithms with *enhanced accuracy*. In many model reduction methods, the matrices L and T are determined from two positive semi-definite matrices P and Q, called generically *gramians*. The gramians can be always determined in Cholesky factorized forms $P = S^TS$ and $Q = R^TR$, where S and R are upper-triangular matrices. The computation of L and T can be done by computing the *singular value decomposition* (SVD)

$$SR^T = \left[\begin{array}{cc} U_1 & U_2 \end{array} \right] \mathrm{diag}(\Sigma_1, \Sigma_2) \left[\begin{array}{cc} V_1 & V_2 \end{array} \right]^T$$

where

$$\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_r), \quad \Sigma_2 = \mathrm{diag}(\sigma_{r+1}, \ldots, \sigma_n),$$

and $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \geq \sigma_n \geq 0$.

The so-called *square-root* (**SR**) methods determine L and T as [19]

$$L = \Sigma_1^{-1/2}V_1^TR, \qquad T = S^TU_1\Sigma_1^{-1/2}.$$

If r is the order of a minimal realization of G then the gramians corresponding to the resulting realization are diagonal and equal. In this case the minimal realization is called *balanced*. The **SR** approach is usually very accurate for well-equilibrated systems. However if the original system is highly unbalanced, potential accuracy losses can be induced in the reduced model if either L or T is ill-conditioned.

In order to avoid ill-conditioned projections, a *balancing-free* (**BF**) approach has been proposed in [17] in which always well-conditioned matrices L and T can be determined. These matrices are computed from orthogonal matrices whose columns span orthogonal bases for the right and left eigenspaces of the product PQ corresponding to the first r largest eigenvalues $\sigma_1^2, \ldots, \sigma_r^2$. Because of the need to compute explicitly P and Q as well as their product, this approach is usually less accurate for moderately ill-balanced systems than the **SR** approach.

A *balancing-free square-root* (**BFSR**) algorithm which combines the advantages of the **BF** and **SR** approaches has been introduced in [21]. L and T are determined as

$$L = (Y^T X)^{-1} Y^T, \qquad T = X,$$

where X and Y are $n \times r$ matrices with orthogonal columns computed from the QR decompositions $S^T U_1 = XW$ and $R^T V_1 = YZ$, while W and Z are non-singular upper-triangular matrices. The accuracy of the **BFSR** algorithm is usually better than either of **SR** or **BF** approaches.

The SPA formulas can be used directly on a balanced minimal order realization of the original system computed with the **SR** method. A **BFSR** method to compute SPAs has been proposed in [20]. The matrices L and T are computed such that the system (LAT, LB, CT, D) is minimal and the product of corresponding gramians has a block-diagonal structure which allows the application of the SPA formulas.

Provided the Cholesky factors R and S are known, the computation of matrices L and T can be done by using exclusively numerically stable algorithms. Even the computation of the necessary SVD can be done without forming the product SR^T. Thus the effectiveness of the **SR** or **BFSR** techniques depends entirely on the accuracy of the computed Cholesky factors of the gramians. In the following sections we discuss the computation of these factors for several concrete model reduction techniques.

3 Algorithms and Software for Stable Systems

In the *balance & truncate* (B&T) method [15] P and Q are the controllability and observability gramians satisfying a pair of continuous- or discrete-time Lyapunov equations

$$AP + PA^T + BB^T = 0, \quad A^T Q + QA + C^T C = 0;$$

$$APA^T + BB^T = P, \quad A^T QA + C^T C = Q.$$

These equations can be solved directly for the Cholesky factors of the gramians by using numerically reliable algorithms proposed in [10]. The **BFSR** version of the B&T method is described in [21]. Its **SR** version [19] can be used to compute balanced minimal representations. Such representations are also useful for computing reduced order models by using the SPA formulas [14] or the *Hankel-norm approximation* (HNA) method [9]. A **BFSR** version of the SPA method is described in [20]. Note that the B&T, SPA and HNA methods belong to the family of absolute error methods which try to minimize $\|\Delta_a\|_\infty$, where Δ_a is the absolute error $\Delta_a = G - G_r$.

The *balanced stochastic truncation* (BST) method [7] is a relative error method which tries to minimize $\|\Delta_r\|_\infty$, where Δ_r is the relative error defined implicitly by $G_r = (I - \Delta_r)G$. In the BST method the gramian Q satisfies a Riccati equation, while the gramian P still satisfies a Lyapunov equation. Although the determination with high accuracy of the Cholesky factor of Q is computationally involved, it is however necessary to guarantee the effectiveness of the **BFSR** approach. Iterative refinement techniques are described for this purpose in [28].

Both the **SR** and **BFSR** versions of the B&T, SPA and BST algorithms are implemented in the RASP-MODRED library. The implementation of the HNA method uses the **SR** version of the B&T method to compute a balanced minimal realization of the original system.

All implemented routines are applicable to both continuous- and discrete-time systems. It is worth mentioning that implementations provided in commercial software [6, 4, 2] are only for continuous-time systems.

The following routines are available in RASP-MODRED for stable model reduction:

SRBT	computes reduced (or minimal) order balanced models using the **SR** B&T method [19]
SRBFT	computes reduced order models using the **BFSR** version of the B&T method [21]
SRBFSP	computes reduced order models using the **BFSR** SPA method [20]
OHNAP	computes reduced order models using the optimal HNA method based on **SR** balancing [9]
SRST	computes reduced order models using the **SR** BST method [16]
SRBFS	computes reduced order models using the **BFSR** BST method [28]
SRESID	computes a reduced order model by using the singular perturbation formulas (1).

For both subroutines SRST and SRBFS, 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.

4 Model reduction of unstable systems

The reduction of unstable systems can be performed by using the methods for stable systems in conjunction with two embedding techniques. The first approach consists in reducing only the stable projection of G and then including the unstable projection unmodified in the resulting reduced model. The following is a simple procedure for this computation:

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

Note that for the model reduction at step 2 any of methods available for stable systems can be used. The second approach is based on computing a stable *rational coprime factorization* (RCF) of G. 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 $G = M^{-1}N$, where M, N are stable and proper rational TFMs.
2. Approximate the stable system of order n $[\, N \;\; M \,]$ with $[\, N_r \;\; M_r \,]$ of order r.

3. Form the r-th order approximation $G_r = M_r^{-1} N_r$.

The coprime factorization approach used in conjunction with the B&T or BST methods fits in the general projection formulation introduced in Section 2. The gramians necessary to compute the projection are the gramians of the system $[\,N\ M\,]$. The computed matrices L and T by using either the **SR** or **BFSR** methods can be directly applied to the matrices of the original system. The main computational problem is how to compute the RCF to allow a smooth and efficient embedding which prevents computational overheads. Two factorization algorithms proposed recently compute particular RCFs which fulfill these aims: the RCF with prescribed stability degree [24] and the RCF with inner denominator [26]. Both are based on a numerically reliable Schur technique for pole assignment. The use of normalized RCFs has a certain importance in some controller reduction applications and algorithms are available (based on solving appropriate Riccati equations) for both continuous-time [29] and discrete-time [5]. Note that the approximations computed for the factors of a coprime factorization with inner denominator or of a normalized coprime factorization by using the SPA method preserve these properties also at the level of the reduced factors.

RASP-MODRED provides several necessary tools to perform the reduction of unstable system. Routines are provided to compute left/right RCFs with prescribed stability degree or with inner denominators, to compute additive spectral decompositions, or to perform the back transformations. A modular implementation allows arbitrary combinations between various factorization and model reduction methods. Prototype routines in MATLAB are available for computing normalized coprime factorizations too.

The following routines, available to compute various decompositions and factorization, will be part of the Proposal for Task I.A.1:

SADSDC	computes the terms G1 and G2 of an additive spectral decomposition of a transfer-function matrix G with respect to a specified region of the complex plane [18].
LCFS	computes the state-space representations for the factors of a left coprime factorization of a transfer-function matrix with prescribed stability degree [24].
RCFS	computes the state-space representations from the factors of a right coprime factorization of a transfer-function matrix with prescribed stability degree [24].
LCFID	computes the state-space representations for the factors of a left coprime factorization with co-inner denominator of a transfer-function matrix [26].
RCFID	computes the state-space representations for the factors of a right coprime factorization with inner denominator of a transfer-function matrix [26].
LCFI	computes the state-space representation corresponding to a left coprime factorization of a transfer-function matrix.
RCFI	computes the state-space representation corresponding to a right coprime factorization of a transfer-function matrix.

5 Frequency-Weighted Model Reduction

The *frequency-weighted model reduction* (FWMR) methods try to minimize a weighted error of the form $\|W_1(G - G_r)W_2\|_\infty$, where W_1 and W_2 are suitable weighting TFMs. Many controller reduction problems can be formulated as FWMR problems [1]. Two basic approaches can be used to solve such problems. The approach proposed in [8] can be easily embedded in the general formulation of Section 2. Provided G and the weights W_1 and W_2 are all stable TFMs, then P and Q are the frequency-weighted controllability and observability gramians of GW_2 and W_1G, respectively (for details see [1]). Unfortunately no proof of stability of the two-sided weighted approximation exists unless either $W_1 = I$ or $W_2 = I$.

In the second approach we assume that G is stable and W_1, W_2 are invertible, having only unstable poles and zeros. The technique proposed in [12] to solve the FWMR problem computes first G_1 the n-th order stable projection of W_1GW_2 and then computes the r-th order approximation G_{1r} of G_1 by using one of methods for stable systems. Finally G_r results as the r-th order stable projection of $W_1^{-1}G_{1r}W_2^{-1}$.

RASP-MODRED provides all necessary tools to perform FWMR. Special routines based on algorithms proposed in [25] are provided to compute efficiently the stable projections for the second approach:

SFRLW	constructs for either $(W^*)^{-1}G$ or W^*G an n-th order state-space realization of its stable projection by using the explicit formulas derived in [25].
SFRRW	constructs for either $G(W^*)^{-1}$ or GW^* an n-th order state-space realization of its stable projection by using the explicit formulas derived in [25].

6 Topics of Interest but not Covered with Software

6.1 Modal Approach

The modal approach to model reduction can be interpreted as performing a similarity transformation Z on the system matrices yielding

$$\left[\begin{array}{c|c} Z^{-1}AZ & Z^{-1}B \\ \hline CZ & D \end{array}\right] := \left[\begin{array}{cc|c} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ \hline C_1 & C_2 & D \end{array}\right], \tag{2}$$

where A_1 and A_2 contains the r *dominant* and respectively, the $n-r$ *non-dominant* eigenvalues (modes) of A, and then defining the reduced model on the basis of this partitioned representation. The above partition of system matrices is equivalent with the additive decomposition $G = G_1 + G_2$, where $G_1 := (A_1, B_1, C_1, D)$ and $G_2 := (A_2, B_2, C_2, 0)$ are the dominant and non-dominant subsystems, respectively.

One difficulty in using the modal approach is the lack of an *a priori* computable bound for the resulting approximation error $\Delta = G - G_r$. The actual error can be computed only after that a choice of the order has been made. Thus model reduction based on the modal approach is done

typically on a *trial and error* basis. In contrast, methods based on balancing, as for example the B&T method [15, 9], provide *a priori* information (the Hankel-singular values) which can be used to select the appropriate order for an acceptable approximation error.

It is possible to combine the modal approach with other techniques. For example, if the system is already decomposed as in (2), then the reduction can be performed separately on G_1 and G_2. Let $G_r = G_{1r} + G_{2r}$ be the resulting reduced model, where G_{1r} and G_{2r} are the resulting reduced subsystems computed say with the B&T method. If for the separate reduction of terms we have that $\|G_i - G_{ir}\| \leq \varepsilon_i$ for $i = 1, 2$, then $\|G - G_r\| \leq \varepsilon_1 + \varepsilon_2$. Thus, by reducing individually the terms, we can also control the resulting global error by choosing appropriate orders for the reduced subsystems. The technique can be readily extended to additive decompositions with more than two terms (see below) and many variations of it are possible by employing alternative model reduction methods.

The real advantage of such combinations is more evident when we have to reduce very large order models, as those which typically result from finite-element analysis of large mechanical structures. Because of large orders of such models, the modal approach is frequently the only method which can be used for order reduction. This reduction is often only a preliminary step which makes tractable further reductions with the help of more powerful methods.

A major problem of modal reduction is the need for reliable modal dominance measures. An enhanced modal dominance analysis technique has been proposed in [27] and is based on the *block-diagonal form* (BDF) of the state matrix A. This decomposition of A corresponds to an additive decomposition of the TFM as $G = \sum_{i=1}^{k} G_i$ and can be used to define dominance measures for nearby eigenvalues grouped into one block A_i of the BDF of A and defining the subsystem G_i. Using this decomposition it is possible to reduce individually each stable subsystem using one of already mentioned methods. Note the one advantage of the modal approach it that it is applicable to both stable and unstable systems, the unstable part being automatically included in the dominant subsystem. For the computation of the BDF the subroutine BDIAG is available in RASP. The standardization of this routine will be part of the Proposal for Task I.A.1.

6.2 H_2-norm Reduction

Optimal H_2-norm approximation of a stable n-th order system $G = (A, B, C)$ by a stable r-th order system $G_r = (A_r, B_r, C_r)$ can be formulated as a minimization problem with respect to the elements of the matrices A_r, B_r and C_r:

$$\min \|G - G_r\|_2.$$

The necessary conditions of optimality lead to a set of *optimal projection equations* [11]. To solve these equations apparently homotopy (or continuation) methods are best suited. An alternative algorithm expressed in terms of the projection matrices L and T has been proposed in [30] and a MATLAB implementation, which could serve for standardization within SLICOT, is available on the WOR-Toolbox [30].

7 Auxiliary Software Tools for Model Reduction

To evaluate the approximation errors for the resulting reduced order models, different norms of the TFMs are necessary to be computed. The following routines are provided in the RASP-MODRED package for this purpose:

SHANRM	computes the Hankel norm and the Hankel singular values of the stable projection of a transfer-function matrix
SL2NRM	compute the L_2- or l_2-norm of a transfer-function matrix

These routines will be part of the Proposal for Task I.A.1. The RPHINR routine, available in RASP, to compute the L_∞-norm, could also serve for standardization purpose, although this would imply complete rewriting of it and substantial algorithmic improvements using the recently developed routines to compute the eigenvalues of Hamiltonian pencils. Note that in this context a routine to compute the eigenvalues of simplectic pencils would be very helpful.

References

[1] B. D. O. Anderson and Y. Liu. Controller reduction: concepts and approaches. *IEEE Trans. Autom. Control*, AC-34:802–812, 1989.

[2] B.D.O Anderson and B. James. *$MATRIX_X$ Model Reduction Module.* ISI, Santa Clara, CA, 1991.

[3] E. Anderson, Z. Bai, J. Bishop, J. Demmel, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK User's Guide, Second Edition.* SIAM, Philadelphia, 1995.

[4] G. Balas, J. Doyle, K. Glover, A. Packard, and R. Smith. *μ-Analysis and Synthesis Toolbox 2.0.* The MathWorks Inc., Natick, MA, 1993.

[5] P. M. M. Bongers and P. S. C. Heuberger. Discrete normalized coprime factorization. In A. Bensoussan and J. L. Lions, editors, *Proc. 9th INRIA Conf. Analysis and Optimization of Systems*, volume 144 of *Lect. Notes Control and Inf. Scie.*, pages 307–313. Springer-Verlag, Berlin, 1990.

[6] R. Y. Chiang and M. G. Safonov. *Robust Control Toolbox 2.0.* The MathWorks Inc., Natick, MA, 1992.

[7] U. B. Desai and D. Pal. A transformation approach to stochastic model reduction. *IEEE Trans. Autom. Control*, AC-29:1097–1100, 1984.

[8] D. Enns. *Model Reduction for Control Systems Design.* PhD thesis, Dept. Aeronaut. Astronaut., Stanford Univ., Stanford, CA, 1984.

[9] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *Int. J. Control*, 39:1115–1193, 1974.

[10] S. J. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. Numer. Anal.*, 2:303–323, 1982.

[11] D. C. Hyland and D. S. Bernstein. The optimal projection equations for model reduction and the relationship among methods of wilson, skelton and moore. *IEEE Trans. Autom. Control*, 30:1201–1211, 1985.

[12] G. A. Latham and B. D. O. Anderson. Frequency-weighted optimal Hankel norm approximation of stable transfer functions. *Systems & Control Lett.*, 5:229–236, 1985.

[13] Y. Liu and B. D. O. Anderson. Controller reduction via stable factorization and balancing. *Int. J. Control*, 44:507–531, 1986.

[14] Y. Liu and B. D. O. Anderson. Singular perturbation approximation of balanced systems. *Int. J. Control*, 50:1379–1405, 1989.

[15] B. C. Moore. Principal component analysis in linear system: controllability, observability and model reduction. *IEEE Trans. Autom. Control*, AC-26:17–32, 1981.

[16] M. G. Safonov and R. Y. Chiang. Model reduction for robust control: a Schur relative error method. *Int. J. Adapt. Contr.&Sign. Proc.*, 2:259–272, 1988.

[17] M. G. Safonov and R. Y. Chiang. A Schur method for balanced-truncation model reduction. *IEEE Trans. Autom. Control*, AC-34:729–733, 1989.

[18] M. G. Safonov, E. A. Jonckheere, M. Verma, and D. J. N. Limebeer. Synthesis of positive real multivariable feedback systems. *Int. J. Control*, 45:817–842, 1987.

[19] M. S. Tombs and I. Postlethwaite. Truncated balanced realization of a stable non-minimal state-space system. *Int. J. Control*, 46:1319–1330, 1987.

[20] A. Varga. Balancing-free square-root algorithm for computing singular perturbation approximations. In *Proc. of 30th IEEE CDC, Brighton, UK*, pages 1062–1065, 1991.

[21] A. Varga. Efficient minimal realization procedure based on balancing. In A. El Moudni, P. Borne, and S. G. Tzafestas, editors, *Prepr. of IMACS Symp. on Modelling and Control of Technological Systems*, volume 2, pages 42–47, 1991.

[22] A. Varga. On computing 2-norms of transfer function matrices. In *Proc. 1992 American Control Conference, Chicago, Illinois*. Elsevier, Amsterdam, 1992.

[23] A. Varga. *RASP Model Order Reduction Programs.* University of Bochum and DLR-Oberpfaffenhofen, TR R88-92, August 1992.

[24] A. Varga. Coprime factors model reduction based on accuracy enhancing techniques. *Systems Analysis Modelling and Simulation*, 11:303–311, 1993.

[25] A. Varga. Explicit formulas for an efficient implementation of the frequency-weighted model reduction approach. In *Proc. 1993 European Control Conference, Groningen, NL*, pages 693–696, 1993.

[26] A. Varga. A Schur method for computing coprime factorizations with inner denominators and applications in model reduction. In *Proc. 1993 American Control Conference, San Francisco, CA*, pages 2130–2131, 1993.

[27] A. Varga. Enhanced modal approach for model reduction. *Mathematical Modelling of Systems*, 1:91–105, 1995.

[28] A. Varga and K. H. Fasol. A new square-root balancing-free stochastic truncation model reduction algorithm. In *Prepr. of 12th IFAC World Congress, Sydney, Australia*, volume 7, pages 153–156, 1993.

[29] M. Vidyasagar. Normalized coprime factorization for nonstrictly proper systems. *IEEE Trans. Autom. Control*, 33:300–301, 1988.

[30] P. Wortelboer. *Frequency-weighted Balanced Reduction of Closed-loop Mechanical Servo-systems: Theory and Tools.* PhD thesis, Techn. Univ. Delft, 1994.

A List of Routines to be Standardized within TASK II.A.1

A.1 Routines for Reduction of Stable Systems

Name	Function
AB09AD	computes reduced (or minimal) order models using either the **SR** or the **BFSR** B&T method
AB09BD	computes reduced order models using the **BFSR** SPA method
AB09CD	computes reduced order models using the optimal HNA method based on **SR** balancing
AB09DD	computes a reduced order model by using the singular perturbation formulas

A.2 Lower Level Routines for Reduction of Stable Systems

Name	Function
AB09AX	computes reduced (or minimal) order balanced models using either the **SR** or the **BFSR** B&T method (scaled system with state matrix in real Schur form)
AB09BX	computes reduced order models using the **BFSR** SPA method (scaled system with state matrix in real Schur form)
AB09CX	computes reduced order models using the optimal HNA method based on **SR** balancing (scaled system with state matrix in real Schur form)

A.3 Routines for Reduction of Unstable Systems

Name	Function
AB09ED	computes reduced order models for unstable systems using the optimal HNA method in conjunction with additive spectral decomposition
AB09FD	computes reduced order models for unstable systems using the **BFSR** B&T method in conjunction with left/right coprime factorization methods
AB09GD	computes reduced order models for unstable systems using the **BFSR** SPA method in conjunction with left/right coprime factorization methods

B List of Routines Necessary to Model Reduction Standardized within TASK I.A.1

B.1 Mathematical Routines

Name	Function
MB03QD	reorders the eigenvalues of a real Schur matrix according to several reordering criteria
MB03UD	computes the singular value decomposition of a square upper-triangular matrix

B.2 Lower Level/Auxiliary Mathematical Routines

Name	Function
MA02AD	transposes all or a part of a matrix
MA02BD	reverses the order of rows and/or columns of a matrix
MA02CD	pertransposes a diagonal band of matrix
MB01SD	scales a matrix by rows or columns
MB03QX	computes the eigenvalues of a matrix in real Schur form
MB03QY	computes the eigenvalues of a 2 by 2 block of matrix in real Schur form and reduces it to the standard LAPACK form

B.3 Transformation Routines

Name	Function
TB01KD	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
TB01LD	performs an orthogonal similarity trasformation to reduce the system state matrix to an ordered real Schur form
TB01WD	performs an orthogonal similarity trasformation to reduce the system state matrix to a real Schur form

B.4 Analysis Routines

Name	Function
AB13AD	computes the Hankel norm and the Hankel singular values of the stable projection of a transfer-function matrix
AB13BD	computes the L_2- or l_2-norm of a transfer-function matrix

B.5 Factorization Routines

Name	Function
SB08AD	computes the state-space representations of the factors of a LCF with prescribed stability degree
SB08BD	computes the state-space representations of the factors of a RCF with prescribed stability degree
SB08CD	computes the state-space representations of the factors of a LCFID of a TFM
SB08DD	computes the state-space representations of the factors of a RCFID of a TFM
SB08GD	computes the state-space representation of the TFM corresponding to a LCF
SB08HD	computes the state-space representation of the TFM corresponding to a RCF

B.6 Lower Level/Auxiliary Routines Necessary for the Factorization Routines

Name	Function
SB01BY	solves an N by N pole placement problem for N = 1 or N = 2
SB01FY	computes the inner denominator of a right-coprime factorization of a system of order N, where N is either 1 or 2