2002 IEEE International Symposium on Computer
Aided Control System Design Proceedings
September 18-20, 2002 • Glasgow, Scotland, U.K.

Numerical Software in SLICOT for Low Order Controller Design

A. Varga

German Aerospace Center (DLR) - Oberpfaffenhofen
Institute of Robotics and Mechatronics
D-82234 Wessling, Germany
Andras.Varga@dlr.de

Abstract

We describe the model and controller reduction software recently developed for the control and systems library SLICOT. Based on the latest algorithmic developments, a powerful collection of Fortran 77 subroutines has been implemented to cover the relative error and frequency-weighted model reduction, as well as special controller reduction approaches. The new model and controller reduction routines are among the most powerful and numerically most reliable software tools available for model and controller reduction. To facilitate their usage in user friendly environments, easy-to-use and flexible interfaces have been developed to integrate them in MATLAB and `Scilab`.

1 Introduction

The design of low order controllers for high order plants is a challenging problem both theoretically as well as from a computational point of view. The advanced controller design methods like the LQG/LTR loop-shaping, H_∞-synthesis, μ and linear matrix inequalities based synthesis methods produce typically controllers with orders comparable with the order of the plant. Therefore, the orders of these controllers tend often to be too high for practical use, where simple controllers are preferred over complex ones. To allow the practical applicability of advanced controller design methods for high order systems, the model reduction methods capable to address controller reduction problems are of primary importance. Comprehensive presentations of controller reduction methods and the reasons behind different approaches can be found in the textbook [30] and in the recent monograph [16].

Software implementing advanced synthesis procedures is available in both commercial as well as in free CACSD software. For example, freely available high quality numerical software for the LQG/LTR loop-shaping and H_∞ control design approaches has been recently developed within the European Project NICONET[1] and is part of the Fortran 77 control and systems library SLICOT [3]. In contrast, software suitable for controller reduction is scarce and, until now, was only available as commercial software [14]. This is why, a systematic effort has been undertaken within the NICONET Project to complement the available design tools with high quality, numerically robust software for model and controller reduction suitable to be employed in obtaining low order controllers.

In this paper we describe the results of the concentrated effort within the NICONET project to develop a powerful collection of Fortran 77 subroutines for model and controller reduction covering a wide range of possible approaches for obtaining low order controllers for high order plants. The new model and controller reduction software implements the latest algorithmic developments for the following approaches:

(1) relative error model reduction using the balanced stochastic truncation approach [4, 17, 28];
(2) model reduction using frequency-weighted balancing related [6, 10, 29, 26] and frequency-weighted Hankel-norm approximation methods [9, 8, 24];
(3) controller reduction methods using frequency-weighted balancing related methods [12, 27] and coprime factorization based techniques [12].

The new model and controller reduction routines for SLICOT are among the most powerful and numerically most reliable software tools available for model and controller reduction. All routines can be employed to reduce both stable and unstable, continuous- or discrete-time models or controllers. The underlying numerical algorithms rely on *square-root* [18] and *balancing-free square-root* [20] accuracy enhancing techniques. To facilitate the usage of the new routines, easy-to-use and flexible interfaces have been developed to integrate them in two popular user-friendly computing environments: the commercial package MATLAB[2] and the free software `Scilab` [5].

[1] `http://www.win.tue.nl/niconet/niconet.html`
[2] MATLAB is a registered trademark of The MathWorks, Inc.

2 Basic model reduction approaches

In this section we discuss shortly the basic model reduction approaches which are potentially applicable to controller reduction as well, and indicate computational methods suitable to solve the corresponding model reduction problems. Consider the n-th order original state-space model $\mathbf{G} := (A, B, C, D)$ with the *transfer-function matrix* (TFM)

$$G(\lambda) = C(\lambda I - A)^{-1}B + D,$$

and let $\mathbf{G}_r := (A_r, B_r, C_r, D_r)$ be an r-th order approximation of the original model ($r < n$), with the TFM

$$G_r(\lambda) = C_r(\lambda I - A_r)^{-1}B_r + D_r.$$

According to the system type, λ is either the complex variable s appearing in the Laplace transform in the case of a continuous-time system or the variable z appearing in the z-transform in the case of a discrete-time system.

The *absolute error* model reduction methods try to minimize the absolute approximation error

$$\|G - G_r\|_\infty. \qquad (1)$$

For a stable original system $\mathbf{G}$, the *balanced truncation* (BT) [15], the *singular perturbation approximation* (SPA) [11] and the *Hankel-norm approximation* (HNA) [7] are the most frequently employed model reduction approaches. In conjunction with modal separation [22] and coprime factorization [21] techniques, these methods can be employed for the reduction of unstable systems as well. Computational methods with enhanced numerical accuracy have been proposed for the BT method in [18, 20] and for the SPA in [19]. These methods underly the robust numerical software available in the SLICOT library [23]. Note that, applications of this software to solve model reduction problems with dense matrices up to an order of $n = 5000$ have been reported.

The *relative error* model reduction methods have several properties which recommend them for both model and controller reduction. While absolute error methods compute good approximations in terms of peak errors (e.g., H_∞-norm), relative error methods have good approximation properties over the whole frequency range. This is why, it is expected that relative error methods in combination with modal separation techniques are better suited for controller approximation than absolute error methods. The *balanced stochastic truncation* (BST) method [4, 17] is a relative error method which tries to minimize $\|\Delta_r\|_\infty$, where Δ_r is the relative error defined implicitly by $G_r = (I - \Delta_r)G$. If $G(\infty)$ is invertible, this is equivalent to minimize

$$\|G^{-1}(G - G_r)\|_\infty. \qquad (2)$$

For a non-square G, the problem can be still be solved if $G(\infty)$ has full row rank (i.e., no zeros at infinity). For a full column rank G, the same problem can be solved for the dual system with the TFM G^T. It is possible combine the additive and relative approaches by performing the BST method on a modified system with the TFM $[\,G\ \beta I\,]$. A zero value of β leads to a pure relative error minimization, while large positive values of α produce approximations which minimize the absolute approximation error (1). When $\beta \to \infty$, the BST method produces identical results with the BT method. The BST can be employed also in conjunction with the SPA approach [13]. An accuracy enhanced BST algorithm has been proposed in [28].

The above mentioned additive and relative error methods have many convenient features which recommend them in solving model and controller reduction problems. All these methods have *a priori* guaranteed approximation error bounds, which can be employed to determine reduced order models satisfying a given approximation error. Moreover, all these methods, when applied to a stable system, produce stable reduced order approximations. In conjunction with modal separation, all these methods preserve the unstable eigenvalues, which is necessary requirement when these methods are employed for controller reduction.

The methods for *frequency-weighted model reduction* (FWMR) try to minimize a *weighted approximation* error of the form

$$\|W_o(G - G_r)W_i\|_\infty, \qquad (3)$$

where W_o and W_i are suitably chosen output and input weighting TFMs, respectively. The presence of weights reflects the desire that the approximation be more accurate at certain frequencies where W_o and/or W_i have larger singular values. The FWMR approach can be interpreted as an extension and a generalization of the absolute and relative error methods. Many controller reduction problems can be formulated as FWMR problems [1] (see next section).

For the solution of the FWMR, a *frequency-weighted BT* (FWBT) approach, extending the BT method, has been proposed in [6], and further extended by various authors [10, 29, 26]. A *frequency-weighted SPA* (FWSPA) method has been discussed in [26] pointing out better approximation properties than for the FWBT, both in terms of smaller errors as well as of stability preserving. The *frequency-weighted HNA* (FWHNA) has been introduced in [9] for single-input single-output systems, and extended to the multivariable case in [8]. Recent developments [24] extend this method to arbitrary invertible weights by using descriptor systems based projection computations.

3 Basic controller reduction approaches

Let $\mathbf{K} = (A_c, B_c, C_c, D_c)$ be a stabilizing controller of order n_c for the system $\mathbf{G}$. We want to find $\mathbf{K}_r$, an r_c-th order approximation of $\mathbf{K}$ having the same number of unstable poles as $\mathbf{K}$, such that the reduced controller $\mathbf{K}_r$ is stabilizing and the closed-loop system using the reducing controller has satisfactory performances. To solve controller reduction problems, virtually all model reduction methods in conjunction with the modal separation approach (to preserve the unstable poles) can be employed. However, when employing general purpose model reduction methods, the closed-loop stability and performance aspects are completely ignored and the resulting controllers are often unsatisfactory.

To address stability and performance preseving issues, controller reduction problems are frequently formulated as FWMR problems with special weights [1]. This amounts to find $\mathbf{K}_r$, the r_c-th order approximation of $\mathbf{K}$ having the same number of unstable poles as $\mathbf{K}$, such that a weighted error of the form

$$\|W_o(K - K_r)W_i\|_\infty, \tag{4}$$

is minimized, where W_o and W_i are suitably chosen weighting TFMs. To enforce closed-loop stability, one-sided weights of the form

$$W_i = I, \quad W_o = (I + GK)^{-1}G \tag{5}$$

or

$$W_i = G(I + KG)^{-1}, \quad W_o = I \tag{6}$$

can be used, while performance-preserving considerations lead to two-sided weights

$$W_o = (I + GK)^{-1}G, \quad W_i = (I + GK)^{-1} \tag{7}$$

Efficient controller reduction methods to solve (4) for the three particular stability and performance enforcing weights defined in (5), (6) and (7) have been recently proposed in [27] by using frequency-weighted balancing related methods. The new method can be seen as an enhancement of the method of [12], where the FWBT approach has been specialized to the case of a *stable* state-feedback and full-order estimator based controller. In contrast to [12], the new method of [27] is applicable to an arbitrary stabilizing controller, regardless the controller is stable or not.

Feedback controllers resulting from LQG designs have a special structure which allows to apply standard model reduction techniques on appropriate "natural" coprime factorized representations. For example, provided the controller has a left coprime factorization $K = M^{-1}N$, the order reduction can be performed by approximating $[\,M \;\, N\,]$ by some lower order approximation $[\,M_r \;\, N_r\,]$. This leads to a reduced order controller defined in a left factorized form $K_r = M_r^{-1}N_r$. For state-feedback and observer-based controllers, "natural" left/right coprime factorizations can be easily determined.

When using the above approach, there is no guarantee for preserving closed-loop stability. This is why, the coprime factorization approach has been combined with frequency weighting in order to enforce the closed-loop stability [12]. Let $K = M^{-1}N$ be a left coprime factorization of the controller. The controller reduction is solved by minimizing the weighted error

$$\|\,[\,M - M_r \;\; N - N_r\,] \begin{bmatrix} Y \\ X \end{bmatrix} \|_\infty \tag{8}$$

for M_r and N_r, where $G = XY^{-1}$ is a right coprime factorization of the plant. Note that a particularly simple computational method is obtained in the case of using full-order observer-based controllers.

4 New model and controller reduction routines in SLICOT

The implementation of the new model and controller reduction routines in SLICOT relies partly on the model reduction routines for absolute error methods (BT, SPA, HNA) already available in the previous release of SLICOT (see [23]). All implementations rely on the standard linear algebra package LAPACK [2]. Some of the new model reduction routines can be seen as generalizations of the functionality of some of existing routines in SLICOT. The following table contains the list of the new user callable subroutines model and controller reduction available in SLICOT:

Name	Function
AB09HD	BST and BST-SPA approaches
AB09ID	FWBT and FWSPA approaches
AB09JD	FWHNA with invertible proper weights
SB16AD	FWBT/FWSPA-based controller reduction for closed-loop stability and performance preserving weights
SB16BD	state-feedback/full-order observer-based controller reduction using coprime factorization in conjunction with BT and SPA techniques
SB16CD	state-feedback/full-order observer-based controller reduction using frequency-weighted coprime factorization in conjunction with BT technique

In implementing these routines, a special attention has been paid to ensure their numerical robustness. All implemented routines rely on the well-established *square-root* (**SR**) and *balancing-free square-root* (**BFSR**) accuracy enhancing techniques [18, 20, 19]. Furthermore, they optionally perform the scaling of the original system. Both techniques substantially contribute to improve the numerical reliability of computations.

Several of the provided functional facilities are common to all routines. The order of the reduced system can be selected by the user or can be determined automatically on the basis of computed Hankel singular values. Each of routines can handle both continuous- and discrete-time systems. Unstable models are handled by separating the stable and unstable parts and applying the model reduction only to the stable parts. All routines for frequency-weighted model reduction approaches can address weighted problems with one-sided or two-sided weights as well as unweighted problems.

In what follows we shortly discuss some particular functionality provided by the main user callable routines.

The routine AB09HD implements the **SR** and **BFSR** relative-error BST approach in conjunction with BT and SPA techniques [28, 13]. The user can select via a parameter β, the absolute/relative weighting in the approximation error. A large positive value of β favours the minimization of the absolute approximation error, while a small value of β is appropriate for the minimization of the relative error. To implement the discrete-time BST method, bilinear continuous-to-discrete transformations have been employed.

The frequency-weighted balancing related model reduction routine AB09ID, implements the recent enhancements of FWBT and FWSPA approaches proposed in [26] to ensure the stability of approximations in the case of two-sided weights. For this purpose, this routine has a large flexibility in combining different choices of the gramians (see [25] for details). The AB09ID routine is completely general, allowing to handle even unstable weights by solving a transformed approximation problem with the original weights replaced by the numerators of appropriate left and right coprime factorizations with inner denominators. Since AB09ID can handle the unweighted case as well, this single routine practically covers now the complete functionality of AB09AD, AB09BD, AB09MD and AB09ND routines already existing in SLICOT (see [23]).

The FWHNA routine AB09JD implements the enhanced approach proposed in [24], which allows for invertible proper weights satisfying appropriate stability/antistability conditions. AB09JD is very flexible in allowing arbitrary combinations of four types of input and output weights: standard, inverse, conjugated, and conjugated inverse. Furthermore, AB09JD allows the user to choose between using projection formulas involving explicit inverses (if they exist) or employing the computationally more expensive, but numerically more reliable, inversion-free approaches. This latter option is always used when the feedtrough matrices of the weights are exactly or nearly singular. For implementing the discrete-time FWHNA method, bilinear continuous-to-discrete transformations are employed. Since AB09JD can handle the unweighted case as well, this routine also covers the functionality of existing AB09CD and AB09ED routines.

The controller reduction routine SB16AD is practically a specialization of AB09ID to the case of controller reduction with special weights used to enforce closed-loop stability and performance when using the reduced controller instead of the full order one. This routine works on a general stabilizing controller.

The coprime factorization based controller reduction routines SB16BD and SB16CD are specially adapted to reduce state feedback and observer-based controllers. The routine SB16BD allows arbitrary combinations of BT and SPA methods with "natural" left and right coprime factorizations of the controller. The routine SB16CD, implementing the frequency-weighted coprime factorization based approach, can be employed only in conjunction with the BT technique. This routine allows to work with both left and right coprime factorization based approaches.

In implementing the new model and controller reduction software, a special emphasis has been put on an appropriate modularization of the routines by isolating some basic computational tasks and implementing them in supporting computational routines. For example, the balancing related approaches (implemented in AB09HD, AB09ID, SB16AD), as well as the frequency-weighted coprime factorization based controller reduction method (implemented in SB16CD), share a common two step computational scheme: (1) compute two non-negative definite matrices P and Q, called generically "gramians"; (2) determine suitable truncation matrices and apply them to obtain the matrices of the reduced model/controller using the BT or SPA methods. For the first step, separate routines have been implemented to compute appropriate gramians according to the specifics of each method. To employ the accuracy enhancing **SR** or **BFSR** techniques, these routines compute in fact, instead of gramians, their Cholesky factors. For the second step, a unique routine has been implemented, which is called by all above routines. Similarly, to compute the stable projections for the FWHNA involving various types of weights, two supporting routines for handling the left and right weights, have been implemented. These routines cover both the inversion-based as well as the inversion-free projection formulas [24].

An important number of new user callable routines have been implemented for the special needs of the new model and controller reduction routines: computation of inverse systems, solution of continuous- and discrete-time Sylvester equations, evaluation of the normal rank of a TFM in state-space form, computation of the L_∞-norm, or performing an orthogonal reductions to generalized Hessenberg form. Additionally, many low level auxiliary routines have been implemented to perform basic control computations. For a detailed description of the new software, see [25]

5 Integration in user-friendly environments

One of the main objectives of the NICONET project was to provide, additionally to standardized Fortran codes, high quality software embedded into user-friendly environments for *computer aided control system design*. Two target environments have been envisaged: the popular commercial numerical computational environment MATLAB and the public domain MATLAB-like environment `Scilab`. Both allows to easily add external functions implemented in general purpose programming languages like C or Fortran. In the case of MATLAB, the external functions are called *mex*-functions and have to be programmed according to precise programming standards. In `Scilab`, external functions can be similarly implemented and only several minor modifications are necessary to the MATLAB *mex*-functions to adapt them to `Scilab`.

Several *mex*-functions, similar to the additive error function `sysred` [23], have been implemented as main MATLAB interfaces to the new model and controller reduction routine available in SLICOT. To provide a convenient interface to work with control objects defined in the MATLAB Control Toolbox, several easy-to-use higher level model and controller reduction *m*-functions have been additionally implemented. The list of available *mex*- and *m*-functions is given below:

Name	Function
mex: `bstred` *m*: `bst`	balanced stochastic truncation based model reduction (based on AB09HD)
mex: `fwered` *m*: `fwbred`	frequency-weighted balancing related model reduction (based on AB09ID)
mex: `fwehna` *m*: `fwhna`	frequency-weighted Hankel-norm approximation (based on AB09JD)
mex: `conred` *m*: `fwbconred`	frequency-weighted balancing related controller reduction (based on SB16AD)
mex: `sfored` *m*: `sfconred`	coprime factorization based reduction of state feedback controllers (based on SB16BD and SB16CD)

All functions are able to reduce both continuos- and discrete-time, stable as well as unstable systems or controllers. If appropriate, the functions can be used for unweighted reduction as well, without any significant computational overhead.

In the implemention of the *mex*- and *m*-functions, one main goal was to allow the access to all functional facilities provided by the underlying Fortran routines. To manage the multitude of user options, a so-called SYSRED structure has been defined. This structure is created and managed via special functions. A selection of options which can be set via the SYSRED structure are shown below:

```
     BalredMethod: [ {bta} | spa ]
AccuracyEnhancing: [ {bfsr} | sr ]
           Tolred: [ positive scalar {0} ]
       TolMinreal: [ positive scalar {0} ]
            Order: [ integer {-1} ]
          BstBeta: [ scalar {0}  ]
  FWEContrGramian: [ {standard} | enhanced ]
 FWEObservGramian: [ {standard} | enhanced ]
    CoprimeFactor: [ left | {right} ]
     OutputWeight: [ {stab} | perf | none]
      InputWeight: [ {stab} | none]
   CFConredMethod: [ {fwe}  | nofwe ]
  FWEConredMethod: [ none | outputstab |
                     inputstab | {performance} ]
     FWEHNAMethod: [ {auto} | inv | noinv ]
        FWEHNAopV: [ {none} | inv | conj | cinv ]
        FWEHNAopW: [ {none} | inv | conj | cinv ]
```

6 Conclusion

A powerful collection of user callable Fortran 77 routines has been implemented for model and controller reduction. The new software is based on the latest algorithmic developments and covers the relative error model reduction using the *balanced stochastic truncation* approach, model reduction using *frequency-weighted balancing* and *frequency-weighted Hankel-norm approximation* methods, as well as special controller reduction methods using *frequency-weighted balancing* and *coprime factorization* based techniques. All implemented routines can be employed to reduce both stable and unstable, continuous- or discrete-time models or controllers. The underlying numerical algorithms are based on recent extensions of the *square-root* and *balancing-free* accuracy enhancing techniques to frequency-weighted balancing-related model reduction. The new model and controller reduction routines for SLICOT are among the most powerful and numerically most reliable software tools available for model and controller reduction. To facilitate their usage, easy-to-use and flexible interfaces have been developed to integrate them in MATLAB and `Scilab`.

Acknowledgments. The author thanks Vasile Sima and Diana Sima for their substantial contributions to software implementations.

References

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

[2] 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, Third Edition*. SIAM, Philadelphia, 1999.

[3] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT – a subroutine library in systems and control theory. In B. N. Datta (Ed.), *Applied and Computational Control, Signals and Circuits*, vol. 1, pp. 499–539. Birkhäuser, 1999.

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

[5] C. Gomez (Ed.). *Engineering and Scientific Computing with Scilab*. Birkhauser, Boston, 1999.

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

[7] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *Int. J. Control*, 39:1115–1193, 1984.

[8] Y. S. Hung and K. Glover. Optimal Hankel-norm approximation of stable systems with first-order stable weighting functions. *Systems & Control Lett.*, 7:165–172, 1986.

[9] 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.

[10] C.-A. Lin and T.-Y. Chiu. Model reduction via frequency weighted balanced realization. *CONTROL - Theory and Advanced Technology*, 8:341–351, 1992.

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

[12] Y. Liu, B. D. O. Anderson, and U. L. Ly. Coprime factorization controller reduction with Bezout identity induced frequency weighting. *Automatica*, 26:233–249, 1990.

[13] M. Green and B.D.O. Anderson. Generalized balanced stochastic-truncation. *Proc. 29th CDC*, pp. 476–481, 1990.

[14] $\mathrm{MATRIX_X}$. *Xmath Model Reduction Module*. ISI, Santa Clara, CA, January 1998.

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

[16] G. Obinata and B. D. O. Anderson. *Model Reduction for Control System Design*. Springer Verlag, Berlin, 2000.

[17] 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.

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

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

[20] A. Varga. Efficient minimal realization procedure based on balancing. In A. El Moudni, P. Borne, and S. G. Tzafestas (Eds.), *Proc. of IMACS/IFAC Symp. on Modelling and Control of Technological Systems, Lille, France*, vol. 2, pp. 42–47, 1991.

[21] A. Varga. Coprime factors model reduction based on square-root balancing-free techniques. In A. Sydow (Ed.), *Computational System Analysis 1992, Proc. 4-th Int. Symp. Systems Analysis and Simulation, Berlin, Germany*, pp. 91–96. Elsevier, Amsterdam, 1992.

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

[23] A. Varga. Model reduction software in the SLICOT library. In B. N. Datta (Ed.), *Applied and Computational Control, Signals and Circuits*, vol. 629 of *The Kluwer International Series in Engineering and Computer Science*, pp. 239–282. Kluwer Academic Publishers, Boston, 2001.

[24] A. Varga. Numerical approach for the frequency-weighted Hankel-norm approximation. *Proc. of ECC'2001, Porto, Portugal*, pp. 640–645, 2001.

[25] A. Varga. New Numerical Software for Model and Controller Reduction. NICONET Report 2002-5, June 2002. `ftp://wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/nic2002-5.pdf`.

[26] A. Varga and B. D. O. Anderson. Square-root balancing-free methods for the frequency-weighted balancing related model reduction. *Proc. of CDC'2001, Orlando, FL*, pp. 3659–3664, 2001.

[27] A. Varga and B. D. O. Anderson. Frequency-weighted balancing related controller reduction. In *Proc. of IFAC'2002 Congress, Barcelona, Spain*, 2002.

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

[29] G. Wang, V. Sreeram, and W. Q. Liu. A new frequency-weighted balanced truncation method and error bound. *IEEE Trans. Autom. Control*, 44:1734–1737, 1999.

[30] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.