Katholieke Universiteit Leuven

Departement Elektrotechniek ESAT-SISTA/TR 1994-42

A Note on Closed-Loop Balanced Truncation. [1] [2]

Geert Schelfhout and Bart De Moor[3]

September 1994
Submitted for publication

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/SISTA/schelfhout/reports/clbr.ps.Z*

[2]The following text presents research results obtained within the framework of the Belgian programme on interuniversity attraction poles (IUAP-17 and IUAP-50) initiated by the Belgian State - Prime Minister's Office - Science Policy Programming. The scientific responsibility is assumed by its authors.

[3]ESAT - Katholieke Universiteit Leuven,
Kardinaal Mercierlaan 94, 3001 Leuven (Heverlee), Belgium,
tel. 32/16/32.18.08, fax 32/16/32.19.86,
geert.schelfhout@esat.kuleuven.ac.be *bart.demoor@esat.kuleuven.ac.be*.

A Note on Closed-Loop Balanced Truncation. *

Geert Schelfhout[†] Bart De Moor[‡]

ESAT–SISTA

Katholieke Universiteit Leuven

Kard. Mercierlaan 94

B-3001 Leuven

BELGIUM

Tel.: 00-32-16-32 18 08 FAX: 00-32-16-32 19 86

e-mail : geert.schelfhout@esat.kuleuven.ac.be bart.demoor@esat.kuleuven.ac.be

Keywords: controller reduction, frequency weighted model reduction, observer-based controller

Abstract

Closed-Loop Balanced Truncation is a model reduction method that aims at preserving the closed-loop properties of the original controller. We show that this method is equivalent to Frequency Weighted Balanced Truncation with certain weightings, and how for observer-based controllers, a significant amount of computations can be saved because of the observer–state feedback structure of the controller.

*This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction (IUAP-17 and IUAP-50), initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors.

[†]GS is a research assistant with the NFWO (Belgian National Fund for Scientific Research)

[‡]BDM is a senior research associate with the NFWO

1 Introduction and Background : Closed-Loop Balanced Truncation (CLBT)

Model reduction has become an important tool in modern control design. Many modeling procedures yield models of rather high order, as in general higher order implies better accuracy; this is especially true for spatially discretized systems, obtained via e.g. Finite Element Methods (FEM). Often modern control design strategies yield controllers of the same order as the process model or higher, depending on the control specifications. When implementing such a controller, however, the high order may be a serious drawback; fortunately, one often finds that the order of these controllers may be reduced considerably without having to relax the specified constraints or impairing the performance too much.

That closed-loop considerations have to be taken into account while designing a controller of reduced order has been pointed out before. A reduced order model may mimic the input-output behavior of a plant rather well, but a control law based on this model may cause unsatisfactory closed-loop behavior because the seemingly insignificant discarded dynamics have greater impact on the closed-loop than on the open-loop transfer; this phenomenon is called *spillover*. A similar thing may happen when the controller is obtained by approximating a controller, stabilizing the full order plant in a satisfactory way, by a low-order one. Many authors have contributed model reduction methods more suitable w.r.t. control design, in the sense that they concentrate more on critical frequency intervals [1, 13, 12, 17, 8, 9].

De Villemagne and Skelton [18] proposed an approach focused on putting the closed-loop controllability gramian in a form that enables one to see which combinations of states are insignificant, but does not employ balancing.
This inspired the Closed-Loop Balanced Truncation (CLBT) approach, an intuitive but remarkably useful method of controller reduction proposed recently by Ceton e.a. [5, 19]. A scheme that shews some similarity is [6]. In Section 2, we show that CLBT can also be viewed as a frequency weighted model reduction method, where the weights are chosen so as to retain closed-loop performance. The main contribution of this paper is to be found in Section 3, where it is shown how significant savings can be made in the number of computations in the important special case of observer-based controllers. In the remainder of this section, we recapitulate the idea of CLBT in the framework of the standard plant

configuration.

Some notation: $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \triangleq C(sI-A)^{-1}B+D$

Suppose one has a standard plant

$$\begin{bmatrix} P_{zw}(s) & P_{zu}(s) \\ P_{yw}(s) & P_{yu}(s) \end{bmatrix} \triangleq \left[\begin{array}{c|cc} A & B_w & B_u \\ \hline C_z & D_{zw} & D_{zu} \\ C_y & D_{yw} & D_{yu} \end{array}\right]$$

, and one has a controller $C(s) \triangleq \left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array}\right]$

that yields a satisfactory closed-loop transfer matrix $H_{zw} = P_{zw} + P_{zu}C(I - P_{yu}C)^{-1}P_{yw}$. Let n be the order of the standard plant $P(s)$ and n_c be the order of $C(s)$. Note that we may assume without loss of generality that $D_{yu} = 0$, since if $P_{yu}(s)$ is not strictly proper, the problem may always be transformed into another one where $P_{yu}(s)$ is strictly proper by means of a loop transformation [10]. In such a case, a state space representation for H_{zw} is

$$H_{zw}(s) \triangleq \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right] = \left[\begin{array}{cc|c} A + B_u D_c C_y & B_u C_c & B_w + B_u D_c D_{yw} \\ B_c C_y & A_c & B_c D_{yw} \\ \hline C_z + D_{zu} D_c C_y & D_{zu} C_c & D_{zw} + D_{zu} D_c D_{yw} \end{array}\right] \quad (1)$$

The controllability gramian $\tilde{P} \triangleq \begin{bmatrix} P_p & P_{pc} \\ P_{pc}^T & P_c \end{bmatrix}$ for this realization of the closed-loop system is computed from $\tilde{A}\tilde{P} + \tilde{P}\tilde{A}^T + \tilde{B}\tilde{B}^T = 0$ and an observability gramian $\tilde{Q} \triangleq \begin{bmatrix} Q_p & Q_{pc} \\ Q_{pc}^T & Q_c \end{bmatrix}$ from $\tilde{A}^T\tilde{Q} + \tilde{Q}\tilde{A} + \tilde{C}^T\tilde{C} = 0$. In order to reduce the order of the controller, while changing the closed-loop transfer function as little as possible, we may perform balanced truncation on the controller $\left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array}\right]$ using P_c and Q_c, the parts of these gramians relevant to the controller states [5]. This procedure may be rather intuitive and of course cannot guarantee stability of the reduced controller, which would not be a very useful result anyway, or of the closed-loop system after reduction, which would be a result too beautiful to be true, but it compares very favorably with many well-known controller reduction methods. To provide some further motivation, we proceed to show that CLBT is equivalent to Enns's well known Frequency Weighted Balanced Truncation (FWBT) with certain performance-oriented weights.

2 Equivalence with Frequency Weighted Balanced Truncation

Enns's FWBT [8, 9, 1] aims at making $\|W_o(s)(G(s)-G_r(s))W_i(s)\|_\infty$ small. Defining $\delta H_{zw} \triangleq P_{zw} + P_{zu}C_r(I-P_{yu}C_r)^{-1}P_{yw} - H_{zw}$, where C_r is a controller of order n_r, and $\delta C \triangleq C_r - C$, one easily finds the expression $\delta H_{zw} \approx P_{zu}(I-CP_{yu})^{-1}\delta C(I-P_{yu}C)^{-1}P_{yw}$, which is correct up to first order terms in δC. Therefore very natural weighting functions for performance-oriented controller reduction are $W_o(s) \triangleq P_{zu}(I-CP_{yu})^{-1}$ and $W_i(s) \triangleq (I-P_{yu}C)^{-1}P_{yw}$. Realizations for these are readily found to be:

$$W_o(s) = \left[\begin{array}{cc|c} A+B_uD_cC_y & B_uC_c & B_u \\ B_cC_y & A_c & 0 \\ \hline C_z+D_{zu}D_cC_y & D_{zu}C_c & D_{zu} \end{array}\right] \text{ and } W_i(s) = \left[\begin{array}{cc|c} A+B_uD_cC_y & B_uC_c & B_w+B_uD_cD_{yw} \\ B_cC_y & A_c & B_cD_{yw} \\ \hline C_y & 0 & D_{yw} \end{array}\right].$$

Enns's scheme with these weightings first computes the controllability gramian $\bar{P}$ of

$$C(s)W_i(s) = \left[\begin{array}{ccc|c} A+B_uD_cC_y & B_uC_c & 0 & B_w+B_uD_cD_{yw} \\ B_cC_y & A_c & 0 & B_cD_{yw} \\ B_cC_y & 0 & A_c & B_cD_{yw} \\ \hline D_cC_y & 0 & C_c & D_cD_{yw} \end{array}\right]$$

and the observability gramian $\bar{Q}$ of

$$W_o(s)C(s) = \left[\begin{array}{ccc|c} A+B_uD_cC_y & B_uC_c & B_uC_c & B_uD_c \\ B_cC_y & A_c & 0 & 0 \\ 0 & 0 & A_c & B_c \\ \hline C_z+D_{zu}D_cC_y & D_{zu}C_c & D_{zu}C_c & D_{zu}D_c \end{array}\right],$$

balances the lower right $n_c \times n_c$ blocks of $\bar{P}$ and $\bar{Q}$, and truncates the controller accordingly. Replacing the state x_i of the realization of $C(s)W_i(s)$ by Tx_i, where $T \triangleq \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & I \end{bmatrix}$, and the state x_o of the realization of $W_o(s)C(s)$ by $(T^{-1})^Tx_o$, yields two new realizations:

$$C(s)W_i(s) = \left[\begin{array}{ccc|c} A+B_uD_cC_y & B_uC_c & 0 & B_w+B_uD_cD_{yw} \\ B_cC_y & A_c & 0 & B_cD_{yw} \\ 0 & 0 & A_c & 0 \\ \hline D_cC_y & C_c & C_c & D_cD_{yw} \end{array}\right],$$

which is seen, by inspection of (1), to have $\begin{bmatrix} \tilde{P} & 0 \\ 0 & 0 \end{bmatrix}$ for controllability gramian, and

$$W_o(s)C(s) = \left[\begin{array}{ccc|c} A + B_u D_c C_y & B_u C_c & 0 & B_u D_c \\ B_c C_y & A_c & 0 & B_c \\ 0 & 0 & A_c & B_c \\ \hline C_z + D_{zu} D_c C_y & D_{zu} C_c & 0 & D_{zu} D_c \end{array}\right],$$

which has $\begin{bmatrix} \tilde{Q} & 0 \\ 0 & 0 \end{bmatrix}$ for observability gramian. Hence

$$\bar{P} = T^T \begin{bmatrix} \tilde{P} & 0 \\ 0 & 0 \end{bmatrix} T = \begin{bmatrix} P_p & P_{pc} & P_{pc} \\ P_{pc}^T & P_c & P_c \\ P_{pc}^T & P_c & P_c \end{bmatrix} \text{ and } \bar{Q} = T^T \begin{bmatrix} \tilde{Q} & 0 \\ 0 & 0 \end{bmatrix} T = \begin{bmatrix} Q_p & Q_{pc} & Q_{pc} \\ Q_{pc}^T & Q_c & Q_c \\ Q_{pc}^T & Q_c & Q_c \end{bmatrix}.$$

Hence, both in FWBT with the weightings above and in CLBT, P_c and Q_c are balanced, and the methods are equivalent; but as the size of the gramians to be computed in CLBT is only $n + n_c$ compared to $n + 2n_c$ in FWBT, the former is more efficient. Moreover, with CLBT there are no restrictions on the poles of the controller: these may be unstable or even lie on the imaginary axis.

Even more savings can be made when the controller is observer-based, as will be shown in the next section.

3 Observer-based controllers

In many control design techniques, a controller is found as the combination of a state feedback controller and a state observer, e.g. when using pole placement or in LQG design [7]. For this type of controllers, the order of the controller equals the order of the plant n, hence the procedure above would normally require the solution of two $2n \times 2n$ Lyapunov equations. However, in this case , the computation of the gramians P_c and Q_c may be simplified. Suppose the state equation of the state estimator is given by $\dot{x}_c = Ax_c + B_u u + K_e(C_y x_c - y)$ and the state feedback gain is K_r, then the controller has state space representation $C(s) = \left[\begin{array}{c|c} A + B_u K_r + K_e C_y & -K_e \\ \hline K_r & 0 \end{array}\right]$, and for the closed-loop system one finds :

$$H_{zw}(s) = \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & D_{zw} \end{array}\right] = \left[\begin{array}{cc|c} A & B_uK_r & B_w \\ -K_eC_y & A+B_uK_r+K_eC_y & -K_eD_{yw} \\ \hline C_z & D_{zu}K_r & D_{zw} \end{array}\right]$$

Defining $T \triangleq \begin{bmatrix} I & I \\ 0 & I \end{bmatrix}$, $\tilde{A}_1 \triangleq T^{-1}\tilde{A}T = \begin{bmatrix} A+K_eC_y & 0 \\ -K_eC_y & A+B_uK_r \end{bmatrix}$, $\tilde{A}_2 \triangleq T^{-T}\tilde{A}T^T = \begin{bmatrix} A+B_uK_r & B_uK_r \\ 0 & A+K_eC_y \end{bmatrix}$, $\tilde{B}_1 \triangleq T^{-1}\tilde{B} = \begin{bmatrix} B_w+K_eD_{yw} \\ -K_eD_{yw} \end{bmatrix}$, $\tilde{B}_2 \triangleq T^{-T}\tilde{B} = \begin{bmatrix} B_w \\ -B_w-K_eD_{yw} \end{bmatrix}$, $\tilde{C}_1 \triangleq \tilde{C}T = \begin{bmatrix} C_z & C_z+D_{zu}K_r \end{bmatrix}$, and $\tilde{C}_2 \triangleq \tilde{C}T^T = \begin{bmatrix} C_z+D_{zu}K_r & D_{zu}K_r \end{bmatrix}$, we have:

$$H_{zw}(s) = \left[\begin{array}{c|c} \tilde{A}_1 & \tilde{B}_1 \\ \hline \tilde{C}_1 & D_{zw} \end{array}\right] = \left[\begin{array}{c|c} \tilde{A}_2 & \tilde{B}_2 \\ \hline \tilde{C}_2 & D_{zw} \end{array}\right]$$

Now if $\tilde{A}_1\tilde{P}_1 + \tilde{P}_1\tilde{A}_1^T + \tilde{B}_1\tilde{B}_1^T = 0$ and $\tilde{A}_2^T\tilde{Q}_2 + \tilde{Q}_2\tilde{A}_2 + \tilde{C}_2^T\tilde{C}_2 = 0$, where $\tilde{P}_1$ and $\tilde{Q}_2$ are partitioned as before: $\tilde{P}_1 \triangleq \begin{bmatrix} P_{p1} & P_{pc1} \\ P_{pc1}^T & P_{c1} \end{bmatrix}$ and $\tilde{Q}_2 \triangleq \begin{bmatrix} Q_{p2} & Q_{pc2} \\ Q_{pc2}^T & Q_{c2} \end{bmatrix}$;
then $\tilde{P} = T\tilde{P}_1T^T$ and $\tilde{Q} = T^{-1}\tilde{Q}_2T^{-T}$, and it is easily seen that $P_c = P_{c1}$ and $Q_c = Q_{c2}$. Therefore CLBT is performed by solving consecutively the following six Sylvester equations :

$$(A+K_eC_y)P_{p1} + P_{p1}(A+K_eC_y)^T + (B_w+K_eD_{yw})(B_w+K_eD_{yw})^T = 0 \qquad (2)$$

$$(A+K_eC_y)P_{pc1} + P_{pc1}(A+B_uK_r)^T - P_{p1}C_y^TK_e^T - (B_w+K_eD_{yw})(K_eD_{yw})^T = 0 \qquad (3)$$

$$(A+B_uK_r)P_{c1} + P_{c1}(A+B_uK_r)^T - K_eC_yP_{pc1} - P_{pc1}^TC_y^TK_e^T + K_eD_{yw}(K_eD_{yw})^T = 0 \qquad (4)$$

and

$$(A+B_uK_r)^TQ_{p2} + Q_{p2}(A+B_uK_r) + (C_z+D_{zu}K_r)^T(C_z+D_{zu}K_r) = 0 \qquad (5)$$

$$(A+B_uK_r)^TQ_{pc2} + Q_{pc2}(A+K_eC_y) + Q_{p2}B_uK_r + (C_z+D_{zu}K_r)^TD_{zu}K_r = 0 \qquad (6)$$

$$(A+K_eC_y)^TQ_{c2} + Q_{c2}(A+K_eC_y) + K_r^TB_u^TQ_{pc2} + Q_{pc2}^TB_uK_r + K_r^TD_{zu}^TD_{zu}K_r = 0, \qquad (7)$$

which will yield $P_{c1} = P_c$ and $Q_{c2} = Q_c$.

Remark that these computations may be done most efficiently if $A+B_uK_r$ and $A+K_eC_y$ are first put in Schur form, since then, solving the six Sylvester equations above requires only two Schur decompositions instead of six. The operation counts below are based on the assumption that the computation of a Schur decomposition costs $25n^3$ flops; this value is of course only approximate [11, p. 380]. $2n^3$ flops are needed to solve a Sylvester equation once the coefficient matrices have been put in Schur form (n^3 for a Lyapunov equation). Counting this way, and noting that four of the six Sylvester equations are actually Lyapunov equations, P_{c1} and Q_{c2} may be computed using about $2(25n^3+2n^3)+4n^3 = 58n^3$ flops, whereas computing the two $2n\times 2n$ closed-loop gramians $\tilde{P}$ and $\tilde{Q}$ requires $25(2n)^3+2(2n)^3 = 216n^3$ flops.

Note furthermore that the computation is even less costly if the eigenvalue decompositions $A + B_uK_r = T_r^{-1}\Lambda_r T_r$ and $A + K_eC_y = T_e^{-1}\Lambda_e T_e$ are available, as will often be the case when K_r and K_e have been obtained using a pole placement technique; then solving each of the six equations above is done in $O(n^2)$ flops. Indeed, these represent similarity transformations that bring $A + B_uK_r$ and $A + K_eC_y$ into diagonal form, so that the Sylvester equations are easily seen to be solved in $O(n^2)$ flops (note that it is trivial to bring $A + B_uK_r$ and $A + K_eC_y$ into real quasi-diagonal form if one wishes to stick to real computations). The price to be paid for this, however, is a loss of numerical reliability, since these transformations are no longer orthogonal.

W.r.t. memory needed, the six Sylvester equations above may be solved consecutively, so that all one needs to store into memory at one time is three $n\times n$ matrices (four when solving the Sylvester equations).

Questions may arise as to whether the procedure given above is as numerically reliable as the normal approach of finding a Schur decomposition of $\tilde{A}$; after all, the similarity transformations using the matrix T constitute non-orthogonal operations as well. But this point is not really valid: the matrices A_i, B_i, C_i, $i = 1, 2$, are not computed as $T^{-1}\tilde{A}T$, etc., but immediately as $\begin{bmatrix} A + B_uK_r & B_uK_r \\ 0 & A + K_eC_y \end{bmatrix}$, etc. Therefore, it would be hard to determine a priori whether e.g. the equation $\tilde{A}_1\tilde{P}_1 + \tilde{P}_1\tilde{A}_1^T + \tilde{B}_1\tilde{B}_1^T = 0$ has worse or better condition than the equation $\tilde{A}\tilde{P} + \tilde{P}\tilde{A}^T + \tilde{B}\tilde{B}^T = 0$.

The usual way of solving a Lyapunov equation $AP + PA^T + BB^T = 0$ is to bring A into (quasi-)triangular form by means of a Schur decomposition, and to solve the transformed Lyapunov equation by means of forward or backward substitution [11, 3]. The splitting of a $2n\times 2n$ Lyapunov equation into three Sylvester equations amounts to a block version of this substitution process. It should be noted that this artifact is not really necessary: if $U_r^T(A + B_uK_r)U_r$ and $U_e^T(A + K_eC_y)U_e$ are in real Schur form,

$\begin{bmatrix} U_e & 0 \\ 0 & U_r \end{bmatrix}^T \tilde{A}_1 \begin{bmatrix} U_e & 0 \\ 0 & U_r \end{bmatrix}$ and $\begin{bmatrix} U_r & 0 \\ 0 & U_e \end{bmatrix}^T \tilde{A}_2 \begin{bmatrix} U_r & 0 \\ 0 & U_e \end{bmatrix}$ will also have the quasi-triangular form necessary for starting the substitution process. The only advantage of this splitting into three equations is the memory argument mentioned before; the operation count is identical for both ways, and w.r.t. accuracy, no remarkable differences were recorded.

A word of caution is in order w.r.t. the fact that only two Schur decompositions are needed to solve equations (2),(3),(4),(5),(6),(7). In principle any Schur decomposition of $A + B_u K_r$ and $A + K_e C_y$ may be used for solving these equations, but for large Sylvester or Lyapunov equations the eigenvalue ordering of the Schur matrices may be important. The QR algorithm naturally yields an ordering with the eigenvalues of the Schur matrix going from large absolute values in the left upper corner to small ones in the right lower corner; this is numerically beneficial for solving eqs. (5),(6) and (7), but in equal degree numerically adverse for eqs. (2),(3) and(4), if the same Schur decompositions are used without any reordering. Therefore, an inverse ordering, which is done using orthogonal transformations only, may be advisable for eqs. (2),(3) and (4), but requires about extra $6n^3$ operations [11, 16], taking the cost to compute P_c and Q_c to $62n^3$ operations.

Note, however, that the same remark applies to the computation of $\tilde{P}$ and $\tilde{Q}$, where we also computed only the Schur decomposition of $\tilde{A}$ and used this for solving both Lyapunov equations. A reordering is equally advisable here, but requires $6(2n)^3 = 48n^3$ operations, adding up to a total cost of $264n^3$ operations.

To avoid the trouble of reordering the Schur decompositions, one might also opt to recompute the Schur decomposition of transposes : in this case our scheme has an operation count of $108n^3$, compared to $416n^3$ for the approach computing $\tilde{P}$ and $\tilde{Q}$.

Once P_{c1} and Q_{c2} are computed, the reduced order controller may be found. For this purpose, it is not necessary to actually balance P_{c1} and Q_{c2}; instead, one may use e.g. Safonov's Schur method [15].

For easy reference, the overall procedure is recapitulated here:

- Consecutively solve equations (2),(3),(4) and (5),(6),(7).
- Compute matrices V_r and V_l, whose n_r columns form bases for the right and left eigenspaces of $P_{c1}Q_{c2}$ associated with the r biggest eigenvalues.
- Compute a singular value decomposition $U_e \Sigma_e V_e^T = V_l^T V_r$.

- Defining $L \triangleq \Sigma_e^{-\frac{1}{2}} U_e^T V_l^T$ and $R \triangleq V_r V_e \Sigma_e^{-\frac{1}{2}}$, $A_r \triangleq L(A + BK_r + K_e C)R$, $B_r \triangleq -LK_e$, $C_r \triangleq K_r R$, the reduced order controller is $C_r(s) \triangleq \left[\begin{array}{c|c} A_r & B_r \\ \hline C_r & 0 \end{array}\right]$.

Evidently this reduced order controller may be considered as the series connection of a reduced order observer $\left[\begin{array}{c|c} L(A + BK_r + K_e C)T & -LK_e \\ \hline T & 0 \end{array}\right]$ and the optimal state feedback gain K_r.

4 Example

The example, a flutter control system for a B-767 airplane, is taken from [13]. The specifications are as in [2], not as in [13]. The standard plant is then, in the notation of [2],

$$P(s) = \left[\begin{array}{c|ccc} A & B_n & 0 & -B \\ \hline Q^{\frac{1}{2}} C_n & 0 & 0 & 0 \\ 0 & 0 & 0 & R^{\frac{1}{2}} \\ C & 0 & V^{\frac{1}{2}} & 0 \end{array}\right]$$

which is used for an H_2 optimal design.

When the Schur decompositions are not reordered, but recomputed for the transposes, the total operation count for this example is 24.9×10^6 for our computational scheme and 69.2×10^6 for the method computing $\tilde{P}$ and $\tilde{Q}$. This count includes the Schur decomposition of $P_c Q_c$ etc., which is why it reduced only by 65 % rather than the 75 % mentioned above.

Even though our contribution offers only equivalent ways of computing a CLBT, a scheme put forward in [5], we include some figures indicating the performance, to make an important point about the applicability of CLBT.
In Table 1, the closed-loop damping of the flutter mode (i.e. the stability margin) and the H_2 norm of the closed-loop transfer matrix H_{zw} are compared for reduced order controllers of order 10, 9, 8, 7, 6, 5, 4 and 3, for the weighted left coprime factorization reduction (LCFW) method of [13] and the CLBT method. The closed-loop H_2 norm for the full order controller is 0.9666, the flutter mode damping is 0.0296; the values in Table 1 are

Order	Damping		$\|H_{zw}\|_2/\|H_{zwf}\|_2$	
	LCFW	CLBT	LCFW	CLBT
10	0.6053	0.9033	1.0309	1.0051
9	0.6554	0.9761	1.0559	1.0103
8	-0.0411	0.9205	∞	1.0085
7	0.3586	0.9465	1.0630	1.0107
6	-0.1332	0.0521	∞	1.0510
5	0.0423	0.1448	1.0902	1.1886
4	0.9175	0.1066	1.2215	2.0577
3	0.0180	-0.0105	2.4712	∞

Table 1: Damping : closed-loop damping of the flutter mode divided by the full order closed-loop damping.
$\|H_{zw}\|_2/\|H_{zwf}\|_2$: closed-loop H_2 norm divided by the full order closed-loop H_2 norm. $\|H_{zw}\|_2 = \infty$ indicates closed-loop instability.

relative to these numbers.

Figure 1 shows the largest singular value of the closed-loop transfer function $H_{zw}(s)$ on the imaginary axis for the full order controller and for 4th and 7th order controllers produced by LCFW and CLBT.

This example shows that closed-loop balanced truncation is performance oriented rather than stability oriented, in the sense that for very low orders (say 3-6), where other reduction methods often yield stabilizing yet significantly suboptimal controllers, CLBT does not give a stable or satisfactory closed-loop, whereas for slightly larger orders (say 7-10), it yields a closed-loop performance much closer to the optimum than the other controller reduction methods. It seems very well suited, therefore, for problems where nearly optimal performance with reasonably small controller order is desired, but where it is not absolutely necessary to achieve the lowest possible order.

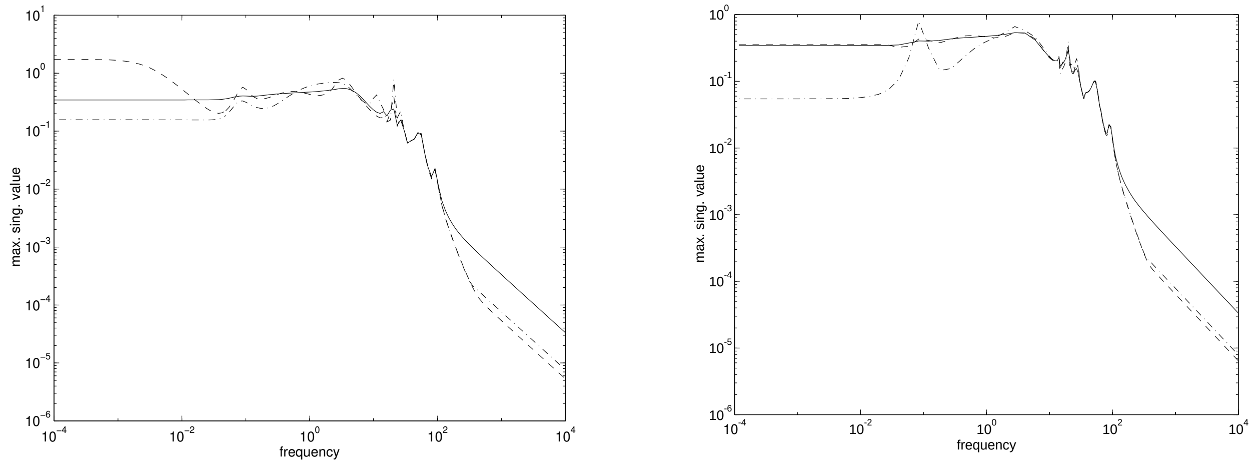

Figure 1: Maximal singular value of $H_{zw}(j\omega)$ for example 2.

Full line : 55th order controller, dashed line : CLBT reduced controller, dash-dotted line : LCFW reduced controller.

Left: 4th order controller, right: 7th order controller.

5 Conclusions

Closed Loop Balanced Truncation [5, 19], is shown to be equivalent to (but more efficient than) Enns's Frequency Weighted Balanced Truncation [8] with certain performance-oriented weightings.

Furthermore, it is indicated how the computational effort of CLBT may be reduced considerably if the method is used to reduce the order of an observer-based controller, as obtained e.g. using pole placement or LQG techniques. In this case, the structure of the system matrices enables one to find the relevant subblocks of the closed-loop gramians using about one fourth of the number of operations needed otherwise.

6 Acknowledgement

The authors wish to thank Prof. Paul Van Dooren for some helpful suggestions.

References

[1] B. D. O. Anderson and J. B. Moore, "Optimal Control - Linear Quadratic Methods," (Chapter 10), *Prentice-Hall*, 1989.

[2] B. D. O. Anderson, U.-L. Ly, and Y. Liu, "Lower Order Controller Design for a High Order Plant," *Benchmark Problems for Control System Design, Report of the IFAC Theory Committee, Davison E.J., ed.*, 1990.

[3] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation $AX + XB = C$," *Comm. ACM*, 1972, vol. 15, no. 9.

[4] S. Boyd and C. Barratt, "Linear controller design: Limits of performance," *Prentice-Hall*, 1990.

[5] C. Ceton, P. M. R. Wortelboer, and O. H. Bosgra, "Frequency-weighted closed-loop balanced reduction," in *Proc. 2nd Eur. Control Conf.*, Groningen, The Netherlands, 1993, pp. 697-701.

[6] J. C. Chen, K. Zhou, and B.-C. Chang, "Closed-Loop Controller Reduction by a Structured Truncation Approach," *Proc. 33rd Conf. Decision Control, Dec. 14-16, Lake Buena Vista, USA*, 1994, pp. 2726-2731.

[7] J. C. Doyle, K. Glover, P. P. Khargonekar and B. A. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Trans. Automat. Control*, vol. 34, no. 8, 1989, pp. 831-847.

[8] D. Enns, "Model reduction for control system design," *Ph.D. dissertation*, Stanford University, Stanford, CA, USA, 1984.

[9] D. Enns, "Model reduction with balanced realization: an error bound and a frequency weighted generalization," in *Proc. 23rd Conf. Decision & Control*, Las Vegas, USA, 1984, pp. 127-132.

[10] K. Glover and J. C. Doyle, "State-space formulae for all stabilizing controllers that satisfy an H_∞ norm bound and relations to risk sensitivity," *Syst. & Contr. Lett.*, vol. 11, 1988, pp. 167-172.

[11] G. H. Golub and C. Van Loan, "Matrix Computations," *John Hopkins University Press*, 1989.

[12] E. A. Jonckheere and L. M. Silverman "A new set of invariants for linear systems - application to reduced order compensator design." *IEEE Trans. Aut. Control*, vol. 28, 1983, pp. 953-964.

[13] Y. Liu, B. D. O. Anderson, and U.-L. Ly, "Coprime Factorization Controller Reduction with Bezout Identity Induced Frequency Weighting," *Automatica*, vol. 26, no. 2, 1990, pp. 233-249.

[14] B. C. Moore, "Principal component analysis in linear systems: controllability, observability, and model reduction," *IEEE Trans. Aut. Control*, vol. 26, no. 1, 1981, pp. 17-31.

[15] M. G. Safonov and R. Y. Chiang, "A Schur method for balanced model reduction," *Amer. Contr. Conf.*, 1988, pp. 1036-1040.

[16] G. W. Stewart, "Algorithm 406: HQR3 and EXCHNG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of a Real Upper Hessenberg Matrix," *ACM Trans. Math. Soft.*, 1976, vol. 2, pp. 275-280.

[17] E. I. Verriest, Low Sensitivity Design and Optimal Order Reduction for the LQG Problem. *Proc. 24th Midwest Symp. Circ. Syst.*, Albuquerque, NM, USA, 1981, pp. 365-369.

[18] C. De Villemagne, and R. E. Skelton, 1988, "Controller Reduction Using Canonical Interactions," *IEEE Trans. Automat. Control*, vol. 33, no. 8, pp. 740-750

[19] Wortelboer, P. M. R., "Frequency-weighted Balanced Reduction of Closed-loop Mechanical Servo-systems: Theory and Tools," *Ph.D. dissertation*, Techn. Univ. Delft, 1994.