A Model Order Reduction Method for Lightly Damped State Space Systems

Matthew Cappello

Follow this and additional works at: https://repository.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.
A Model Order Reduction Method for Lightly Damped State Space Systems

by

Matthew Cappello

A Thesis Submitted

in

Partial Fulfillment

of the

Requirements for the Degree of

MASTER OF SCIENCE

In

Electrical Engineering

PROF.            _______________________
            ______________________________
            (Thesis Advisor’s Name, Printed)

PROF.            _______________________________________________________

PROF.            _______________________________________________________

PROF.            _______________________________________________________
            (Department Head’s Name, Printed)

DEPARTMENT OF ELECTRICAL AND MICROELECTRONIC ENGINEERING

COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

SEPTEMBER 2012
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract and Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1. Literary Review</td>
<td>9</td>
</tr>
<tr>
<td>1.1 Krylov Subspace</td>
<td>9</td>
</tr>
<tr>
<td>1.2 Balanced Reduction</td>
<td>14</td>
</tr>
<tr>
<td>2. Mathematical background</td>
<td>17</td>
</tr>
<tr>
<td>2.1 Approximate orthogonality</td>
<td>17</td>
</tr>
<tr>
<td>2.2 Numeric Grammian Calculations</td>
<td>25</td>
</tr>
<tr>
<td>2.3 Modal Importance Calculations</td>
<td>28</td>
</tr>
<tr>
<td>3. Investigation</td>
<td>32</td>
</tr>
<tr>
<td>3.1 Continuous Time Models</td>
<td>32</td>
</tr>
<tr>
<td>3.2 Discrete Time Models</td>
<td>64</td>
</tr>
<tr>
<td>4. Conclusion</td>
<td>72</td>
</tr>
<tr>
<td>Appendix A Continuous and discrete time</td>
<td>75</td>
</tr>
<tr>
<td>grammian calculation functions</td>
<td></td>
</tr>
<tr>
<td>Appendix B FRF Sweep Reduction Benchmark</td>
<td>76</td>
</tr>
<tr>
<td>Appendix C Continuous Time Diagonal System Reduction</td>
<td>78</td>
</tr>
<tr>
<td>Appendix D Discrete Time Diagonal System Reduction</td>
<td>82</td>
</tr>
</tbody>
</table>
Abstract

This paper introduces a model order reduction method that takes advantage of the near orthogonality of lightly damped modes in a system and the modal separation of diagonalized models to reduce the model order of flexible systems in both continuous and discrete time. The reduction method is computationally fast and cheap, not requiring any singular value decompositions or large matrix operations. Numeric solutions to the infinite time Lyapunov equations are presented, and used to solve for the observability and controllability grammians of diagonalized models. Four different modal importance calculations are produced from the diagonalized model’s grammians and are compared to the Hankel singular values of balanced model order reduction. The frequency response functions (FRF) of the reduced diagonalized models are compared to models reduced using the balanced reduction method. A weighted integral of the FRF error is taken as a metric for judging which reduction method is better for each individual model. For low order or lightly damped higher order systems the diagonal reduction method results in significantly less FRF error than the balanced model order reduction.

Introduction

In control theory many dynamical systems and structures can be modeled by a large order state space model. The continuous time state space model is written by four matrices \( \{A, B, C, D\} \) and the discrete time state space model has the same four matrices plus a sample period\( \{A, B, C, D, T_s\} \). The size of the A matrix is the square of the number of states. The B and C matrices are proportional to the number of states in the modes. Because of this it is often impractical to work with models that have a large number of states for both simulation and controller design purposes. It is often necessary to reduce the order of the state space model. There are various types of model order reduction methods such as the moment matching methods (Krylov-Subspace) and SVD based methods (Balanced Model Order Reduction). These methods will be explained in more detail in the literary review section of this paper.
The purpose of this paper is to introduce a new alternative model order reduction method based on the frequency response function (FRF) near-orthogonality of lightly damped modes of a diagonalized state space model. The diagonalized state space model is used because of its sparse A matrix, and because each pole pair and corresponding sections of the B and C matrices that are used to calculate the FRF are decoupled from each other. The FRF near-orthogonality of the individual lightly-damped modes of a diagonalized model is proven in the mathematical background section. The sparseness of the A matrix is used to advantage in calculating the observability and controllability grammians algebraically. These grammians are then used to compute an importance weighting for each individual mode in the system, so as to determine the order in which the modes should be removed. This reduction method is computationally simple compared to both the Krylov-Subspace and Balanced model order reduction methods because the only diagonalization needed is the initial diagonalization of the state-space model. This makes the reduction method presented here a much quicker and more efficient way of reducing lightly damped models, in addition to producing reduced-order models that better preserve important characteristics of the original high-order model's FRF. The diagonal reduction method also allows easy restriction of frequency ranges over which modes may be removed, or not removed. It also allows for frequency weighting. For example, the lower frequencies can be weighted more importantly if the lower frequencies are to remain untouched. Any mismodeling generally should occur at higher, rather than lower, frequencies because the purpose of the model is to be used as the basis for designing a controller with lower-frequency bandwidth.
**Problem Statement**

The frequency response function (FRF) is useful in determining the forced response of a system to a sinusoidal input. In continuous time it can be obtained from the state space model through by

\[ H(s) = C \times (sI - A)^{-1} \times B + D \] (1)

The closer a model can get to the system’s actual FRF the better the original controller will be at controlling the system. When reducing a system, it is favorable to have the reduced order model’s FRF to be as close to the original model’s FRF as possible. Since controllers usually perform better in the lower frequency ranges and worse in the higher frequency ranges it is optimal to have the reduced model nearly perfect in the lower frequency ranges. The higher frequency range is somewhat less important.

There are two commonly-used approaches to model order reduction for linear systems, moment matching methods, and singular value decomposition based methods. Moment matching methods usually focus on time domain analysis such as the Krylov Subspace method. Singular-value decomposition (SVD) methods of balanced model order reduction focus on distributing the error across the entire frequency range.

The aim of balanced model order reduction, a SVD method, is to minimize the unweighted H-infinity norm of the error [7]. This ensures the maximum error across the whole frequency range is as low as possible [8]. The balanced model order reduction is the most effective one of these techniques, although it has a weakness when it comes to very lightly damped poles. This is due to the fact that the poles are nearly orthogonal and are not coupled together. It is tough to remove sections of the function that evens out the effect of the reduction over all frequencies. Some methods have been proposed to
do balanced model order reduction without having to balance the entire state space system. The main idea behind them is trying to solve the Lyapunov equations indirectly for large state space systems.

One such method uses the power method and matrix-vector multiplications to solve for the dominant eigenvalues of the Lyapunov equations [9]. This method is better suited to sparse coefficient matrices. Such systems may be too large to feasibly do balanced model order reduction, and this method takes advantage of using fewer matrix operations to approximate the solutions to the Lyapunov equations.

This idea is expanded in the Approximate Subspace Iteration method, which extends the method of balanced model order reduction onto larger order systems [10]. Another method, called Proper Orthogonal Decomposition (POD) [11], allows the extraction of a reduced set of basis functions using eigenvalue analysis. The reduced system is a linear combination of the basis functions. This method is less effective than the balanced model order reduction, but can be used for larger order models effectively.

Another method is introduced in [12] to try to hybridize the POD and balanced model order reduction methods. It is less effective than balanced model order reduction but is able to handle larger systems. This method is further explored, and the balanced model order reduction, POD, and the hybrid balanced model order reduction using POD are compared and contrasted [13]. The POD method can also be used for non-linear systems as explored in [14,15].

In the present work, a new method is proposed to reduce lightly damped models effectively, preserving the more important low frequency components of the FRF and affecting the higher frequency values minimally. Unlike [11], this method does not seek to find orthogonal functions with which to decompose the state space system. It does, however, assume that the state space system is very lightly damped, and therefore the diagonalized form of the state space model should form an early orthogonal basis. Because the system of very lightly damped poles is nearly orthogonal when in diagonal form, a set of
normal basis functions need not be computed. To reduce the system, the mode with the smallest overall effect on the FRF can be removed, and still keep the original system's and reduced system’s FRF nearly identical.

The metric used to determine effectiveness of the reduction method is a frequency weighted error measurement integrated over all observed frequencies. Frequency weighting is done indirectly, by computing the FRF with log-scaled frequency points. Thus, in the error computations, lower frequencies are equalized in importance with higher frequencies, by virtue of their higher density (i.e., more terms) in the summation.

Additional benefits are that this method is a very quick method and computationally easy. No singular value decompositions, Cholesky decompositions, or other large-scale time-consuming matrix operations are needed. Aside from saving a lot of computational time, it also saves on memory-usage. This is a specified model order reduction method specializing in minimizing models of flexible structures. No similar minimization method has been found in the extant literature. It seems that this has not been developed before because most model order reduction techniques focus on robustness over a large variety of models. This is not a general-purpose method to be used for reducing many diverse models. Rather, it specifically targets very lightly damped systems, and is very effective in reducing, for example, models of flexible structures. Large order flexible structures are often found in the field of aerospace engineering. A model order reduction capable of reducing very high order models of flexible structures with almost no impact on model accuracy at lower frequencies will be very beneficial in this field.
### Description of Diagonal Model Order Reduction

<table>
<thead>
<tr>
<th>Step</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Diagonalize the state space system</td>
</tr>
<tr>
<td>2</td>
<td>Calculate the controllability and observability grammians</td>
</tr>
<tr>
<td>3</td>
<td>Calculate the importance values from the grammians</td>
</tr>
<tr>
<td>4</td>
<td>Rearrange the diagonal system matrices in descending order of importance values</td>
</tr>
<tr>
<td>5</td>
<td>Truncate the system matrices</td>
</tr>
<tr>
<td>6</td>
<td>Add in the dc gain lost from the removal of the poles of the system into the D matrix</td>
</tr>
</tbody>
</table>

1. In order to perform the diagonal system reduction first the system must be transformed into diagonal form.

2. The observability and controllability grammians can be calculated algebraically by using the equations 2.2.3 and 2.2.4 for continuous time systems and equations 2.2.7 and 2.2.8 for discrete time systems.

3. The importance values calculations are shown in equations 2.3.1-2.3.4. The importance calculation that introduced the least amount of FRF error is in equation 2.3.1, the diagonal singular value.

4. The diagonal system matrices should be rearranged from the largest importance value to the smallest importance value. For example if the highest importance value corresponds to the pole located in row and column five of the original diagonal system A matrix it should be moved to row and column one of the rearranged A matrix. The fifth row of the B matrix should be moved into the first row of the rearranged B matrix and the fifth column of the C matrix should be moved into the first row of the rearranged C matrix.
5. Once the matrix is rearranged in descending importance the matrix can be truncated by the desired number of poles. If ten poles are to be removed the last ten rows and columns of the A matrix should be removed, the last ten rows of the B matrix should be removed and the last ten columns of the C matrix should be removed.

6. The removal of the modes of the system introduces a loss of dc gain into the system. This must be fixed by adding the loss into the system D matrix. The loss of the dc gain can be seen in the frequency response function (FRF) when $\omega = 0$ for the continuous time case. For each pole removed the new D matrix must be

$$\tilde{D} = D + \frac{\tilde{c} \cdot \tilde{b}}{\lambda}$$

For the discrete time case it is when $\omega = 1$ so

$$\tilde{D} = D + \frac{\tilde{c} \cdot \tilde{b}}{1 - \lambda}$$
1. Literary Review

The following section reviews two existing model order reduction methods. The first method that is reviewed in section 1.1 is the Krylov-Subspace model order reduction. This is an example of a moment matching model order reduction method. The second and final method reviewed in section 1.2 is the Balanced model order reduction. This is an example of a singular value decomposition model order reduction method.

1.1 Krylov-Subspaces

Another commonly-used large-scale model-order reduction method is the Krylov-Subspace method. It is often used in modeling for steady-state and transient (time-domain) analysis.

For mathematical simplicity, in this section this method is only explored for SISO models having no feed through matrix $D$. The transfer function for a SISO state-space system is written as

$$H(s) = c \ast [(s \ast I - A)^{-1} \ast b], \quad 1.1.1$$

Where $I$ is the appropriate-size identity matrix. This transfer function is guaranteed to be a strictly proper rational function. To compute the expansion of the transfer function 1.1.1 around the point $s_0$ let

$$A_{s_0} = -(s_0 \ast I - A)^{-1}$$
$$b_{s_0} = (s_0 \ast I - A)^{-1} \ast b$$

$$H(s) = c \ast [(s_0 \ast I - A) + (s - s_0)I]^{-1} \ast b = c \ast [I - (s - s_0)A_{s_0}]^{-1} \ast b_{s_0} \quad 1.1.2$$

The Taylor series expansion of 1.1.2 about point $s_0$ is

$$H(s) = c \ast b_{s_0} + c \ast A_{s_0} \ast b_{s_0} (s - s_0) + c \ast A_{s_0}^2 \ast b_{s_0} (s - s_0)^2 + ...$$
\[ = m_0 + m_1(s - s_0) + m_2(s - s_0)^2 + \cdots \]

Where \( m_k = c \times A_{s_0}^{-k} \times b_{s_0} \) and each \( m_k \) is said to be a moment about point \( s_0 \). A function \( H_p(s) \) is said to be a \( p \)\(^{th} \) Padé approximation of \( H(s) \) about the point \( s_0 \) if it matches with the moments of \( H(s) \) as far as possible [4].

\[
H_p(s) = \frac{a_{p-1}s^{p-1} + \cdots + a_1s + a_0}{b_ps^p + b_{n-1}s^{n-1} + \cdots + b_1s + 1}
\]

The coefficients of the denominator can be found by solving the following matrix equation.

\[
\begin{bmatrix}
m_0 & m_1 & \cdots & m_{p-1} \\
m_1 & m_2 & \cdots & m_p \\
\vdots & \vdots & \ddots & \vdots \\
m_{p-1} & m_p & \cdots & m_{2p-2}
\end{bmatrix}
\begin{bmatrix}
b_n \\
b_{n-1} \\
\vdots \\
b_1
\end{bmatrix}
= 
\begin{bmatrix}
m_p \\
m_{p-1} \\
\vdots \\
m_{2p-1}
\end{bmatrix}
\]

The Hankel matrix is known as the moment matrix \( M_p \). After solving for the \( b_i \), the coefficients of the numerator can be found by solving the following matrix equation

\[
\begin{bmatrix}
a_0 \\
a_1 \\
\vdots \\
a_{p-1}
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
b_{p-1} & b_{p-2} & \cdots & 1
\end{bmatrix}
\begin{bmatrix}
m_0 \\
m_1 \\
\vdots \\
m_{p-1}
\end{bmatrix}
\]

The moment matrix is quite often a very ill-conditioned matrix. That makes it so this method is only effective for a very small frequency range around \( s_0 \). Fortunately, this can be remedied by the connection between the Padé approximation and the Lanczos process.

The known model is projected onto Krylov subspaces, which are subspaces created by the \( A_{s_0} \) matrix and the \( b_{s_0} \) vector. If \( A_{s_0} \) is a non-symmetric matrix then the vector \( c \) must be used also.

The right Krylov subspace is generated by
\[ K_p(A_{s_0}, b_{s_0}) = \text{span}\{b_{s_0}, A_{s_0} \ast b_{s_0}, A_{s_0}^2 \ast b_{s_0}, \ldots, A_{s_0}^{p-1} \ast b_{s_0}\} \]

The left Krylov subspace is generated by

\[ K_p(A_{s_0}^T, c) = \text{span}\{c, A_{s_0}^T \ast c, A_{s_0}^{T2} \ast c, \ldots, A_{s_0}^{Tp-1} \ast c\} \]

These bases contain all of the proper moments of the function but they are unsatisfactory bases. The solution to this problem is to do a basis-transformation to a more suitable set of basis vectors using the Lanczos process. These vectors are known as the Lanczos vectors.

\[ K_p(A, r) = \text{span}\{v_1, v_2, v_3, \ldots, v_p\} \]

The left Krylov subspace is generated by

\[ K_p(A^T, l) = \text{span}\{w_1, w_2, w_3, \ldots, w_p\} \]

The Lanczos process creates these vectors in such a way that they are biorthogonal

\[ w^T_j \ast v_k = 0 \text{ for all } j \neq k \]

The Lanczos vectors can be calculated by using the following two recurrence equations

\[ A_{s_0} \ast V_p = V_p \ast T_p + \rho_{p+1} \ast v_{p+1} \ast e_p^T \]

\[ A_{s_0}^T \ast W_p = W_p \ast \tilde{T}_p + \eta_{p+1} \ast w_{p+1} \ast e_p^T \]

Where \( T_p \) and \( \tilde{T}_p \) are the following tridiagonal matrices.
The two matrices are related by the diagonal similarity transformation

\[ T_p = \begin{bmatrix}
\alpha_1 & \beta_2 \\
\rho_2 & \alpha_2 & \ddots \\
& \ddots & \ddots & \ddots \\
& & \ddots & \ddots & \beta_p \\
& & & \rho_p & \alpha_p
\end{bmatrix} \]

\[ T_p^* = \begin{bmatrix}
\alpha_1 & \gamma_2 \\
\eta_2 & \alpha_2 & \ddots \\
& \ddots & \ddots & \ddots \\
& & \ddots & \ddots & \gamma_p \\
& & & \eta_p & \alpha_p
\end{bmatrix} \]

The projection of \( A_{s_0} \) onto \( K_p(A,r) \) is represented by

\[ W_p^T A_{s_0} V_p = D_p T_p \]

If the process is carried all the way to the \( M^{th} \) step then the process is the same as tridiagonalizing \( A_{s_0} \) by a similarity transformation,

\[ V_M^{-1} A_{s_0} V_M = T_M. \]

\( T_p \) is the leading \( p \times p \) principal submatrix of the tridiagonal matrix \( T_M \)

The following is the basic algorithm for performing the Lanczos process to reduce a system to \( p^{th} \) order

1: \( \rho_1 = \|b_0\|_2 \)
2: \( \eta_1 = \|c\|_2 \)
3: \( v_1 = \frac{b_{s_0}}{\rho_1} \)
4: \( w_1 = \frac{c}{\eta_1} \)
5: for \( k = 1:p \)
6: \( \delta_k = w_k^T v_k \)
7: \( \alpha_k = \frac{w_k^T A_{s_0} v_k}{\delta_k} \)
8: \( \beta_k = \frac{\delta_k}{\delta_{k-1}} \cdot \eta_k \)
9: \( \gamma_k = \frac{\delta_k}{\delta_{k-1}} \cdot \rho_k \)
10: \( v = A_{s_0} v_k - v_k \cdot \alpha_k - v_{k-1} \cdot \beta_k \)
11: \( w = A_{s_0}^T w_k - w_k \cdot \alpha_k - w_{k-1} \cdot \gamma_k \)
12: \( \rho_{k+1} = \|v\|_2 \)
Now the transfer function of the original system can be written as

\[ H(s) = c \cdot b_{20} \cdot \frac{\text{det} (I - (s - s_0)) \cdot T'_M}{\text{det} (I - (s - s_0)) \cdot T_M} \]

where \( T'_M \) is obtained by deleting the first row and column of \( T_M \).

The following is an example of the Krylov subspace used to reduce an 834th order model to a 45th order model. The model comes from references [5] and the example can be found in reference [4]. The expansion point \( s_0 \) is chosen to be 0.

As can be seen in figure 1.1.1(a) the original system has about 22 very strong resonant peaks and could be approximated very well by 40th to a 50th order model. The original system is reduced to a 45th order.

This reduction yields almost no error at low frequencies and nearly matches the system at higher frequencies.
1.2 Balanced Model Order Reduction

One of the most commonly used model-order reduction methods is the balanced model order reduction, which is derived in this section. This model order reduction method is used to minimize the unweighted H-infinity norm of the error [7]. This method spreads out the error in model order reduction nearly evenly over all frequencies.

A continuous state space system with \( m \) inputs, \( p \) outputs, and \( n \) states is generally represented by the following state space equations

\[
\dot{x} = A \ast x + B \ast u
\]

\[
y = C \ast x + D \ast u
\]

Where \( x \) and \( \dot{x} \) are \( n \)-by-1 vectors, \( A \) is an \( n \)-by-\( n \) matrix, \( B \) is an \( n \)-by-\( m \) matrix, \( u \) is an \( m \)-by-1 vector, \( y \) is a \( p \)-by-1 vector, \( C \) is a \( p \)-by-\( n \) matrix, and \( D \) is a \( p \)-by-\( m \) matrix. The model-order equals the number of system states \( n \), which is also the number of elements in state-vector \( x \).

To perform a balanced-model order reduction, first the state-space model must be transformed into internally balanced form. A system is said to be internally balanced when its observability grammian \( \bar{Q} \) is equal to its controllability grammian \( \bar{P} \).

The infinite-time controllability grammian \( \bar{P} \) can be found by solving the Lyapunov equation

\[
A \ast \bar{P} + \bar{P} \ast A^H + B \ast B^H = 0_{n \times n}.
\]

Similarly the infinite-time observability grammian \( \bar{Q} \) can be found by solving

\[
A^H \ast \bar{Q} + \bar{Q} \ast A + C^H \ast C = 0_{n \times n}.
\]
To transform the system into internally balanced form, a transformation matrix $T$ is required such that the following relation is satisfied,

$$T^{-1} \ast \bar{P} \ast (T^{-1})^H = T^H \ast \bar{Q} \ast T,$$

or

$$\bar{P} = T \ast T^H \ast \bar{Q} \ast T \ast T^H.$$

The Cholesky decomposition is performed on the controllability grammian, yielding an upper triangular matrix $R$, such that

$$R^H \ast R = \bar{Q}.$$ 

So

$$\bar{P} = T \ast T^H \ast R^H \ast R \ast T \ast T^H.$$

Pre-multiplying by $R$ and post-multiplying by $R^H$ gives

$$R \ast \bar{P} \ast R^H = R \ast T \ast T^H \ast R^H \ast R \ast T \ast T^H \ast R^H.$$

Taking the singular value decomposition of the left hand side of the equation and factoring the right side of the equation,

$$U \ast \Sigma \ast U^H = (R \ast T \ast T^H \ast R^H)^2.$$

Taking the square root of both sides,

$$U \ast \Sigma^{\frac{1}{2}} \ast (U \ast \Sigma^{\frac{1}{2}})^H = R \ast T \ast (R \ast T)^H.$$

Therefore,

$$U \ast \Sigma^{\frac{1}{2}} = R \ast T.$$

So the transformation matrix that balances the system is

$$T = R^{-1} \ast U \ast \Sigma^{\frac{1}{2}},$$

where $R$ is obtained by the Cholesky Decomposition of $\bar{Q}$, and $U$ and $\Sigma$ are obtained by the singular value decomposition of $R \ast \bar{P} \ast R^H$. 

16
The Hankel singular values can be obtained from the diagonal of either of the balanced-model grammians, which are equal diagonal-matrices. A reduced-model $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ can be obtained by truncating the rows and columns corresponding to those Hankel singular values that are below some specific threshold. If $k$ states are to be preserved, the balanced system matrices should be truncated as indicated in the below partitioning. The reduced model is often transformed back into a more structured form, such as modal-canonic.

$$A = \begin{bmatrix}
A_{11} & A_{12} & \ldots & A_{1k} & \ldots & A_{1n} \\
A_{21} & A_{22} & \ldots & A_{2k} & \ldots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
A_{k1} & A_{k2} & \ldots & A_{kk} & \ldots & A_{kn} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
A_{n1} & A_{n2} & \ldots & A_{nk} & \ldots & A_{nn}
\end{bmatrix}
$$

$$B = \begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_k \\
\vdots \\
B_n
\end{bmatrix}
$$

$$C = \begin{bmatrix}
C_1 & C_2 & \ldots & C_k & \ldots & C_n
\end{bmatrix}
$$

More on the balanced model order reduction can be found in sources [1] and [2].
2. Mathematical Background

This section covers important mathematical concepts pertaining to the diagonal model order reduction method. Section 2.1 proves the concept of approximate orthogonality of lightly damped resonant modes in a diagonal state space model. Section 2.2 derives the algebraic calculations for solving for the infinite time observability and controllability grammians in both continuous and discrete time. Section 2.3 shows how the importance values (similar to Hankel Singular Values) are calculated.

2.1 FRF approximate orthogonality of lightly damped modes

This section explains the concept of "approximate orthogonality" of the FRF's of resonant modes when at least one of them is very lightly damped.

![Figure 2.1.1 Single mode with damping factor 0.1](image)
As a resonant mode in a system is made more lightly-damped, it becomes more orthogonal to each of the other modes, in terms of their frequency-response functions (FRF's). For a more heavily damped mode, the FRF is significantly nonzero over a wider range of frequencies. In figure 2.1.1 the FRF of the mode with the damping factor of 0.1 is significantly nonzero for more than three orders-of-magnitude of...
frequency. As a mode becomes more lightly damped, its FRF becomes more impulse-like. In figure 2.1.2 when the damping factor is ten times lighter, at 0.01, the FRF is nonzero for less than two orders of magnitude. When the damping factor is ten times lighter again, at 0.001, the FRF is almost impulse like. It is only significantly nonzero for one tenth of an order-of-magnitude, as shown in figure 2.1.3.

Figure 2.1.4 Multiple modes with damping factor 0.1
As the modes get more and more lightly damped they become more nearly orthogonal. There is a lot of overlap of the two functions in Fig. 2.1.4. There is less overlap between the functions in Fig. 2.1.5. There appears to be no overlap at all in Fig. 2.1.6. This means the functions appear to be orthogonal. A
A functional inner product can be used to show the “angle” between two functions mathematically. The definition of functional inner product can be found in equation 2.2.1

\[ \langle H_1(j\omega), H_2(j\omega) \rangle = \int_{-\infty}^{\infty} H_1(j\omega) \cdot H_2^*(j\omega) \]  

The normalized inner product value between two functions can be calculated as

\[ \frac{\langle H_1(j\omega), H_2(j\omega) \rangle}{\|H_1(j\omega)\| \cdot \|H_2(j\omega)\|} \]

Where \(\|H(j\omega)\|\) is the square root of the inner product of \(H(j\omega)\) with itself (the norm). The angle between the two functions can be calculated as

\[ \theta = \cos^{-1} \left( \frac{\langle H_1(j\omega), H_2(j\omega) \rangle}{\|H_1(j\omega)\| \cdot \|H_2(j\omega)\|} \right) \]

Table 2.1.1 Orthogonality calculations 7/10 of a decade apart

<table>
<thead>
<tr>
<th>(\zeta_1)</th>
<th>(\zeta_2)</th>
<th>Normalized inner product</th>
<th>(\theta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>3.08x10^{-2}</td>
<td>88.235</td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
<td>3.23x10^{-3}</td>
<td>89.815</td>
</tr>
<tr>
<td>0.001</td>
<td>0.001</td>
<td>3.24x10^{-4}</td>
<td>89.981</td>
</tr>
<tr>
<td>0.0001</td>
<td>0.0001</td>
<td>3.24x10^{-5}</td>
<td>89.998</td>
</tr>
<tr>
<td>0.1</td>
<td>0.01</td>
<td>1.03x10^{-2}</td>
<td>89.410</td>
</tr>
<tr>
<td>0.1</td>
<td>0.001</td>
<td>3.27x10^{-3}</td>
<td>89.813</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0001</td>
<td>1.03x10^{-3}</td>
<td>89.941</td>
</tr>
</tbody>
</table>

These measurements are for the modes positioned as shown in the above figures, centered at 7/10 of a decade apart from one another. When both modes damping factors are reduced by an order of magnitude the normalized inner product also changes by roughly an order of magnitude. When one mode is held at constant damping and the second mode’s damping factor is lightened by an order of magnitude the normalized inner product changes by roughly the square root of 10.
As seen in Table 2.1.2 when the modes are moved closer together the more heavily damped modes become more coupled. When there are many modes clustered together they must be very lightly damped or else the modes do not exhibit this near-orthogonality to one another.
Damping factors below 0.001 are quite common in flexible structures, and their near-orthogonality implies that when such lightly-damped resonances are nearly uncontrollable and unobservable, they can be excised from the model with minimal effect on the FRF of the remaining model.

The mathematical contribution of a single pole-pair \(-\alpha_1 \pm j\omega_1\) to the FRF is of the form \(\frac{k_1(j\omega)}{\omega_1^2+(j\omega+\alpha_1)^2}\). If the pole is lightly enough damped \(|\omega_1| \gg |\alpha_1|\) and the FRF contribution can be approximated as

\[
\frac{k_1(j\omega)}{\omega_1^2+(j\omega)^2} = \frac{k_1(j\omega)}{\omega_1^2 - \omega^2}.
\]

Taking the inner product as in equation 2.1.1 of two such contributing functions can prove orthogonality of the two functions.

\[
\int_{-\infty}^{\infty} \frac{k_1(j\omega)}{(\omega_1^2 - \omega^2)} \frac{k_2(j\omega)}{(\omega_2^2 - \omega^2)} \, dw = -2k_1k_2 \int_{0}^{\infty} \frac{\omega^2}{(\omega^2 - \omega_1^2)(\omega^2 - \omega_2^2)} \, dw
\]

Using partial fraction expansion

\[
-2k_1k_2 \int_{0}^{\infty} \left( \frac{\omega_1^2}{(\omega^2 + \omega_1^2)} + \frac{\omega_2^2}{(\omega^2 + \omega_1^2)} \right) \, dw
\]

\[
= \frac{-2k_1k_2}{\omega_1^2 + \omega_2^2} \left[ \int_{0}^{\infty} \frac{\omega_1^2}{\omega^2 - \omega_1^2} \, dw - \int_{0}^{\infty} \frac{\omega_2^2}{\omega^2 - \omega_2^2} \, dw \right]
\]

Setting \(-2k_1k_2w_1^2\) = \(c_1\) and \(-2k_1k_2w_2^2\) = \(c_2\) for simplicity, and splitting the integrals where their denominators equal zero,

\[
c_1 \left[ \int_{0}^{w_1} \frac{1}{\omega^2 - \omega_1^2} \, dw + \int_{w_1}^{\infty} \frac{1}{\omega^2 - \omega_1^2} \, dw \right] + c_2 \left[ \int_{0}^{w_2} \frac{1}{\omega^2 - \omega_2^2} \, dw + \int_{w_2}^{\infty} \frac{1}{\omega^2 - \omega_2^2} \, dw \right].
\]
From the integral table in [6]

\[ \int \frac{1}{a^2 - x^2} = \frac{1}{2a} \ln \left| \frac{a - x}{x + a} \right| \]

So the integral is

\[ \frac{c_1}{2w_1} [\ln(0) - \ln(1) + \ln(1) - \ln(0)] + \frac{c_2}{2w_2} [\ln(0) - \ln(1) + \ln(1) - \ln(0)] = 0 \]

Because the inner product of these two functions is zero, the functions are orthogonal.
2.2 Observability and Controllability grammian calculations

The continuous time controllability ($\bar{P}$) and observability grammians ($\bar{Q}$) satisfy the following Lyapunov equations.

\[
\begin{align*}
\Lambda * \bar{P} + \bar{P} * \Lambda^H + B * B^H &= 0_{n \times n} \quad 2.2.1 \\
\Lambda^H * \bar{Q} + \bar{Q} * A + C^H * C &= 0_{n \times n} \quad 2.2.2
\end{align*}
\]

Written out the observability grammian Lyapunov equation can be simplified to.

\[
\begin{bmatrix}
\lambda_1 P_{11} & \lambda_1 P_{12} & \cdots & \lambda_1 P_{1n} \\
\lambda_2 P_{21} & \lambda_2 P_{22} & \cdots & \lambda_2 P_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_n P_{n1} & \lambda_n P_{n2} & \cdots & \lambda_n P_{nn}
\end{bmatrix} + \begin{bmatrix}
\bar{\lambda}_1 P_{11} & \bar{\lambda}_2 P_{12} & \cdots & \bar{\lambda}_n P_{1n} \\
\bar{\lambda}_1 P_{21} & \bar{\lambda}_2 P_{22} & \cdots & \bar{\lambda}_n P_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\bar{\lambda}_1 P_{n1} & \bar{\lambda}_2 P_{n2} & \cdots & \bar{\lambda}_n P_{nn}
\end{bmatrix} = - B * B^H
\]

And further simplified to

\[
\begin{bmatrix}
(\lambda_1 + \bar{\lambda}_1) P_{11} & (\lambda_2 + \bar{\lambda}_2) P_{12} & \cdots & (\lambda_1 + \bar{\lambda}_n) P_{1n} \\
(\lambda_2 + \bar{\lambda}_1) P_{21} & (\lambda_2 + \bar{\lambda}_2) P_{22} & \cdots & (\lambda_2 + \bar{\lambda}_n) P_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
(\lambda_n + \bar{\lambda}_1) P_{n1} & (\lambda_n + \bar{\lambda}_2) P_{n2} & \cdots & (\lambda_n + \bar{\lambda}_n) P_{nn}
\end{bmatrix} = \begin{bmatrix}
B_1 * B_1^H & B_1 * B_2^H & \cdots & B_1 * B_n^H \\
B_2 * B_1^H & B_2 * B_2^H & \cdots & B_2 * B_n^H \\
\vdots & \vdots & \ddots & \vdots \\
B_n * B_1^H & B_n * B_2^H & \cdots & B_n * B_n^H
\end{bmatrix}
\]

The grammian can be solved for element by element.

\[
\begin{bmatrix}
P_{11} & P_{12} & \cdots & P_{1n} \\
P_{21} & P_{22} & \cdots & P_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
P_{n1} & P_{n2} & \cdots & P_{nn}
\end{bmatrix} = \begin{bmatrix}
B_1 * B_1^H & B_1 * B_2^H & \cdots & B_1 * B_n^H \\
B_2 * B_1^H & B_2 * B_2^H & \cdots & B_2 * B_n^H \\
\vdots & \vdots & \ddots & \vdots \\
B_n * B_1^H & B_n * B_2^H & \cdots & B_n * B_n^H
\end{bmatrix}
\]

Similarly the Controllability grammian can be determined by
The discrete time controllability ($\bar{P}$) and observability grammians ($\bar{Q}$) satisfy the following Lyapunov equations.

\[
\Lambda \ast \bar{P} \ast \Lambda^H - \bar{P} + B \ast B^H = 0_{nxn} \tag{2.2.5}
\]

\[
\Lambda^H \ast \bar{Q} \ast \Lambda - \bar{Q} + C^H \ast C = 0_{nxn} \tag{2.2.6}
\]

Written out the observability grammian Lyapunov equation can be simplified to.

\[
\begin{bmatrix}
    P_{11} & P_{12} & \ldots & P_{1n} \\
    P_{21} & P_{22} & \ldots & P_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    P_{n1} & P_{n2} & \ldots & P_{nn}
\end{bmatrix}
- \begin{bmatrix}
    \lambda_1 \ast \bar{\lambda}_1 P_{11} & \lambda_1 \ast \bar{\lambda}_2 P_{12} & \ldots & \lambda_1 \ast \bar{\lambda}_n P_{1n} \\
    \lambda_2 \ast \bar{\lambda}_1 P_{21} & \lambda_2 \ast \bar{\lambda}_2 P_{22} & \ldots & \lambda_2 \ast \bar{\lambda}_n P_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    \lambda_n \ast \bar{\lambda}_1 P_{n1} & \lambda_n \ast \bar{\lambda}_2 P_{n2} & \ldots & \lambda_n \ast \bar{\lambda}_n P_{nn}
\end{bmatrix} = B \ast B^H
\]

And further simplified to

\[
\begin{bmatrix}
    (1 - \lambda_1 \ast \bar{\lambda}_1)P_{11} & (1 - \lambda_1 \ast \bar{\lambda}_2)P_{12} & \ldots & (1 - \lambda_1 \ast \bar{\lambda}_n)P_{1n} \\
    (1 - \lambda_2 \ast \bar{\lambda}_1)P_{21} & (1 - \lambda_2 \ast \bar{\lambda}_2)P_{22} & \ldots & (1 - \lambda_2 \ast \bar{\lambda}_n)P_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    (1 - \lambda_n \ast \bar{\lambda}_1)P_{n1} & (1 - \lambda_n \ast \bar{\lambda}_2)P_{n2} & \ldots & (1 - \lambda_n \ast \bar{\lambda}_n)P_{nn}
\end{bmatrix}
= \begin{bmatrix}
    B_1 \ast B_1^H & B_1 \ast B_2^H & \ldots & B_1 \ast B_n^H \\
    B_2 \ast B_1^H & B_2 \ast B_2^H & \ldots & B_2 \ast B_n^H \\
    \vdots & \vdots & \ddots & \vdots \\
    B_n \ast B_1^H & B_n \ast B_2^H & \ldots & B_n \ast B_n^H
\end{bmatrix}
\]

So the grammian can be found element by element.
Similarly the Controllability grammian can be solved for

\[
\begin{pmatrix}
    Q_{11} & Q_{12} & \cdots & Q_{1n} \\
    Q_{21} & Q_{22} & \cdots & Q_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    Q_{n1} & Q_{n2} & \cdots & Q_{nn}
\end{pmatrix}
= \begin{pmatrix}
    C_1^H \ast C_1 & C_1^H \ast C_2 & \ldots & C_1^H \ast C_n \\
    (1 - \bar{\lambda}_1 \ast \lambda_1) & (1 - \bar{\lambda}_1 \ast \lambda_2) & \ldots & (1 - \bar{\lambda}_1 \ast \lambda_n) \\
    C_2^H \ast C_1 & C_2^H \ast C_2 & \ldots & C_2^H \ast C_n \\
    (1 - \bar{\lambda}_2 \ast \lambda_1) & (1 - \bar{\lambda}_2 \ast \lambda_2) & \ldots & (1 - \bar{\lambda}_2 \ast \lambda_n) \\
    \vdots & \vdots & \ddots & \vdots \\
    C_n^H \ast C_1 & C_n^H \ast C_2 & \ldots & C_n^H \ast C_n \\
    (1 - \bar{\lambda}_n \ast \lambda_1) & (1 - \bar{\lambda}_n \ast \lambda_2) & \ldots & (1 - \bar{\lambda}_n \ast \lambda_n)
\end{pmatrix}
\]
2.3 Importance value approximations

The balanced grammian Hankel singular values were compared to the diagonal elements of the grammians of an unbalanced second-order SISO system, as a function of pole-damping. The pole magnitude is fixed and the damping factor is swept from very lightly to heavily damped.

![Figure 2.3.1 Grammian Values vs. damping factor in a second-order SISO system](image)

In Fig 2.3.1 Pb(1,1) is the balanced systems (1,1) element in both the controllability and observability grammian. This is the equivalent to the largest Hankel singular value in the system. Pb(2,2) is the (2,2) element in the observability and controllability grammian, or the second largest Hankel singular value in the system. Q and P denote controllability and observability grammians of the diagonal system respectively. This same notation is used in figures 2.3.2-4. In the second order system the Hankel singular values are extremely close to the geometric mean of the diagonal observability and controllability grammians, when the damping factor is less than 0.04.
Next, the balanced grammian Hankel singular values were compared to the diagonal elements of the grammians of an unbalanced fourth-order SISO system. One pole-pair magnitude was fixed and its damping factor was swept from very lightly damped to heavily damped, while the second pole-pair was held constant. The results of this sweep can be found in Fig 2.3.2 below.

![Figure 2.3.2 Hankel singular values vs. damping](image)

Since the Hankel singular values are in descending order, in the figure the poles appear to switch halfway through the damping range. However, the singular values geometric mean remains very close to the geometric mean of the grammians’ diagonal elements.
The Hankel singular values are extremely close to the geometric mean for the fourth-order MIMO case, as well. The Hankel singular values can be accurately estimated for smaller systems by calculating the
geometric mean of the diagonal elements of the controllability and observability grammians of the unbalanced diagonalized model with lightly-damped poles.

Four different methods of estimating the Hankel singular values were considered. The first method for calculating the diagonalized system's Hankel singular values is the diagonal importance value. This method takes the least amount of computational effort, as it is simply the geometric means of the diagonals of the observability and controllability grammians, and these are easy to compute due to the diagonalized A-matrix.

\[ DIV(k) = \sqrt{Q(k,k) \cdot P(k,k)} \]  \hspace{1cm} 2.3.1

For the three other methods that were investigated, more computational effort is required, because more elements (entire rows, instead of diagonal only) of the two grammians must be computed. Computational requirements are still minimized by the fact that the A-matrix is diagonal.

The second method is the inner product importance value. This method takes the inner product of the rows of the observability and columns of the controllability grammians.

\[ IPIV(k) = \sqrt{Q(k,:) \cdot P(:,k)} \]  \hspace{1cm} 2.3.2

The third method is the row sum importance value. This method takes the geometric mean of the sum of the elements in the rows of the observability and columns of the controllability grammians.

\[ RSIV(k) = \sqrt{\sum(Q(k,:)) \cdot \sum(P(:,k))} \]  \hspace{1cm} 2.3.3

The fourth method is the row magnitude importance value. This method takes the geometric mean of the vector magnitudes of the rows of the observability and columns of the controllability grammians.

\[ RMIV(k) = \sqrt{\|Q(k,:)\| \|P(:,k)\|} \]  \hspace{1cm} 2.3.3
3. **Investigation**

This section compares the reduction of pseudo random state space models using the Diagonal model order reduction and the balanced model order reduction. Section 3.1 looks at continuous time state space models. Section 2.3 looks at discrete time state space models.

### 3.1 Continuous-time systems

The main goal of this research is to find a computationally simple way to obtain a good reduced-order model, based on the modal-canonic form of the state-space model. The need for this arises from the difficulty of applying the usual model-order reduction process to extremely high-order models. In the usual process, computation time and memory requirements increase proportional to (at least) the square of the number of states. The goal of this work was to find a method where that increase is **linearly** proportional to the number of states.

The modal-canonic form of a state-space model has, in essence, the most nearly diagonal A-matrix that is completely real-valued. While the *diagonalized* state-space model has completely decoupled individual states, the modal-canonic A-matrix has completely decoupled individual resonant-modes, *i.e.*, complex-conjugate pole-pairs. The relation between these two forms is very simple, and transforming the A-matrix from diagonal form to modal-canonic form is quite simple, computationally, and it is always possible to do. For that reason, we assume here that these two forms are interchangeable.

Balanced-model order reduction works well for some systems, but it has drawbacks. To use this algorithm, it is first necessary to transform a state-space model into balanced form, requiring a Cholesky decomposition, a singular value decomposition, and several matrix multiplications. For very high-order systems these operations are computationally time-consuming and can be numerically inaccurate. Balanced-model order reduction also typically moves the poles and the transmission zeros of the
original system to different locations in the reduced-order model. One of the most difficult aspects of computations involving the balanced-model is that the A-matrix is full, i.e., all the elements of the A-matrix are typically nonzero. The number of nonzero elements in the balanced-model A-matrix is generally proportional to the square of the number of states. By contrast, the modal-canonic form A-matrix is relatively sparse, with the number of nonzero elements being linearly proportional to the number of states.

Balanced-model order reduction uses the Hankel singular values to determine the overall importance of the model's states. These singular values are the diagonal elements of the balanced-model controllability or observability grammians, which are equal. The least important states are removed, judging the states' importance simply by the magnitude of the associated Hankel singular value. The disadvantage is that balanced-model states are not simply related to modal-canonic states, which are very simply related to the A-matrix eigenvalues (complex-conjugate pole pairs), the modal resonances.

The modal-canonic state-space model has fully decoupled resonant modes. Now, because all of the modes of a lightly-damped system are nearly orthogonal to one another in an FRF sense, it seems that removing the resonant modes (pole-pairs) that have the least observability and controllability could result in a better reduced-order model than the balanced reduction. The question is how to determine the modal observability and controllability without leaving the modal-canonic form (or the essentially equivalent diagonal form).

The most fundamental observation, indeed the starting point of this research, is that the Lyapunov equations for the observability and controllability grammians can be solved fairly simply when the state-space model is diagonalized. Estimates of the Hankel singular values can then be obtained from the controllability and observability grammians. But, because these two grammians are not diagonal, the
Hankel singular values are not necessarily easy to estimate from them. This paper explores several different methods of estimating the Hankel singular values from these diagonalized-model grammians.

To perform a continuous-time diagonalized-model order reduction, a model must be transformed into diagonalized form. This will always be possible if there are no repeated poles. Often, in models of flexible structures, there are no multiple-integrators in the model, so it is reasonable to assume there are no repeated poles:

\[ \{A, B, C, D\} \Rightarrow \{\bar{A}, \bar{B}, \bar{C}, D\} \]

Actually, we make this assumption without any loss of generality. Assuming there are multiple integrators, the modal-canonic form is still tri-diagonal, and the Jordan blocks associated with the multiple integrators are fully decoupled from the block of resonant modes. Therefore, the submodel associated with the resonant modes can be diagonalized the same way as it would without the presence of the multiple integrators.

From the diagonal form the observability and controllability grammians are calculated. Because of the diagonal form, the two Lyapunov equations are easily solved by Eq. 2.2.3 and 2.2.4, instead of having to call more complicated Lyapunov solver-functions. From the diagonalized system grammians, the Hankel singular values are estimated for each mode of the system using the various calculations of Eq. 2.3.1-2.3.4. The states of the system, expressed in matrices A, B, and C, are rearranged by order of decreasing observability and controllability. The system matrices are then truncated to the desired order.

To avoid a change in the dc-gain due to truncating some of the modes, the model is compensated by modifying the D matrix. This is equivalent to modifying the second-order transfer function of each resonant mode before truncating it, so that it has zero dc-gain (i.e., adding a numerator-zero at zero frequency). This modification also enhances the near-orthogonality of the modes being truncated. The D matrix must be compensated as below for each pole removed.
The diagonalized reduced-order system is now \( \{\Lambda, \hat{B}, \hat{C}, \hat{D}\} \).

In order to obtain high-order models to experiment with, a function that generates pseudorandom, realistic state space models was employed. This function generates pseudorandom state-space models that exhibit many of the characteristics associated with flexible structures, particularly in terms of pole and zero density and distribution across the frequency-range of the model. The function allows the average damping factor of the poles to be specified as an input argument, enabling the creation of models with modes having any desired average damping factor.

The first system explored is a 20\(^{th}\)-order continuous-time single-input single-output (SISO) system. The estimated Hankel singular values are calculated and arranged from largest to smallest in magnitude.

![Figure 3.1](image)

**Figure 3.1.** Hankel singular values and *estimated* Hankel singular values
Figure 3.2. Comparison of singular value estimation errors

As can be seen in Fig. 3.1 and 3.2 all of the different methods of estimating the Hankel singular values are very accurate in a small model. The largest error is $3.976 \times 10^{-4}$. They are very accurate because of the diagonal dominance of the observability and controllability matrices. When the off-diagonal elements of the controllability and observability grammians are larger there is more cross-correlation between the different modes of the system. In this case, though, the different HSV estimation methods for such cases yield essentially identical results.
Figure 3.3. Diagonalized-model controllability grammian, diagonal dominance

The diagonal dominance seen in Fig. 3.3 was computed from the absolute values of the elements in the grammian, because the non-diagonal elements of the grammian are typically non-real-valued.

Figure 3.4. FRF error of balanced model reduced by one pole-pair
Fig. 3.4 and 3.5 above show the FRF error resulting from reduction of a SISO 20\textsuperscript{th}-order system to an 18\textsuperscript{th}-order system. The diagonal reduction successfully removes the mode that has the least effect on the overall FRF. By comparison, the balanced reduction loses that same mode, but also significantly changes some aspects of the other modes, as well as the dc-gain, as shown in Fig. 3.6. The diagonal reduction is much more effective at reducing the model, in terms of preserving the original frequency-response function (FRF), as well as the original pole locations.
Figure 3.6. 20th-order model pole map, with reduced-model poles overlaid

The pole map in Fig. 3.6 shows that the diagonal reduction deletes the pole pair at -16 ± 16000j, and perfectly preserves the locations of all other poles. The balanced reduction also deletes that pole-pair and keeps some poles (those near the origin) close to their original locations, but significantly moves other poles (those that are farther away from the origin). In terms of preserving the original pole locations, the balanced reduction is clearly less effective than the diagonal reduction.

The next set of experiments was performed on a 50th-order continuous-time SISO system.
As the model-order increases, the Hankel singular values and the estimated Hankel singular values begin to diverge, particularly those with small magnitudes. The diagonal singular value calculation yields the closest result to the actual Hankel singular values. In reducing the 50th-order system to 40th-order, the diagonal method is again much better than the balanced reduction, in terms of preserving the FRF and the pole locations of the original model.

The row-sum and diagonal methods of Eq. 2.3.3 are further explored, but the inner-product and vector-magnitude methods of Eq. 2.3.4 were not pursued any further, due to the latter methods being demonstrably less successful than the former methods.

**Figure 3.7.** Hankel singular values, and estimated Hankel singular values
Figure 3.8. Balanced reduction of 50\textsuperscript{th}-order SISO down to 40\textsuperscript{th}-order

Figure 3.9. Diagonal reduction of 50\textsuperscript{th}-order SISO down to 40\textsuperscript{th}-order

The diagonal reduction preserves the FRF better than the balanced reduction, as shown in Fig. 3.8 and 3.9. This is seen very clearly at frequencies below 1,000 Hz.
Figure 3.10. Balanced reduction of 50\textsuperscript{th}-order SISO down to 40\textsuperscript{th}-order, high-frequency loss

Figure 3.11. Diagonal reduction of 50\textsuperscript{th}-order SISO down to 40\textsuperscript{th}-order, high-frequency loss
Fig. 3.10 and 3.11 are zoomed-in to the frequencies where both methods have difficulty preserving the FRF. It can be seen that even in these areas, there is generally less error using the diagonal reduction method than using the balanced reduction.

![Figure 3.12. SISO 50th-order pole map, with reduced-model poles overlaid](image)

The balanced reduction moves one pole-pair to a very heavily-damped location. Moreover, while the structure of the original system is lightly-damped, the balanced method changes that structure, resulting in a model in which many of the poles have been moved to even lighter-damped locations than they originally had.

In order to easily compare the various reduced-order models, in terms of their FRF error, it is useful to have a single number that captures the overall FRF error. For SISO models, the FRF error is the difference between the FRF vector of the actual model and the FRF vector of the reduced model. For a multiple-input multiple-output (MIMO) model, however, the FRFs are not vectors, so the FRF error is calculated as the Frobenius norm (at each frequency) of the error matrix. The Frobenius norm is used
because it is the closest measure of the Euclidean norm of a matrix. This gives a “matrix length” of the error, and the shorter the length of the error the closer the reduced system is to the original system. The Frobenius norm is easy to compute and invariant under rotation. The error matrix is the difference between the original model and reduced model FRF matrices. The Frobenius norm is computed by

\[ \sqrt{\sum_{k=1}^{\min\{m,n\}} \sigma_k^2} \]

where \( \sigma_k \) is the \( k^{th} \) singular value of the error matrix. To determine a single number that captures the overall error function, the FRF error function is integrated with frequency-weighting. Frequency-weighting is accomplished by using logarithmically equally-spaced frequencies, effectively putting equal emphasis on errors at all frequencies.

Based on the previous experiments, a new method of determining modal importance was developed and called the \textit{FRF sweep}. A single mode (i.e., a complex pole-pair) is removed from the model and the FRF error between that and the original model is calculated. That mode is put back into the model and the next mode is removed. Finally, the modes are arranged according to the amount of FRF error produced by their individual removal. Because lightly-damped modes are approximately orthogonal in the FRF sense, the FRF sweep should give the best results. The FRF sweep takes much longer than the other importance calculation methods, but it turns out to be useful as a benchmark to judge the other methods. The MATLAB code to run the FRF sweep can be found in Appendix B.

A continuous-time 50\textsuperscript{th}-order two-input two-output (2I2O) system is generated next. The model order is first reduced by removing ten poles.
Figure 3.13. 50th-order continuous-time 2I2O reduced to 40th-order using balanced reduction

Figure 3.14. 50th-order continuous-time 2I2O reduced to 40th-order using diagonal reduction

For a ten-pole reduction, the balanced-model order reduction has an error sum of 2392.1 and the diagonalized-model order reduction has an error sum of 273.7. Based on this measure, the balanced
reduction produced a reduced-order model that has 8.7 times greater FRF error than the model produced by the diagonal method.

All possible model-order reduction cases were examined, successively removing more and more modes from the model. The model was reduced from 50\(^\text{th}\)-order down to 48\(^\text{th}\), 46\(^\text{th}\), ..., 4\(^\text{th}\), and 2\(^\text{nd}\)-order. In each case, the resulting FRF errors were examined. As one might expect, FRF error tended to increase with the number of poles being removed. Interestingly, however, that trend was not typically observed when model-order was reduced by the standard balanced-model order reduction, as can be seen in most of the figures that follow.

![Figure 3.15. 50\(^\text{th}\)-order continuous-time 212O model reductions, FRF errors](image)

The two diagonalized-model reduction methods and the FRF reduction method are compared for all possible model-order reductions. As shown in Fig. 3.15 and 3.16, in terms of FRF error, the diagonal method does better than the row sum method. Moreover, the balanced method has worse FRF error in
all cases, and often much worse. In almost every case, the diagonal method produced a model with FRF error very close to that of the FRF sweep method.

**Figure 3.16.** 100th-order lightly-damped continuous-time 3I2O reductions, FRF errors

**Figure 3.17.** 200th-order lightly-damped continuous-time 3I2O reductions, FRF errors
The same thing that was observed in the 100\textsuperscript{th}-order case is also observed in the 200\textsuperscript{th}-order case. When only a few poles are removed from the model, the balanced reduction method does almost as well as the diagonal reduction.

From this point on, the row-sum method was not pursued because it did not exhibit any clear and consistent advantage, compared to the diagonal method. Furthermore, the diagonal method is quicker because it requires less computation.
Figure 3.19. 350\textsuperscript{th}-order lightly-damped continuous-time 2I2O reductions, FRF errors

Figure 3.20. 350\textsuperscript{th}-order lightly-damped continuous-time 2I2O errors (zoomed in)
In the 350\textsuperscript{th}-order experiments there are a few cases in which the balanced method results in lower FRF error, but in general the diagonal method results in the least error. The balanced method was better when only a relatively few poles were being removed; it turns out that this distinction disappears when the model has even more-lightly-damped poles.

\textbf{Figure 3.21.} \textsuperscript{500}th-order lightly-damped continuous-time 2I2O reductions, FRF errors
For the 500th order model, the balanced method is better when less than 150 poles are being removed from the system. When more than 150 poles are being removed, the diagonal method is better. When only a few poles are removed from the model, it seems that the balanced method has more freedom relocating the remaining poles. When more poles are removed, the balanced method has a tougher time compensating for the losses.

As more modes are placed closer together in the higher-order systems, the modes become less orthogonal to their neighboring modes. Because the diagonal reduction method is based on orthogonality between the modes, this coupling of modes makes the diagonal reduction less effective.

The diagonal method requires the least amount of calculation and is the most effective method of reducing the flexible models. The importance value for the diagonal importance value is

\[
\frac{\sqrt{B_k^*B_k^H}}{(\lambda_k + \bar{\lambda}_k)} \cdot \frac{C_k^H C_k}{(\bar{\lambda}_k + \lambda_k)} = \frac{\|C_k^*\| \|B_k\|}{2 \cdot \text{real}(\lambda_k)}
\]
The FRF can be calculated from equation 1. Looking more in depth to the diagonal form the transfer function can be written as

\[ H(s) = C \ast (sI - A)^{-1} \ast B + D \]

Neglecting the DC term and expanding out the matrices

\[
H(s) = [C_1 \cdots C_n] \ast \begin{bmatrix}
\frac{1}{s - \lambda_1} & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \frac{1}{s - \lambda_n}
\end{bmatrix} \ast [B_1 \cdots B_n]^T
\]

\[
H(s) = \begin{bmatrix}
c_{11} & \cdots & c_{n1} \\
\frac{s - \lambda_1}{s - \lambda_n} & \cdots & \frac{s - \lambda_n}{s - \lambda_n} \\
c_{1p} & \cdots & c_{np} \\
\frac{s - \lambda_1}{s - \lambda_n} & \cdots & \frac{s - \lambda_n}{s - \lambda_n}
\end{bmatrix} \ast [B_1 \cdots B_n]^T
\]

\[
H(s) = \begin{bmatrix}
c_{11} * b_{11} & \cdots & c_{n1} * b_{n1} \\
\frac{s - \lambda_1}{s - \lambda_n} & \cdots & \frac{s - \lambda_n}{s - \lambda_n} \\
c_{1p} * b_{11} & \cdots & c_{np} * b_{n1} \\
\frac{s - \lambda_1}{s - \lambda_n} & \cdots & \frac{s - \lambda_n}{s - \lambda_n}
\end{bmatrix}
\]

Note that this is a pxm matrix

The numerators can be simplified as outer products

\[
H(s) = \frac{C_1 * B_1}{s - \lambda_1} + \frac{C_2 * B_2}{s - \lambda_2} + \cdots + \frac{C_n * B_n}{s - \lambda_n}
\]

So the contribution from each pole can be written as

\[
\frac{C_k * B_k}{s - \lambda_k}
\]

Setting \( s \) equal to the imaginary part of \( \lambda_k \), which is very close to the peak of the FRF in a lightly damped system the contribution to the FRF at that frequency is

\[
\frac{C_k * B_k}{-\text{real}(\lambda_k)} \text{ \ 3.1.2}
\]
Taking the norm of Eq. 3.1.2

\[
\|C_k \ast B_k\| \over \text{real}(\lambda_k)
\]

\(\|C_k \ast B_k\|\) is an inner product between two vectors. By definition of the inner product of two vectors, this can be rewritten as \(\|C_k\| \ast \|B_k\| \ast \cos(\theta)\), where \(\theta\) is the angle in between the two vectors. Since \(\cos(\theta)\) is always less than or equal to one this relates to Eq. 3.1.1 by the inequality and a scaling factor of 2 since \(\|C_k \ast B_k\| \leq \|C_k\| \ast \|B_k\|\). Since all of the modes are scaled the same way the scaling factor is arbitrary. The diagonal importance value is the upper bound of the magnitude of the FRF contribution of a single pole near its maximum magnitude. This importance value is very good at determining the overall impact of each pole on the entire FRF since the poles are lightly damped.

![Diagram](image)

**Figure 3.23.** 500\(^{th}\) order continuous-time 2120 reduced to 348\(^{th}\)-order by diagonal reduction
When the 500\textsuperscript{th}-order system is reduced to a 348\textsuperscript{th}-order system the diagonal method performs better based on the FRF error integration metric. There is a nearly constant −17 dB FRF error using the balanced reduction method, across all frequencies. By comparison, the diagonal method very accurately preserves the FRF at frequencies below 500 Hz. Beyond 500 Hz, the error peaks are generally higher than the balanced method, with the largest error peak at about +3 dB, and the average error is also higher. However, this higher FRF error still does not cause the weighted integral of the error to significantly exceed that of the balanced method. It is also often true that we prefer any mismodeling to occur at higher, rather than lower, frequencies because the purpose of the model is to be used as the basis for designing a controller with lower-frequency bandwidth.
In Fig. 3.25, it is interesting to see how many of the poles are moved to much more heavily damped locations when the balanced reduction method is used. In fact, all of the original poles are moved to new locations, although most of them are not moved such long distances. However, by comparison, all remaining poles in the diagonal reduction are in exactly the same locations they had in the original model.
**Figure 3.26.** 500th order continuous-time 2I2O reduced to 348th-order transmission zero map

**Figure 3.27.** 500th-order continuous 2I2O reduced to 348th-order transmission zero map (zoomed in)
It is also interesting to compare the locations of the *transmission zeros* of the original model to those of the two reduced-order models. As can be seen in Fig. 3.26 and 3.27 the transmission zeros are generally not moved as far from their original locations when the diagonal reduction method is used. The diagonal method does move two transmission zeros to very negative locations on the real axis, but for practical purposes, these have very similar effect to transmission zeros at infinity. In that sense, it is much the same as deleting two of the original finite transmission zeros.

By contrast, Fig. 3.27 shows that the balanced reduction method moves many of the transmission zeros quite far from their original locations.

![Figure 3.28. 1000th-order lightly-damped continuous-time Z2O error plots](image)

In Fig. 3.28 and 3.29 the system being shown has damping factors that range from 0.0010 to 0.0109. In terms of FRF error, the balanced method is better in removing up to about 320 poles of the 1000 poles, or about one-third of them. However, when removing more than 320 poles, the diagonal method produces models with less FRF error.
Figure 3.29. 1000\textsuperscript{th}-order lightly-damped continuous-time $2\times 2$O error plots (zoomed in)

Figure 3.30. 1000\textsuperscript{th}-order continuous-time Hankel singular values
Figure 3.31. 1000\textsuperscript{th}-order continuous-time 2\textsuperscript{I}2\textsuperscript{O} reduced to 500\textsuperscript{th} order using diagonal reduction

Figure 3.32. 1000\textsuperscript{th}-order continuous-time 2\textsuperscript{I}2\textsuperscript{O} reduced to 500\textsuperscript{th} order using balanced reduction
Figure 3.33. 1000th-order lightly-damped continuous-time 2I2O error plots

Figure 3.34. 1000th-order lightly-damped continuous-time 2I2O error plots (zoomed in)
In Fig. 3.33 and 3.34, the system being shown has damping factors that range from 0.0005 to 0.0017. When the damping is this light, the diagonal method is better for removing any number of modes. This is attributable to the near orthogonality of every mode of the system, at such low levels of damping.

**Figure 3.35.** 1000\textsuperscript{th}-order lightly-damped continuous-time Hankel singular values

When a system is more lightly damped, the Hankel singular values are higher in magnitude than in the more heavily-damped system. As can be seen in Fig. 3.30 the lowest Hankel singular value is at -91dB. In the lightly damped system shown in Fig. 3.35, the lowest Hankel singular value is only at -65dB. As the system becomes less damped each mode of the system becomes more controllable and observable. Note, however, that the ratio of largest HSV to smallest HSV is roughly the same in both cases.
Figure 3.36. 1000\textsuperscript{th}-order lightly-damped 2I2O reduced to 500\textsuperscript{th} order using diagonal reduction

As can be seen in Fig. 3.36 and 3.37 the diagonal reduction method results in much less error than the balanced method. Even though the peak error of the diagonal reduction method is 6dB, but for the
balanced method only 3dB, the measure of the error favors the diagonal method because the largest errors in the diagonal reduction occur only over relatively narrow frequency ranges.

When comparing Fig. 3.36 and 3.37 to the more heavily damped system of Fig. 3.31 and 3.32, the diagonal reduction method results in 2131.7 less FRF error and the balanced reduction results in 3566.4 more error. Generally, we observed that when a system is very lightly damped, the diagonal reduction method results in less FRF error. This makes the diagonal reduction method a better choice over the balanced method for the reduction of high-order models of flexible structures, which tend to have lightly damped resonant modes.
3.2 Discrete-time systems

Most controllers are implemented inside computers. Since computers work in discrete-time it is often necessary to use a sampled discrete-time model. It turns out that the diagonal reduction computations to reduce the discrete-time models are just as computationally simple as the continuous-time counterparts.

To perform a discrete-time diagonalized-model reduction, the model must first be transformed into diagonalized form.

\[
\{A, B, C, D, T_z\} \Rightarrow \{\bar{A}, \bar{B}, \bar{C}, D, T_z\}
\]

As in the continuous-time case, without loss of generality, we can assume that the model of the flexible structure does not contain any repeated-integrator poles. However, in discrete-time it is also possible to have multiple delay poles, which are located at \(z=0\). In such a case, similar to the multiple-integrator case, the model can be decoupled into the states related to the (unique) resonant pole-pairs and the states related to a Jordan-block representing the delay poles. Due to that complete decoupling, the comments in this section apply, without any reservations, to reducing the number of complex pole-pairs.

From the diagonalized form, the observability and controllability grammians can be calculated in a straightforward way. This is done using the diagonalized-model grammian calculations in Eq. 2.2.7 and 2.2.8, instead of solving the Lyapunov equations in the usual way, because the diagonal calculation is much simpler. From the grammians of the diagonalized-model, estimates of the Hankel singular values are calculated using Eq. 2.3.1-4. The matrices \(A, B,\) and \(C\) are then rearranged in order of decreasing observability and controllability. The model matrices can then be truncated to the desired order.
As in the continuous-time case, the unavoidable change in dc gain resulting from the removal of modes is compensated by modifying the D matrix, The D matrix must be compensated as below for each pole removed.

\[ \tilde{D} = D + \frac{\bar{c}^T\bar{b}}{1-\bar{A}} \]

The reduced-order diagonalized-model is now \( \{\Lambda, \tilde{B}, \tilde{C}, \tilde{D}, T_s\} \)

As described in the rest of this section, the discrete-time results are quite similar to the continuous-time results.

![Graph](image)

**Figure 3.38.** 50\(^{th}\) order discrete-time 2120 error plots

The diagonalized-model reduction results in less error than the balanced method for the 50\(^{th}\) order discrete case.
Figure 3.39. 50th-order discrete-time 2I2O error plots

Figure 3.40. 50th-order discrete-time 2I2O reduced to 32nd-order using diagonal reduction
Figure 3.41. 50th-order discrete-time 2I2O reduced to 32nd-order using balanced reduction

The balanced reduction method moves poles from their original locations in discrete-time, just as it does in the continuous-time case.

The balanced reduction method does not perform well, in terms of maintaining the same FRF, as can be seen in Fig. 3.41. There is a significant amount of error at nearly every frequency. The diagonal reduction method, as seen in Fig. 3.40, is considerably better at preserving the model FRF, particularly at low frequencies. In the diagonal reduction, the least important modes of the system are removed and a very accurate model is produced.
Figure 3.42. 100\textsuperscript{th}-order lightly-damped discrete-time 3I2O error plots

Figure 3.43. 200\textsuperscript{th}-order lightly-damped discrete-time 2I3O error plots
As can be seen in Fig. 3.42 and 3.43 the diagonal method is better once again for 100\textsuperscript{th}-order and 200\textsuperscript{th}-order models, reduced by any number of poles, all the way down to second-order. As in the continuous-time case, the measure of goodness is based on the difference between the full-order model FRF and the reduced-order model FRF.

![Graph showing error vs. number of poles remaining for 350\textsuperscript{th}-order system](image)

**Figure 3.44.** 350\textsuperscript{th}-order lightly-damped discrete-time 2I2O error plots

As can be seen in Fig. 3.45 and 3.46 the diagonal method is comparable to the balanced method when removing up to 50 poles from a 350\textsuperscript{th} order system. The diagonal method results in less error when more than 50 poles are removed.
Figure 3.45. 350\textsuperscript{th}-order lightly-damped discrete-time 2I2O error plots (zoomed in)

Figure 3.46. 500\textsuperscript{th} order lightly-damped discrete time 2I2O error plots
As can be seen in Fig. 3.46 and 3.47, the balanced method results in less error in reducing the 500th-order model, when removing up to 200 poles. When the number of poles being removed is between 200 and 350, the resulting errors are comparable in both methods. But when more than 350 poles are being removed, the diagonal reduction method is better.

The advantage seen in these figures, for the balanced method, when removing only a small number of poles, does extend to cases where the poles are very lightly-damped. In such cases, the diagonal reduction method is superior no matter how many poles are being removed. That result mirrors the result seen in the continuous-time case.
4. Conclusion

The approximate orthogonality of lightly damped modes and the decoupling of the modes in diagonal form make the diagonal reduction method a computationally cheap and effective way of reducing models of flexible structures. In low order models, if the modes are spaced farther apart in frequency the diagonal model order reduction is more successful than other methods in reducing models even when the model is more heavily damped. As the order of the model increases and the modes are closer together in frequency, the modes become more coupled and less orthogonal. When the modes are more coupled, the diagonal reduction method becomes less effective, and the balanced reduction method is more effective at removing them. If the damping of the modes is significantly light, then even if the modes are close together they are still, for practical purposes, nearly orthogonal, and the diagonal method is more effective in these cases.

The diagonal reduction method maintains the original structure of the model in the reduced model. The least observable and controllable poles of the system are removed, and the remaining poles of the reduced system are in exactly the same locations they had in the original model. The transmission zeros locations move, but they do not move very far. By comparison, the balanced reduction method moves both the poles and the transmission zeros of the original system when reducing a system. The system goes from being a very lightly damped structure to a structure with more heavily damped poles. It is often more desirable to preserve the structure of the system, as the diagonal method does.

Up to this point all of the systems explored contained only complex pole pairs. The balanced model order reduction method will move real valued poles from their original locations. On the other hand, the diagonal reduction method will not move (or remove) real valued poles, which is preferred in many situations.
The diagonal method preserves the FRF of the system at all frequencies except in a close neighborhood around the mode that was removed. When the mode that is removed is very lightly damped this introduces an error over a very narrow frequency band. All other frequencies are essentially unaffected. This method is very effective at maintaining almost no FRF error in the low frequency range, and it produces no error in the dc gain of the model. By contrast, the balanced reduction method reduces a model in such a way that the FRF error is distributed nearly equally across all frequency points.

In the future, more diverse systems should be explored. A system with a wider variety of damping factors should be investigated. A damping factor weighting can be added into the importance value calculation if needed to compensate for the greater variety in pole damping. Other frequency weighting methods can also be explored for cases where accuracy at higher frequency poles in the model may be considered more important than accuracy around lower frequency poles.
References


Appendix A Continuous and discrete time grammian calculation functions

function [Q,P]=c_diag_gram_calc(A,B,C)
%This function takes three continuous time state space matrices A,B,C, and assumes
%diagonal A matrix and outputs the controllability and observability matricies

order=length(A,1);
P=zeros(order);
Q=zeros(order);

for i=1:order
    for j=1:order
        P(i,j)=-(B(i,:)*B(j,:)')/(A(i,i)+conjugate(A(j,j)));
        Q(i,j)=-(C(:,i)'*C(:,j))/(A(i,i)+conjugate(A(j,j)));
    end
end

function [Q,P]=d_diag_gram_calc(A,B,C)
%This function takes the three discrete time state space matrices A,B,C, and assumes
%diagonal A matrix and outputs the controllability and observability matricies

order=length(A,1);
P=zeros(order);
Q=zeros(order);

for i=1:order
    for j=1:order
        P(i,j)=-(B(i,:)*B(j,:)')/(1-(A(i,i)*conjugate(A(j,j))));
        Q(i,j)=-(C(:,i)'*C(:,j))/(1-(A(i,i)*conjugate(A(j,j))));
    end
end
function [reduced_sys] = frf_err_redux(diag_sys, red_order)
% this function removes one pair of poles from the diag_sys,
% calculates the error created in the frf by removing the pole and removes
% the poles with the least amount of effect on the frf

A = diag_sys.a;
B = diag_sys.b;
C = diag_sys.c;
D = diag_sys.d;
Ts = diag_sys.ts;

n_states = size(A,1);
if Ts == 0
    w = (2*pi) * logspace(-1, 4.5, 10000);
else
    w = (2*pi) * logspace(-1, log10(0.5/Ts), 10000);
end
D_temp = D;
err_vec = zeros(1, n_states/2);
sys_frf = freqresp(diag_sys, w);

wbar = waitbar(0, 'FRF Reduction...
for k=1:2:n_states
waitbar(k/n_states);
if k == 1
    A_temp = A(3:n_states, 3:n_states);
    B_temp = B(3:n_states,:);
    C_temp = C(:, 3:n_states);
elseif k < n_states - 1
    temp = [A(k+2:n_states, 1:k-1), A(k+2:n_states, k+2:n_states)];
    A_temp = [A(1:k-1, 1:k-1), A(1:k-1, k + 2:n_states); temp];
    B_temp = [B(1:k-1,:); B(k + 2:n_states,:)];
    C_temp = [C(:, 1:k-1), C(:, k+2:n_states)];
else
    A_temp = A(1:n_states - 2, 1:n_states - 2);
    B_temp = B(1:n_states - 2,:);
    C_temp = C(:, 1:n_states - 2);
end
reduced_sys = ss(A_temp, B_temp, C_temp, D_temp, Ts);
reduced_frf = diag_freq_resp(reduced_sys, w);
err_vec(round(k/2)) = frf_error(sys_frf, reduced_frf);
end

close(wbar)
clearwbar

% sys reduction
[err_vec, index] = sort(err_vec, 'descend'); % #ok<ASGLU>
A_ordered = zeros(size(A));
B_ordered = zeros(size(B));
C_ordered = zeros(size(C));
for k = 1:numel(index)
    A_ordered(2*k-1,2*k-1) = A(2*index(k)-1,2*index(k)-1);
    A_ordered(2*k,2*k) = A(2*index(k),2*index(k));
    B_ordered(2*k-1,:) = B(2*index(k)-1,:);
    B_ordered(2*k,:) = B(2*index(k),:);
    C_ordered(:,2*k-1) = C(:,2*index(k)-1);
    C_ordered(:,2*k) = C(:,2*index(k));
end
A_red = A_ordered(1:red_order,1:red_order);
B_red = B_ordered(1:red_order,:);
C_red = C_ordered(:,1:red_order);

%modify the D matrix to make up for loss of DC data
DC_comp_nr = n_states-red_order;
D_red = zeros(size(D));
for k=1:DC_comp_nr
    D_red = D_red + D_temp = -C(:,n_states - k+1)*A(k,k)*B(k,:)-C(:,k + 1)*A(k,k)*B(k + 1,:);
end
reduced_sys = ss(A_red, B_red, C_red, D_red, Ts);
function [Diag Redux, Scaled Diag Redux, IP Redux, Row Sum Redux, Row Abs Sum Redux, ... Row Mag Redux, Q Diag, P Diag, est sv] = ...
c_diag_sys_reduction(A_diag, B_diag, C_diag, D_diag, reduction_order)
% this function takes in a diagonal system and minimizes it without
% diagonalizing the system

% this function calculates the grammians of a diagonalized system
[Q_diag, P_diag] = c_diag_gram_calc(A_diag, B_diag, C_diag);

order = numel(A_diag(1,:));

% Different estimates of observability/controllability importance
est_diag_sv = zeros(order, 1);
est_scaled_diag_sv = zeros(order, 1);
est_ip_sv = zeros(order, 1);
est_row_sum_sv = zeros(order, 1);
est_row_abs_sum_sv = zeros(order, 1);
est_row_mag_sv = zeros(order, 1);

for k = 1:order
    est_diag_sv(k) = sqrt(Q_diag(k,k) * P_diag(k,k));
est_scaled_diag_sv(k) = est_diag_sv(k)/abs(imag(A_diag(k,k)));
est_ip_sv(k) = abs(sqrt(Q_diag(k,:)*P_diag(:,k)));
est_row_sum_sv(k) = sqrt(abs(sum(Q_diag(k,:)) * abs(sum(P_diag(:,k)))));
est_row_abs_sum_sv(k) = sqrt(sum(abs(Q_diag(k,:)) * sum(abs(P_diag(:,k)))));
est_row_mag_sv(k) = sqrt(vec_mag(Q_diag(k,:)) * vec_mag(P_diag(:,k)));
end

% diag sys reduction
[est_diag_sv, index] = sort(est_diag_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
    A_ordered(k,k) = A_diag(index(k),index(k));
    B_ordered(k,:) = B_diag(index(k),:);
    C_ordered(:,k) = C_diag(:,index(k));
end

A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);

% modify the D matrix to make up for loss of DC data
DC_comp_nr = order - reduction_order;
D_red = D_diag;
if DC_comp_nr ~= 0
for k=1:DC_comp_nr
    D_red = D_red - C_ordered(:,n_states - k+1)*A_ordered(n_states - k+1,n_states - k+1)*B_ordered(n_states - k+1,:);
end
end

Diag_redux = ss(A_red, B_red, C_red, D_red);

% scaled diag sys reduction
[est_scaled_diag_sv, index] = sort(est_scaled_diag_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
    A_ordered(k,k) = A_diag(index(k),index(k));
    B_ordered(k,:) = B_diag(index(k),:);
    C_ordered(:,k) = C_diag(:,index(k));
end

A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);

% modify the D matrix to make up for loss of DC data
D_red = D_diag;
if DC_comp_nr ~= 0
    for k=1:DC_comp_nr
        D_red = D_red - C_ordered(:,n_states - k+1)*A_ordered(n_states - k+1,n_states - k+1)*B_ordered(n_states - k+1,:);
    end
end

Scaled_Diag_redux = ss(A_red, B_red, C_red, D_red);

% inner product sys reduction
[est_ip_sv, index] = sort(est_ip_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
    A_ordered(k,k) = A_diag(index(k),index(k));
    B_ordered(k,:) = B_diag(index(k),:);
    C_ordered(:,k) = C_diag(:,index(k));
end

A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);

% modify the D matrix to make up for loss of DC data
D_red = D_diag;
if DC_comp_nr ~= 0
    for k=1:DC_comp_nr
        D_red = D_red - C_ordered(:,n_states - k+1)*A_ordered(n_states - k+1,n_states - k+1)*B_ordered(n_states - k+1,:);
    end
end
IP_redux = ss(A_red, B_red, C_red, D_red);

% row sum sys reduction
[est_row_sum_sv, index] = sort(est_row_sum_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
A_ordered(k,k) = A_diag(index(k),index(k));
B_ordered(k,:) = B_diag(index(k,:));
C_ordered(:,k) = C_diag(:,index(k));
end

A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);
D_red = D_diag;
if DC_comp_nr ~= 0
for k=1:DC_comp_nr
D_red = D_red - C_ordered(:,n_states - k+1)*A_ordered(n_states - k+1,n_states - k+1)*B_ordered(n_states - k+1,:);
end
end

Row_Sum_redux = ss(A_red, B_red, C_red, D_red);

% inner product sys reduction
[est_row_abs_sum_sv, index] = sort(est_row_abs_sum_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
A_ordered(k,k) = A_diag(index(k),index(k));
B_ordered(k,:) = B_diag(index(k,:));
C_ordered(:,k) = C_diag(:,index(k));
end

A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);
D_red = D_diag;
if DC_comp_nr ~= 0
for k=1:DC_comp_nr
D_red = D_red - C_ordered(:,n_states - k+1)*A_ordered(n_states - k+1,n_states - k+1)*B_ordered(n_states - k+1,:);
end
end
Row_Abs_Sum_redux = ss(A_red, B_red, C_red, D_red);

%row_mag sys reduction
[est_row_mag_sv, index] = sort(est_row_mag_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));
for k = 1:order
A_ordered(k,k) = A_diag(index(k),index(k));
B_ordered(k,:) = B_diag(index(k),:);
C_ordered(:,k) = C_diag(:,index(k));
end
A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);
%modify the D matrix to make up for loss of DC data
D_red = D_diag;
ifDC_comp_nr ~= 0
for k=1:DC_comp_nr
D_red = D_red -C_ordered(:,n_states - k+1)*A_ordered(n_states - k+1,n_states - k+1)*B_ordered(n_states - k+1,:);
end
end

Row_Mag_redux = ss(A_red, B_red, C_red, D_red);

est_sv=zeros(order,6);
est_sv(:,1)=est_diag_sv;
est_sv(:,2)=est_ip_sv;
est_sv(:,3)=est_row_sum_sv;
est_sv(:,4)=est_row_abs_sum_sv;
est_sv(:,5)=est_row_mag_sv;
est_sv(:,6)=est_scaled_diag_sv;
Appendix D  Discrete Time Diagonal System Reduction

function[Diag_redux, Scaled_Diag_redux, IP_redux, Row_Sum_redux, Row_Mag_redux, Q_diag, P_diag, est_sv] = ...
d_diag_sys_reduction(A_diag, B_diag, C_diag, D_diag, Ts, reduction_order)
%this function takes in a diagonal system and minimizes it without
%diagonalizing the system

%this function calculates the grammians of a diagonalized system
[Q_diag, P_diag] = d_diag_gram_calc(A_diag, B_diag, C_diag);
order = numel(A_diag(1,:));

%Different estimates of observability/controllability importance
est_diag_sv = zeros(order, 1);
est_scaled_diag_sv = zeros(order, 1);
est_ip_sv = zeros(order, 1);
est_row_sum_sv = zeros(order, 1);
est_row_abs_sum_sv = zeros(order, 1);
est_row_mag_sv = zeros(order, 1);

for k = 1:order
    est_diag_sv(k) = sqrt(Q_diag(k,k) * P_diag(k,k));
est_scaled_diag_sv(k) = est_diag_sv(k)/abs(angle(A_diag(k,k)));
est_ip_sv(k) = abs(sqrt(Q_diag(k,:) * P_diag(:,k)));
est_row_sum_sv(k) = sqrt(abs(sum(Q_diag(k,:))) * abs(sum(P_diag(:,k))));
est_row_abs_sum_sv(k) = sqrt(sum(abs(Q_diag(k,:))) * sum(abs(P_diag(:,k))));
est_row_mag_sv(k) = sqrt(vec_mag(Q_diag(k,:)) * vec_mag(P_diag(:,k)));
end

%diag sys reduction
[est_diag_sv, index] = sort(est_diag_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
    A_ordered(k,k) = A_diag(index(k),index(k));
    B_ordered(k,:) = B_diag(index(k),:);
    C_ordered(:,k) = C_diag(:,index(k));
end

A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);
%modify the D matrix to make up for loss of DC data
DC_comp_nr = order - reduction_order;

if DC_comp_nr ~= 0
    for k = 1:DC_comp_nr
D_red = D_red + C_ordered(:,order - k+1)/(1-A_ordered(order - k+1,order - k+1))*B_ordered(order - k+1,:); end end

Diag_redux = ss(A_red, B_red, C_red, D_red, Ts);

%scaled diag sys reduction
[est_scaled_diag_sv, index] = sort(est_scaled_diag_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
A_ordered(k,k) = A_diag(index(k),index(k));
B_ordered(k,:) = B_diag(index(k),:);
C_ordered(:,k) = C_diag(:,index(k));
end

A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);
%modify the D matrix to make up for loss of DC data
D_red = D_diag;
if DC_comp_nr ~= 0
for k=1:DC_comp_nr
D_red = D_red + C_ordered(:,order - k+1)/(1-A_ordered(order - k+1,order - k+1))*B_ordered(order - k+1,:);
end
end

Scaled_Diag_redux = ss(A_red, B_red, C_red, D_red, Ts);

%inner product sys reduction
[est_ip_sv, index] = sort(est_ip_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
A_ordered(k,k) = A_diag(index(k),index(k));
B_ordered(k,:) = B_diag(index(k),:);
C_ordered(:,k) = C_diag(:,index(k));
end

A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);
%modify the D matrix to make up for loss of DC data
D_red = D_diag;
if DC_comp_nr ~= 0
for k=1:DC_comp_nr
D_red = D_red + C_ordered(:,order - k+1)/(1-A_ordered(order - k+1,order - k+1))*B_ordered(order - k+1,:);
end
end
end
end

IP_redux = ss(A_red, B_red, C_red, D_red, Ts);

% row sum sys reduction
[est_row_sum_sv, index] = sort(est_row_sum_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
    A_ordered(k,k) = A_diag(index(k),index(k));
    B_ordered(k,:) = B_diag(index(k),:);
    C_ordered(:,k) = C_diag(:,index(k));
end
A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);

% modify the D matrix to make up for loss of DC data
D_red = D_diag;
ifDC_comp_nr ~= 0
    for k=1:DC_comp_nr
        D_red = D_red + C_ordered(:,order-k+1)/(1-A_ordered(order-k+1,order-k+1))*B_ordered(order-k+1,:);
    end
end

Row_Sum_redux = ss(A_red, B_red, C_red, D_red, Ts);

% inner product sys reduction
[est_row_abs_sum_sv, index] = sort(est_row_abs_sum_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
    A_ordered(k,k) = A_diag(index(k),index(k));
    B_ordered(k,:) = B_diag(index(k),:);
    C_ordered(:,k) = C_diag(:,index(k));
end
A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);

% modify the D matrix to make up for loss of DC data
D_red = D_diag;
ifDC_comp_nr ~= 0
    for k=1:DC_comp_nr
        D_red = D_red + C_ordered(:,order-k+1)/(1-A_ordered(order-k+1,order-k+1))*B_ordered(order-k+1,:);
    end
end

Row_Abs_Sum_redux = ss(A_red, B_red, C_red, D_red, Ts);
%row_mag sys reduction
[est_row_mag_sv, index] = sort(est_row_mag_sv,'descend');
A_ordered = zeros(size(A_diag));
B_ordered = zeros(size(B_diag));
C_ordered = zeros(size(C_diag));

for k = 1:order
A_ordered(k,k) = A_diag(index(k),index(k));
B_ordered(k,:) = B_diag(index(k),:);
C_ordered(:,k) = C_diag(:,index(k));
end

A_red = A_ordered(1:reduction_order,1:reduction_order);
B_red = B_ordered(1:reduction_order,:);
C_red = C_ordered(:,1:reduction_order);
if DC_comp_nr ~= 0
for k=1:DC_comp_nr
D_red = D_red +C_ordered(:,order-k+1)/(1-A_ordered(order-k+1,order-k+1))*B_ordered(order-k+1,:);
end
end

Row_Mag_redux = ss(A_red, B_red, C_red, D_red, Ts );

est_sv=zeros(order,6);
est_sv(:,1)=est_diag_sv;
est_sv(:,2)=est_ip_sv;
est_sv(:,3)=est_row_sum_sv;
est_sv(:,4)=est_row_abs_sum_sv;
est_sv(:,5)=est_row_mag_sv;
est_sv(:,6)=est_scaled_diag_sv;