MODEL REDUCTION IN SCHUR BASIC WITH POLE RETENTION AND H₂ NORM ERROR BOUND

Vu Ngoc Kien, Nguyen Hong Quang*
Thai Nguyen University of Technology, No. 666, 3/2 Street, Thai Nguyen, Viet Nam

* Corresponding Author Email: quang.nguyenhong@tnut.edu.vn

ABSTRACT: Model order reduction is a field of study to identify low-order systems to replace the high-order systems while preserving the essential properties of the original system. One of critical properties of the high-order systems that need to be preserved in the order reduction is dominant poles. This paper introduces a model order reduction algorithm by preserving dominant poles of the original system in the reduced-order system. By evaluating the dominance of the poles according to the H₂ standard, the algorithm will convert the matrix A of the original system to an upper triangular form and then arrange the poles of the system according to the decreasing importance on the main diagonal of the upper triangular matrix. Using the truncation technique, the algorithm will retain the dominant poles corresponding to the top positions on the main diagonal of the upper triangular matrix to obtain the reduced-order system. Applying the algorithm to reduce the order of the high-order controller and the high-order filter shows that the algorithm is able to arrange the poles according to the H₂ dominance and preserve the dominant poles of the original system in the order reduction system.

KEYWORDS: Model reduction algorithm, H₂ standard, Dominant poles, Upper triangular matrix, Truncation technique.

INTRODUCTION

Truncation algorithm is a popular technique used in the field of model order reduction. This algorithm is implemented by two steps. In the first step, a non-singular matrix is used to transform the original system to an equivalent state-space system. In the second step, based on the desired order of the reduction system, a corresponding number of rows and columns are omitted to obtain the desired degree reduction system. Among algorithms that use the truncation technique for order reduction, the most widely used algorithm is the balanced truncation algorithm [1-7]. The balanced truncation algorithm determines a non-singular T matrix to convert the original system to a balanced equivalent system. In this form, the observation Gramian matrix and the control Gramian matrix of the system are diagonal. To determine T, the balanced truncation algorithm often uses Singular Value Decomposition (SVD) analysis. However, in order to use SVD analysis, the initial system must satisfy many conditions, which limits the applicable objects of the algorithm. To obtain an order-reduced system, the balanced truncation algorithm is based on the magnitude of the Hankel singular values and seeks to retain the large Hankel singular values of the original system. According to the balanced truncation algorithm, the poles of the original system will not be preserved in the order-reduced system. However, the poles in general and the dominant poles in particular of the original system are immutable and should be kept in the order-reduced system. Based on the perspective of preserving the dominant poles, many algorithms have been proposed to retain the dominant poles of the original system during truncation.

Among the proposed methods based on the basis of preserving the eigenvalues of the root system in the reduced-order system, the most general method is the aggregation method [8]. The advantage of the methods based on preserving eigenvalues is that by keeping the eigenvalues of the original model in the reduced-order model, the stability of the reduced-order model is preserved. However, the aggregation method to determine the degree reduction model requires eigenvalues and eigenvectors of the matrix A. Therefore, if the matrix A is extensive, the computational process consumes a massive amount of time. The second feature of the aggregation method is that the step response h(t) of the original model and the reduced-order model can be significantly different, which can be solved by combining the aggregation method with moments matching methods [9]. The critical problem in the aggregation method is how to
select the eigenvalues of the original model. This problem can be solved by a characteristic which is applied in analysis and synthesize system technique. The energy ratio criterion, based on the total impulse response energy of the output of the original model preserving the eigenvalues that contribute most of the total impulse response energy used to determine the most appropriate order for the reduced model, is proposed in [10]. In [11], the unit pulse is used to find a quantity that measures the influence of each specific value of the matrix A as a basis for determining decisive values. Another criterion proposed in [12] is to select main eigenvalues based on the contribution of each time-varying model to the characteristics of the input and output of the system. Another method for determining the main eigenvalue that has great attention recently is the dominant pole algorithm [13], [14]. The pole's dominance, which is based on the contribution of the poles to the output pulse response, is considered in [13], [14]. Then, the Arnoldi and Jacobi-Davidson methods, the Krylov subspace are used to identify and classify poles and preserve the dominant poles in the reduced-order system. However, the disadvantage of the algorithm [13], [14] is the high computational complexity and does not provide the upper bound formula for the order reduction error. On the other hand, we often use standard $H_2$ or $H_\infty$ to evaluate the order reduction error, so using the standard $H_2$ and $H_\infty$ to evaluate the dominance of the pole can help to obtain the small order reduction error. Therefore, in this paper, we introduce an algorithm to overcome the disadvantages of the algorithm in [13], [14]. Specifically, we use the standard $H$ to evaluate the dominance of the pole; Proposing to convert the matrix-A of original system to the upper triangular matrix form for simple calculation. The sequence of the algorithm can be described in 3 steps as follows:

Step 1: Transform the matrix-A of the original system to the upper triangular matrix form using Schur analysis (to limit the use of SVD analysis).

Step 2: Evaluate the critical properties of the pole based on the $H_2$ standard and arrange the dominant poles according to the descending order of dominance on the main diagonal of the upper triangular matrix.

Step 3: Delete some rows and columns according to step 2 of the truncation technique to get the reduced-order system.

Through three steps of the algorithm, the dominant poles of the original system will be preserved in the reduced system. The rest of this paper is organized as follow: Section 2 introduces the detailed content of the new algorithm. Section 3 presents examples illustrating the efficiency of the new algorithm. Finally, the conclusion of the paper is presented in Section 4.

MODEL ORDER REDUCTION ALGORITHM

Problem of model order reduction

Consider a multiple-input and multiple-output linear continuous n-order system, represented by state-space equations as follows:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

$x \in \mathbb{R}^n, u \in \mathbb{R}^p, y \in \mathbb{R}^q, A \in \mathbb{R}^{nxn}, B \in \mathbb{R}^{nxp}, C \in \mathbb{R}^{qxn}$.

The aim is to determine an r-order system represented by state-space equations

$$\dot{x}_r = A_r x_r + B_r u$$
$$y_r = C_r x_r$$

$x_r \in \mathbb{R}^r, u \in \mathbb{R}^p, y_r \in \mathbb{R}^q, A_r \in \mathbb{R}^{rxr}, B_r \in \mathbb{R}^{rxp}, C_r \in \mathbb{R}^{qrx}$ with $r \leq n$;

so that the r-order system (2) can replace the original system (1).

Model reduction in Schur basic with pole retention

The order reduction algorithm is described as following:
Algorithm 1: Transform the system to the upper triangle form.

Input: The system is described in equation (1).

Step 1: Determine matrix \( Q \) by solving the equation \( \mathbf{A}^* \mathbf{Q} + \mathbf{Q} \mathbf{A} + \mathbf{C}^* \mathbf{C} = 0 \)

Step 2: Determine matrix \( R \) by Cholesky analysis \( \mathbf{Q} = \mathbf{R}^* \mathbf{R} \).

Step 3: Determine the unitary matrix \( U \) and the upper triangle matrix \( \Delta \) by solving: \( \mathbf{R} \mathbf{A} \mathbf{R}^{-1} = \mathbf{U} \mathbf{A} \mathbf{U}^* \)

Step 4: Determine matrix \( T = \mathbf{R}^{-1} \mathbf{U} \)

Step 5: Determine the system \( (\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \tilde{\mathbf{D}}) \) as below:

\[
\begin{align*}
\tilde{\mathbf{A}} &= \mathbf{T}^{-1} \mathbf{A} \\
\tilde{\mathbf{B}} &= \mathbf{T}^{-1} \mathbf{B} \\
\tilde{\mathbf{C}} &= \mathbf{C} \mathbf{T} \\
\tilde{\mathbf{D}} &= \mathbf{D}
\end{align*}
\]

Output: The system \( (\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \tilde{\mathbf{D}}) \) is in a form of upper triangle.

Algorithm 2: Rearrange the poles on the main diagonal of the matrix above, based on the \( H_2 \) dominant index of the poles.

Input: System \( (\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \tilde{\mathbf{D}}) \)

Step 1: Determine the value of the \( H_2 \) dominant index of each pole \( \lambda_i, i = 1 \ldots n \), using the formula:

\[
\tilde{S}_i = \sqrt{\text{trace}(\mathbf{B}_i^* \mathbf{B}_i)}.
\]

Step 2: Compare the values of \( H_2 \) dominant index between the poles to determine the pole with the largest \( H_2 \) dominant index \( \tilde{S}_h \)

Step 3: Replace the pole \( \lambda_{i_1} \) (which has the largest \( H_2 \) dominant index \( \tilde{S}_h \)) (and the conjugate value \( \bar{\lambda}_{i_1} \) if it exists) onto the first position on the diagonal of the matrix \( \tilde{\mathbf{A}} \) using matrix \( U_1 \) as below:

\[
U_1^* \tilde{\mathbf{A}} U_1 = \begin{bmatrix}
\lambda_{i_1} & * & * & * \\
* & \bar{\lambda}_{i_1} & * & * \\
* & * & * & * \\
\vdots & \ddots & \ddots & \ddots
\end{bmatrix}
\]

To perform this step in Matlab programming, we can use the “ordschur” command [15], [16].

Step 4: Using matrix \( U_1 \) to calculate equivalent system \( (U_1^* \tilde{\mathbf{A}} U_1, U_1^* \tilde{\mathbf{B}}, \tilde{\mathbf{C}} U_1) \).
Step 5: Determine the system \((\hat{A}, \hat{B}, \hat{C})\) with a size of \((n-2)\) by removing the first 2 rows and columns of the system 
\((U_1^T \hat{A} U_1, U_1^T \hat{B}, \hat{C} U_1)\)

Step 6: Apply steps 1 to 5 for the system \((\hat{A}, \hat{B}, \hat{C})\) and repeat these steps for the smaller subsystems until all the poles of the system have been arranged on the main diagonal of the matrix based on the magnitude of the H2 dominant index.

Output: A new system \(\left( \hat{A}, \hat{B}, \hat{C}, \hat{D} \right)\) that has poles with large H2 dominant index located at the first positions on the main diagonal of the matrix \(\hat{A}\).

Algorithm 3: Order reduction of the equivalent system

Input: System \(\left( \hat{A}, \hat{B}, \hat{C} \right)\)

Step 1: Choose \(r\) as the desired order of the reduced-order system so that \(r < n\).

Step 2: Perform in the following block form:

\[
\hat{A} = \begin{bmatrix}
\hat{A}_{11} & \hat{A}_{12} \\
0 & \hat{A}_{22}
\end{bmatrix}, \quad \hat{B} = \begin{bmatrix}
\hat{B}_1 \\
\hat{B}_2
\end{bmatrix}, \quad \hat{C} = \begin{bmatrix}
\hat{C}_1 & \hat{C}_2
\end{bmatrix}, \quad \hat{A}_{11} \in \mathbb{R}^{r \times r}, \quad \hat{B}_1 \in \mathbb{R}^{r \times p}, \quad \hat{C}_1 \in \mathbb{R}^{p \times r}
\]

Output: An \(r\)-order system \(\left( \hat{A}_{11}, \hat{B}_1, \hat{C}_1 \right)\)

THE ILLUSTRATIVE EXAMPLES

Model reduction of high order digital filter

Consider a high order digital filter in [17], represented as a transfer function model, as follows

\[
W(s) = \frac{-0.1242s^5 - 0.4629s^4 - 0.0823s^3 + 1.504s^2 + 1.959s + 1.401}{s^5 + 4.905s^4 + 10.82s^3 + 13.13s^2 + 9.533s + 3.834s + 0.9407}
\]

High order digital filter introduces several drawbacks as using in real applications such as: complicated structure; increase the signal processing time of the digital filter; the response time of the system using a digital filter is slow, it may cause the system cannot adapt the requirements of the system in real-time. Therefore, we need to reduce the order of high order digital filter to simplify the design of the digital filter, reduce the signal processing time of the digital filter, and reduced response time for systems using digital filters, but remain the main characteristic of the original digital filter.

Reducing the order of the high order digital filter follows algorithm 1, we get the equivalent system in an upper triangular form:

\[
\hat{A} = \begin{bmatrix}
-1.5218 & -3.0171 & -1.9266 & 2.4388 & -1.7309 & 0.0907 \\
0.2010 & -1.3028 & -1.7826 & 2.2565 & -1.6015 & 0.0839 \\
0.0000 & 0.0000 & 0.6098 & -0.3661 & -1.0957 & 0.0574 \\
0.0000 & 0.0000 & 1.9099 & -0.9771 & 1.3869 & -0.0727 \\
-0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.4922 & 0.5555 \\
0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.5039 & 0.0014
\end{bmatrix}, \quad \hat{B} = \begin{bmatrix}
-0.0127 \\
-0.3086 \\
-0.4320 \\
0.2982 \\
0.0873
\end{bmatrix}
\]

\[
\hat{C} = \begin{bmatrix}
1.7446 & 1.6142 & 1.1043 & -1.3979 & 0.9921 & -0.0520
\end{bmatrix}
\]
Following the steps of algorithm 2 to arrange the dominant poles, based on the descending magnitude of the $H_2$ dominant index, we obtain the system as follows:

$$\begin{bmatrix}
-0.0123 & -0.4527 & 0.0956 & -0.3608 & -0.0095 \\
0.6067 & -0.4812 & -0.5972 & 2.2539 & 1.7467 \\
0.0000 & 0.0000 & -0.1853 & -0.9097 & 1.0839 \\
-0.0000 & -0.0000 & 2.3083 & -4.0907 & -0.1396 \\
0.0000 & 0.0000 & 0.0000 & 0 & -1.5850 \\
0.0000 & 0.0000 & 0.0000 & -0.0000 & -1.1921 \\
\end{bmatrix} \quad ; \quad \begin{bmatrix}
-0.7449 \\
1.0839 \\
0.0000 \\
-1.1921 \\
0.0239 \\
0.0269 \\
\end{bmatrix}$$

Apply algorithm 3 to reduce the order of the system $(\hat{A}, \hat{B}, \hat{C})$, we obtain results as given in Table 1.

<table>
<thead>
<tr>
<th>Order</th>
<th>Transfer function $- W_{cr}(s)$</th>
<th>Error $|W(s) - W_{cr}(s)|_{H_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>$-0.1258s^4 - 0.51s^3 + 0.3043s^2 + 1.413s + 1.698$</td>
<td>0.0206</td>
</tr>
<tr>
<td></td>
<td>$s^5 + 4.903s^4 + 9.523s^3 + 8.828s^2 + 4.008s + 1.151$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$-0.1685s^3 - 0.1015s^3 + 0.2835s + 1.052$</td>
<td>0.0446</td>
</tr>
<tr>
<td></td>
<td>$s^4 + 3.318s^3 + 4.263s^2 + 2.07s + 0.7264$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$-0.2933s^2 + 0.3471s + 0.05881$</td>
<td>0.3580</td>
</tr>
<tr>
<td></td>
<td>$s^3 + 0.6788s^2 + 0.372s + 0.05199$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$-0.2549s + 0.3754$</td>
<td>0.1512</td>
</tr>
<tr>
<td></td>
<td>$s^2 + 0.4935s + 0.2806$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$-0.117$</td>
<td>10.9757</td>
</tr>
<tr>
<td></td>
<td>$s + 0.01233$</td>
<td></td>
</tr>
</tbody>
</table>

In addition to evaluating the order reduction results according to the order reduction error, we use step response and frequency response to evaluate reduced-order digital filters. Simulation results using Matlab/Simulink of the step response and the frequency response are shown in Figures 1 and Figure 2.
Figure 1. Simulating results of step response $h(t)$ of 6th-order digital filter and reduced-order digital filters
Figure 2. Simulating results of frequency response (bode plot) of 6th-order digital filter and reduced-order digital filter
Remark: From the resulting h(t) and bode plot of the original digital filter and reduced-order digital filter, we see:

+ For h(t): The h(t) of the 5th and 4th-order digital filter have slight deviations from h(t) of the 6th-order digital filter. The h(t) of the 3rd, 2nd, 1st-order digital filter have large deviations from the h(t) of the 6th-order digital filter.

+ For bode plot: The bode plot of the 5th-order digital filter exactly matches the bode plot of the 6th-order digital filter; The bode plot of the 4th-order digital filter has small deviations from the bode plot of the 6th-order digital filter. The bode plot of the 3rd, 2nd, 1st-order digital filter has a big deviation from the bode plot of the 6th-order digital filter.

Thus, the 5th, 4th-order digital filter can be used instead of the 6th order digital filter.

Reducing higher-order controller

Consider a 6th-order controller in the balanced two-wheel vehicle control problem [18] as follows:

\[
W_c(s) = \frac{1275s^5 + 8.695 \times 10^5 s^4 + 5.151 \times 10^7 s^3 + 1.359 \times 10^8 s^2 + 2.435 \times 10^7 s + 1.091 \times 10^6}{s^6 + 715.7s^5 + 2.355 \times 10^4 s^4 + 2.789 \times 10^5 s^3 + 3.802 \times 10^6 s^2 + 6.519 \times 10^5 s + 2.872 \times 10^4}
\]

The use of the 6th-order controller to control the balancing two-wheel vehicles will lead to the following problems: the control program code is complicated, the processing time will be longer, which can lead to the controller not responding to real-time control requirement. In order to make the control program code simpler, reduction of processing time and meet the real-time control requirements, we will reduce the 6th-order controller by algorithm 1, 2, 3. The results of controller order reduction are shown in Table 2.

### Table 2. Results of controller order reduction

<table>
<thead>
<tr>
<th>Order</th>
<th>Transfer function – (W_{cr}(s))</th>
<th>Error (|W_c(s) - W_{cr}(s)|_{\text{H}_2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>[\frac{1275s^4 + 8.694 \times 10^5 s^3 + 4.367 \times 10^7 s^2 + 1.359 \times 10^8 s + 1.209 \times 10^7}{s^5 + 715.6s^4 + 2.349 \times 10^4 s^3 + 2.768 \times 10^5 s^2 + 3.777 \times 10^6 s + 3.183 \times 10^5}]</td>
<td>1.1044.10-5</td>
</tr>
<tr>
<td>4</td>
<td>[\frac{1275s^3 + 348.1s^2 + 1.993 \times 10^5 s + 1.773 \times 10^4}{s^4 + 33.87s^3 + 397.9s^2 + 5540s + 446.9}]</td>
<td>0.0071</td>
</tr>
<tr>
<td>3</td>
<td>[\frac{1275s^2 + 234.8s + 1.993 \times 10^5}{s^3 + 33.87s^2 + 395s + 5506}]</td>
<td>0.3687</td>
</tr>
<tr>
<td>2</td>
<td>[\frac{1151s - 315.5}{s^2 + 30.25s + 94.43}]</td>
<td>81.4477</td>
</tr>
<tr>
<td>1</td>
<td>[\frac{1006}{s + 26.71}]</td>
<td>71.7159</td>
</tr>
</tbody>
</table>

To evaluate the results of the reduced order controller, we use step response and frequency response to evaluate reduced-order controllers. Simulation results using Matlab/Simulink of the step response and the frequency response are shown in Figures 3 and Figure 4.
Figure 3. Simulating results of response $h(t)$ of 6th-order controller and reduced-order controller
Figure 4. Simulating results of frequency response (bode plot) of 6th-order controller and reduced-order controller.
From the results in Table 2 and Figure 3, Figure 4, we see:

+ For the h(t) response: the h(t) response of the 5th-order and 4th-order reduction controllers perfectly matches the h(t) response of the 6th-order controller. The h(t) response of the 3rd-order reduction controller has a slight deviation from the h(t) response of the 6th-order controller; The h(t) response of the 2nd-order, 1st-order controller has a large deviation from the h(t) response of the 6th-order controller.

+ For frequency response: the frequency response of 5th, 4th, and 3rd-order controllers perfectly matches the frequency response of 6th-order controllers. The frequency response of 2nd-order and 1st-order controllers is much different from the frequency response of 6th-order controllers.

Therefore, we can use the 5th, 4th, and 3rd-order controller to replace the 6th-order controller.

To clarify the ability to preserve the dominant point when performing order reduction according to the new algorithm, we compare the reduced controller results using the new algorithm with the reduced controller results using Hankel-Norm Approximation [3], [19]. Using the Hankelmr command in Matlab software to reduce the controller hierarchy by the standard Hankel approximation (Hankel-Norm Approximation), we obtained the following results:

<table>
<thead>
<tr>
<th>Order</th>
<th>Transfer function – ( W_{cr}(s) )</th>
<th>Error ( | W_c(s) - W_{cr}(s) |_{\infty} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>( \frac{1275s^4 + 8.774 \times 10^5 s^3 + 4.388 \times 10^5 s^2 + 1.371 \times 10^5 s + 1.21 \times 10^7}{s^5 + 721.9s^4 + 2.37 \times 10^4 s^3 + 2.793 \times 10^3 s^2 + 3.812 \times 10^6 s + 3.212 \times 10^5} )</td>
<td>4.2842 \times 10^{-5}</td>
</tr>
<tr>
<td>4</td>
<td>( \frac{1275s^3 + 348.1s^2 + 1.993 \times 10^5 s + 1.774 \times 10^4}{s^4 + 33.87s^3 + 397.9s^2 + 5540s + 467} )</td>
<td>0.0067</td>
</tr>
<tr>
<td>3</td>
<td>( \frac{1161s^2 + 229.8s + 1.806 \times 10^5}{s^3 + 30.74s^2 + 395.4s + 4986} )</td>
<td>10.0489</td>
</tr>
<tr>
<td>2</td>
<td>( \frac{36.13s + 284}{s^2 + 0.9961s + 480} )</td>
<td>148.5354</td>
</tr>
<tr>
<td>1</td>
<td>( \frac{1.123 \times 10^5}{s + 2891} )</td>
<td>1.4515 \times 10^3</td>
</tr>
</tbody>
</table>

To evaluate the reduction results, we use step response and frequency response to evaluate order-reduced systems. Simulation results using Matlab/Simulink of the step response and the frequency response are shown in Figures 5 and Figure 6.
Figure 5. Simulation results of response $h(t)$ of 6th-order controller and reduced order controller using Hankel-Norm Approximation
Figure 6. Simulation results of the frequency response of 6th-order controller and reduced order controller using Hankel-Norm Approximation
From the results in Table 3 and Figure 5, Figure 6, we see:

+ For the h(t) response: the h(t) response of the 5th-order and 4th-order reduced controllers perfectly matches the h(t) response of the 6th-order controller; The h(t) response of the 3rd-order reduced controller has a slight deviation from the h(t) response of the 6th-order controller; The h(t) response of the 2nd-order and 1st-order reduced controller has a large deviation from the h(t) response of the 6th-order controller.

+ For frequency response: the frequency response of 5th and 4th-order controllers completely matches the frequency response of 6th-order controllers; The frequency response of the 3rd-order reducing controller is slightly different from the frequency response of the 6th-order controller; The frequency response of 2nd and 1st-order controllers is much different from the frequency response of 6th-order controllers.

Therefore, we can use the 5th, 4th, and 3rd-order controllers to replace the 6th-order controller.

To clarify the poles conservation of the algorithm, we represent the poles of the 6th-order controller and the 3rd-order controller as follows:

+ Poles of 6th-order controller: -681.74; -26.71; -3.5353 + 13.9156i; -3.5353 - 13.9156i; -0.09; -0.08
+ Poles of 3rd-order reduction controller obtained by Hankel-Norm Approximation: -22.9746; -3.8823 +14.211i; -3.8823 -14.211i
+ Poles of 3rd-order reduction controller obtained by the new algorithm: -26.71; -3.5353 +13.9156i; -3.5353 -13.9156i

Therefore, the dominant poles of the 6th-order controller are preserved in the 3rd-order controller using the new algorithm. With the Hankel-Norm Approximation, the dominant poles of the 6th-order controller are not preserved in the 3rd-order controller.

Comparing the controller order reduction error in Table 2 and Table 3 we see: except the errors corresponding to the 4th-order controller, the errors corresponding to the 5th, 3rd, 2nd, 1st-order controller obtained by the new algorithm are smaller than the errors corresponding to the 5th, 3rd, 2nd, 1st-order controller obtained by Hankel-Norm Approximation.

CONCLUSIONS

The paper introduces the model reduction algorithm preserving the dominant poles based on H2 dominance. The algorithm uses Shur analysis to convert the matrix A of the original system to the upper triangle, then arrange the poles according to the descending H2 dominance on the main diagonal of the upper triangle matrix A. In the above way, the algorithm both has a simple calculation and preserves the dominant poles of the original system in the order reduction system. Applying the algorithm to reduce the order of the 6th-order digital filter shows that the algorithm is capable of reducing the order of the high order digital filter. The results of applying the algorithm to reduce the order of the 6th-order controller show that the 3rd-order reduced controller both preserves the dominant points of the original controller and has a small reduction error and can use the 3rd-order reduced controller replaces the 6th-order controller. Our on-going research directions are identifying the formula for reducing order errors and applying model order reduction algorithms for sustainable control, telecommunications, and information technology.

ACKNOWLEDGMENT

This research was funded by Thai Nguyen University of Technology, No. 666, 3/2 Street, Thai Nguyen, Viet Nam.

REFERENCES


