A Rational Interpolation Method to Compute Frequency Response *

Charles Kenney, Stephen Stubberud, and Alan J. Laub
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560

Abstract

A rational interpolation method for approximating a frequency response is presented. The method is based on a product formulation of finite differences, thereby avoiding the numerical problems incurred by near-equal-valued subtraction. Also, resonant pole and zero cancellation schemes are developed that increase the accuracy and efficiency of the interpolation method. Selection techniques of interpolation points are also discussed.

1 Introduction

Consider the linear time-invariant system given by the state-space model

$$\dot{x} = Ax + Bu \tag{1}$$

$$y = Cx \tag{2}$$

where $A \in \mathbb{R}^{n_A \times n_A}$, $B \in \mathbb{R}^{n_A \times n_B}$, $C \in \mathbb{R}^{n_C \times n_A}$, and the state vector, input vector, and output vector, x, u, and y, respectively, are properly dimensioned. We shall refer to the matrices, A, B, and C, as the state coupling matrix, the input coupling matrix, and the output coupling matrix, respectively.

The frequency response of such a modeled system is defined as the Laplace transform of the input-output relationship evaluated along the $j\omega$-axis,

$$G(\omega) = C(j\omega I - A)^{-1}B \tag{3}$$

where

$$0 < \omega < \infty.$$

*This research was supported in part by the Air Force Office of Scientific Research under Contract No. AFOSR91-0240.

In this paper, a fast and reliable interpolation method to compute frequency response is presented. The basic idea of this method is based on the simple Taylor series approximation

$$G(\omega + h) = T_0 + T_1 h + \cdots + T_k h^k + E_k \tag{4}$$

but considered in the general interpolation form with $k+1$ interpolation points $h_0, h_1, \ldots, h_k$:

$$G(\omega + h) = G_0 + G_1(h - h_0) + \cdots + G_k(h - h_0)(h - h_1)\cdots(h - h_{k-1}) + E_k. \tag{5}$$

The coefficient matrices, $G_0, G_1, \ldots, G_k$ are of size $n_C \times n_B$ as is the truncation error E_k. Therefore, the cost of evaluating the matrix polynomial approximation

$$P_k(h) = G_0 + G_1(h - h_0) + \cdots + G_k(h - h_0)(h - h_1)\cdots(h - h_{k-1}) \tag{6}$$

is just $k n_B n_C$ floating-point operations (flops). The cost of computing each coefficient matrix is approximately the same as evaluating G by the method that would normally be preferred.

The polynomial interpolation scheme works well as long as ω is not near a resonant pole or zero of the system. In order to avoid this problem, we introduce methods of preliminary pole and zero cancellation. These greatly increase the accuracy of the interpolation scheme while causing only a negligible increase in the cost of computing the coefficient matrices.

We shall also discuss the implementation of this algorithm including ideas on the selection of interpolation points.

2 Existing Frequency Response Methods

2.1 Straightforward Computation

An obvious method for computing frequency response for a system modeled in state-space form is first to perform an LU decomposition in order to solve the linear system

$$(j\omega I - A)X = B, \tag{7}$$

followed by a matrix multiplication involving the solution to (7),

$$G(\omega) = CX.$$

This method does not exploit any special structure, e.g., sparse or banded, and therefore would only be used for general systems. To compute a frequency response implementing this method, for just one value of ω, approximately $\frac{1}{3}n_A^3 + \frac{1}{2}(n_B + n_C)n_A^2 + n_A n_B n_C$ flops are required. As the number of desired frequency points becomes large, the calculation of the entire frequency response becomes computationally intensive.

2.2 The Principal Vector Algorithm

In order to reduce computation cost, several methods have been developed in order to reduce the cost of solving the linear system (7) either by exploiting the structure of the state coupling matrix or by implementing a similarity transformation to put the matrix A into an exploitable form. One method of the latter variety is the Principal Vector Algorithm (PVA) [10].

The idea of the PVA is to initially transform the state coupling matrix into a Jordan Canonical Form (JCF). The algorithm uses the principal vectors to compute the JCF in a more accurate way than previous such algorithms. Let

$$A = M^{-1}JM \tag{8}$$

where J is in Jordan form. If we substitute this identity into (3), the frequency response becomes

$$\begin{aligned} G(\omega) &= CMM^{-1}(j\omega I - A)^{-1}MM^{-1}B \\ &= \tilde{C}(j\omega I - J)^{-1}\tilde{B}. \end{aligned} \tag{9}$$

The initial transformation using the PVA to compute the JCF requires only $O(n_A^3)$ flops if the state coupling matrix is not defective while $O(n_A^4)$ flops are required if the matrix A is defective. Note that this transformation only occurs once, thus the cost is only incurred once. The advantage occurs in computations at each frequency point where the cost is reduced to $O(n_A + n_A n_B n_C)$ flops in the nondefective case and $O(\frac{3}{2}n_A + n_A n_B n_C)$ flops in the defective case. So the computational saving occurs after the computation of one frequency point in the former case and n_A frequency points in the latter case.

Although this algorithm produces significant savings in the computational cost of a frequency response, it can also frequently encounter numerical instabilities. First, the JCF is extremely unstable. The slightest perturbation can change a defective matrix into a nondefective matrix. Another problem is that the similarity transform may be ill-conditioned with respect to inversion depending on the basis of eigenvectors. If they are guaranteed to form a matrix which is well-conditioned with respect to inversion as would occur if the matrix were normal, the algorithm is very effective.

2.3 The Hessenberg Method

Another algorithm which uses similarity transformations to put the state coupling matrix into an exploitable form is the Hessenberg Method [4]. This algorithm is the current standard for computing the frequency response for generic dense systems. The Hessenberg Method, as its name implies, performs an initial transformation on the state coupling matrix to reduce it to upper Hessenberg form. So in this case, we use the identity

$$A = Q^{-1}HQ,$$

where H is in upper Hessenberg form, instead of the JCF identity (8), in the frequency response (3).

As with PVA, this initial transformation is performed only once at the start of the algorithm at a cost of $O(n_A^3)$ flops. When this transformation is used, the cost of computing the frequency response at each value of ω becomes $O(n_A^2(n_B+1)+n_A n_B n_C)$ flops. Usually, $n_B \ll n_A$ so a significant reduction in computation can be realized.

Fortunately, there always exists an orthogonal transformation to reduce the state coupling matrix into an upper Hessenberg form. This prevents ill-conditioning from being introduced into the calculations by the similarity transformation as can occur with the Principal Vector Algorithm.

2.4 Sparse Systems

Many of today's large ordered systems are sparse systems. A sparse system is one whose modeling matrices have relatively few nonzero entries when compared to the total number of entries. In such cases the Hessenberg Method should not be used. Instead of maintaining sparsity, the initial transformation will create a large dense system which then must be solved. There exist many storage techniques for sparse matrices which require a significantly smaller amount of memory allocation than a full matrix of the same order would require. Also, sparse matrix algorithms have been developed to exploit sparsity in order to reduce the computational costs in comparison to their dense counterparts. (See [6], [9], and [7].) These algorithms attempt to prevent the cost of solving the linear system (7) from growing to $O(n_A^3)$ flops.

2.5 Frequency Selection Routines

The cost of computing an entire frequency response can also be reduced by eliminating needless recalculations or overcalculations in attempts to get a desired resolution in the solution. When the frequency mesh is too coarse to give the required information, usually the user recomputes the entire frequency response. Often, the response from the previously computed frequency values either is recalculated or just ignored in the new calculation. Also, many times the user creates a fine frequency point mesh across the entire frequency range. Usually, only in small subregions is the finer mesh needed. A coarser mesh would suffice over the rest of the frequency range.

In an effort to eliminate these unnecessary calculations but still give the required accuracy, so-called adaptive routines have been developed. These routines adapt the frequency point's selection to the characteristics of the system being analyzed.

One such adaptive scheme is similar in nature to the QUANC8 adaptive integration routine [1]. The basic idea is first to select the endpoints of an interval in the desired frequency region. Then the frequency responses of the two points are compared. If the difference between their magnitudes or their phases is greater than specified tolerances, the interval is divided in half. Then the three points are compared. If their differences are outside the tolerances, the subintervals are again halved. This subinterval halving continues until the tolerances are met across the entire interval or until a specified number of frequency points has been calculated. A single-input single-output variation of a method based on subinterval halving has been implemented commercially [5].

The use of *a priori* information, e.g., the locations of poles and zeros of a system, can also be used in the choice of frequency locations. More points are placed in the areas where the poles and zeros of a given system have an effect. Fewer points are placed outside these areas. Such a method is now being implemented in a linear system package [2] to automatically choose the frequency range over which the frequency response is computed as well as to determine the number of points needed to be calculated.

These adaptive schemes also can be combined to form hybrid routines. This would permit an initial placement of points with the *a priori* method and then create the frequency mesh to join the regions between the areas of the initial placement.

3 Polynomial Interpolation

In order to compute the coefficient matrices, $G_1, \ldots, G_k$, of the interpolation equation

$$P_k(h) = G_0 + G_1(h-h_0) + \cdots + G_k(h-h_0)(h-h_1)\cdots(h-h_{k-1}) \tag{10}$$

finite differences will be employed. The first-order difference is defined as

$$M[h_0, h_1] = \frac{M(h_1) - M(h_0)}{h_1 - h_0} \tag{11}$$

while higher-order differences are defined as

$$M[h_0, h_1, ..., h_n] = \frac{M[h_1, ..., h_n] - M[h_0, ..., h_{n-1}]}{h_n - h_0}. \tag{12}$$

If we let

$$M(h) = (jhI - A_0)^{-1} \tag{13}$$

where

$$A_0 = -jwI + A, \tag{14}$$

the k^{th}-order interpolation approximation can be written as

$$\begin{aligned} P_k(h) \;&=\; C(M(h_0) + M[h_0, h_1](h-h_0) + \cdots \\ &\quad + M[h_0, h_1, ..., h_k](h-h_0)(h-h_1)\cdots(h-h_{k-1}))B \end{aligned} \tag{15}$$

with the interpolation error

$$\begin{aligned} E_k &= G(\omega + h) - P_k(h) \\ &= C(M[h_0, h_1, ..., h_k, h] \prod_{i=0}^{k} (h - h_i))B. \end{aligned} \tag{16}$$

Now, for convenience, define

$$\begin{aligned} \tilde{P}_k(h) &= M(h_0) + M[h_0, h_1](h - h_0) + \cdots \\ &\quad + M[h_0, h_1, ..., h_k](h - h_0)(h - h_1) \cdots (h - h_{k-1}) \end{aligned} \tag{17}$$

and

$$\begin{aligned} \tilde{E}_k &= M(h) - \tilde{P}_k(h) \\ &= M[h_0, h_1, ..., h_k, h] \prod_{i=0}^{k} (h - h_i). \end{aligned} \tag{18}$$

Although finite differences have a certain elegance to their formulation, they can encounter numerical inaccuracies due to the subtraction of near-equal-valued quantities. An extreme example of this is the case in which all of the interpolation points are the same. In theory, the first-order difference is exactly the first derivative of M, but numerically it is useless. Fortunately, the differences of the resolvent function (13), can be expressed in matrix product forms which avoid these cancellation problems as the following theorem shows.

Theorem 1 *For the resolvent function, the matrix difference functions in (12) satisfy*

$$M[h_0, h_1, \ldots, h_m] = (-j)^m M(h_0)M(h_1) \cdots M(h_m). \tag{19}$$

Proof. Using (13)

$$\begin{aligned} M(h_1) - M(h_0) &= (jh_1 I - A_0)^{-1} - (jh_0 I - A_0)^{-1} \\ &= (jh_0 I - A_0)^{-1} \{jh_0 I - A_0 - (jh_1 I - A_0)\} (jh_1 I - A_0)^{-1} \\ &= (-j)(h_1 - h_0) M(h_0) M(h_1). \end{aligned}$$

Thus the first finite difference becomes

$$M[h_0, h_1] = -jM(h_0)M(h_1)$$

which proves (19) for $m = 1$. Now suppose that (19) is true for $m-1$. Since $M(h_0), \ldots, M(h_m)$ all commute with each other, we find that

$M[h_0, h_1, \ldots, h_m]$

$$\begin{aligned}
&= (M[h_1,\ldots,h_m] - M[h_0,\ldots,h_{m-1}])/(h_m - h_0) \\
&= (-j)^{m-1}\left(M(h_1)\cdots M(h_m) - M(h_0)\cdots M(h_{m-1})\right)/(h_m - h_0) \\
&= (-j)^{m-1}\left(M(h_1)\cdots M(h_{m-1})\right)\left(M(h_m) - M(h_0)\right)/(h_m - h_0) \\
&= (-j)^{m}\left(M(h_1)\cdots M(h_{m-1})\right)M(h_0)M(h_m) \\
&= (-j)^{m}M(h_0)M(h_1)\cdots M(h_m).
\end{aligned}$$

Thus (19) is true for m and thus, by induction, the theorem is true. □

If we now substitute the resolvent identity (19) into (17) and (18) and use the commutative property of the resolvent functions, the interpolation approximation becomes

$$\begin{aligned}
\tilde{P}_k(h) &= M(h_0) + (-j)M(h_1)M(h_0)(h-h_0) + \cdots \\
&\quad + (-j)^k M(h_k)M(h_{k-1})\cdots M(h_0)(h-h_0)\cdots(h-h_{k-1})
\end{aligned} \tag{20}$$

with the error formula

$$\begin{aligned}
\tilde{E}_k &= M(h) - \tilde{P}_k(h) \\
&= (-j)^{k+1}\prod_{i=0}^{k} M(h_i)\prod_{i=0}^{k}(h-h_i)M(h).
\end{aligned} \tag{21}$$

The next lemma gives an interpolation series for the resolvent using the original $k+1$ interpolation points and setting all of the higher-order interpolation points equal to zero. For convenience we shall use the notation $M(0) = M_0$. Note that if all of the interpolation points are set equal to zero the analysis would be that of the Taylor series.

Lemma 2 *Let $h_0,\ldots,h_k$ be given and set $h_m = 0$ for all $m > k$. For*

$$|h| < \min_{\lambda\in\Lambda(A)} |j\omega - \lambda|, \tag{22}$$

M may be expanded as

$$M(h) = \sum_{m=0}^{+\infty}(-j)^m \prod_{i=0}^{m} M(h_i)\prod_{i=0}^{m-1}(h-h_i). \tag{23}$$

Proof: Let $\ell > k$. By (21),

$$\begin{aligned}
M(h) &- \sum_{m=0}^{\ell}(-j)^m \prod_{i=0}^{m} M(h_i)\prod_{i=0}^{m-1}(h-h_i) \\
&= (-j)^{\ell+1}\prod_{i=0}^{\ell} M(h_i)\prod_{i=0}^{\ell}(h-h_i)M(h) \\
&= (-j)^{\ell+1}\prod_{i=0}^{k} M(h_i)\prod_{i=0}^{k}(h-h_i)M(h)\prod_{i=k+1}^{\ell} M_0 h \\
&= \left\{(-j)^{\ell+1}\prod_{i=0}^{k} M(h_i)\prod_{i=0}^{k}(h-h_i)M(h)\right\}(hM_0)^{\ell-k}.
\end{aligned}$$

But $(hM_0)^{\ell-k} \to 0$ as $\ell \to +\infty$ if and only if $\rho(hM_0) < 1$, which is the well-known convergence requirement for a geometric series. From the definition of M_0, we have $M_0 = (j\omega I - A)^{-1}$. Hence,

$$\begin{aligned}\rho(jhM_0) &= |h| / \min_{\lambda \in \Lambda(A)} |j\omega - \lambda| \\ &= |h|/r .\end{aligned}$$

where

$$r = \min_{\lambda \in \Lambda(A)} |j\omega - \lambda|. \tag{24}$$

□

This lemma is also important in the development of a pole and zero cancelling routine.

4 Pole and Zero Cancellation

Polynomial interpolation approximation works well unless the LTI system being analyzed has poles or zeros near the imaginary axis. Such poles and zeros are called resonant poles and resonant zeros. The following examples provide the general idea of the effect.

Example: The deleterious effect of poles and zeros can be illustrated by means of a scalar rational function example. Consider

$$f(x) = \frac{1 + 2x + 3x^2}{1 - x} = 1 + 3x + 6x^2 + \ldots \tag{25}$$

We can use a polynomial approximation to evaluate this function at various values of x. Suppose that we choose a second-order polynomial approximation:

$$\tilde{f}(x) = 1 + 3x + 6x^2.$$

If we evaluate $\tilde{f}$ for $x = 0.01$ and $x = 0.99$, we get the approximations

$$\begin{aligned}\tilde{f}(0.01) &= 1.0306, \\ &\text{and} \\ \tilde{f}(0.99) &= 9.8506,\end{aligned}$$

respectively. If we compare these to the actual values,

$$\begin{aligned}f(0.01) &= 1.0306061 \\ &\text{and} \\ f(0.99) &= 592.03,\end{aligned}$$

we can see that as we approach a pole, a much higher-order approximation is required in order to get even modest accuracy.

However, if initially we eliminate the pole before we make our calculations for values near $x = 1$, the accuracy of the method increases dramatically. Again, use a second-order approximation with pole cancellation, and we get

$$\bar{f}(x) = (1-x)\frac{1+2x+3x^2}{1-x} = (1+2x+3x^2).$$

After evaluating $\bar{f}$, we let

$$\tilde{f} = \frac{\bar{f}}{(1-x)}.$$

As can be seen in this case, the second-order interpolation is exact. In most cases, however, only a marked increase in accuracy is realized.

In order to cancel a pole in our frequency response, we write

$$M(h) = \frac{(jh + j\omega - \lambda)M(h)}{(jh + j\omega - \lambda)}$$

and then find a polynomial approximation of $(jh+j\omega-\lambda)M(h)$. Therefore, our interpolation becomes

$$G(\omega + h) = \frac{G_0 + G_1(h-h_0) + \cdots + G_k(h-h_0)\cdots(h-h_{k-1})}{(jh + j\omega - \lambda)}, \tag{26}$$

where the coefficient matrices are for a system devoid of the resonance problem. The following lemma shows how to compute the new coefficient matrices while preserving the form of the interpolating series.

Lemma 3 *Let $h_0, \ldots, h_k$ be given and set $h_m = 0$ for all $m > k$. For $|h| < r$, where r is defined in (24), define the coefficient matrices $F_m^{(n)}$ implicitly via*

$$\prod_{\ell=1}^{n}(jh + j\omega - \lambda_\ell)M(h) = \sum_{m=0}^{+\infty} F_m^{(n)} \prod_{i=0}^{m-1}(h - h_i) \ . \tag{27}$$

Then

$$F_m^{(0)} = (-j)^m \prod_{i=0}^{m} M(h_i) \ , \tag{28}$$

and

$$F_m^{(n)} = (jh_m + j\omega - \lambda_n)F_m^{(n-1)} + jF_{m-1}^{(n-1)}, \qquad m = 0, 1, \ldots \tag{29}$$

where we define $F_{-1}^{(\ell)} = 0$ for all ℓ.

Proof: Equation (28) is immediate from (23). By (27),

$$(jh + j\omega - \lambda_n) \sum_{m=0}^{+\infty} F_m^{(n-1)} \prod_{i=0}^{m-1} (h - h_i) = \sum_{m=0}^{+\infty} F_m^{(n)} \prod_{i=0}^{m-1} (h - h_i). \tag{30}$$

The assumptions that $h_m = 0$ for all $m > k$ and $|h| < r$ ensure that the series in (30) are absolutely convergent. We may thus rearrange the left-hand summation as follows:

$$\begin{aligned}
&(jh + j\omega - \lambda_n) \sum_{m=0}^{+\infty} F_m^{(n-1)} \prod_{i=0}^{m-1} (h - h_i) \\
&= \sum_{m=0}^{+\infty} (jh_m + j\omega - \lambda_n + j(h - h_m))\, F_m^{(n-1)} \prod_{i=0}^{m-1} (h - h_i) \\
&= \sum_{m=0}^{+\infty} (jh_m + j\omega - \lambda_n) F_m^{(n-1)} \prod_{i=0}^{m-1} (h - h_i) \\
&\quad + \sum_{m=0}^{+\infty} jF_m^{(n-1)} \prod_{i=0}^{m} (h - h_i) \\
&= \sum_{m=0}^{+\infty} (jh_m + j\omega - \lambda_n) F_m^{(n-1)} \prod_{i=0}^{m-1} (h - h_i) \\
&\quad + \sum_{m=0}^{+\infty} jF_{m-1}^{(n-1)} \prod_{i=0}^{m-1} (h - h_i) \\
&= \sum_{m=0}^{+\infty} \left((jh_m + j\omega - \lambda_n) F_m^{(n-1)} + jF_{m-1}^{(n-1)} \right) \prod_{i=0}^{m-1} (h - h_i).
\end{aligned}$$

Comparison with (30) gives (29). $\square$

If we need to cancel resonant zeros, we then need to find a polynomial approximation of $\frac{M(h)}{(jh+j\omega-z)}$. The following lemma illustrates how this is done.

Lemma 4 *Let $h_0, h_1, \ldots, h_k$ be given and set $h_m = 0$ for all $m > k$. For $|h| < r$, where r is defined in (24), define the coefficient matrices $D_m^{(n)}$ implicitly via*

$$M(h) = \prod_{\ell=1}^{n} (jh + j\omega - z_\ell) \sum_{m=0}^{+\infty} D_m^{(n)} \prod_{i=0}^{m-1} (h - h_i). \tag{31}$$

Then

$$D_m^{(0)} = (-j)^m \prod_{i=0}^{m} M(h_i), \tag{32}$$

and

$$D_m^{(n)} = (D_m^{(n-1)} - jD_{m-1}^{(n)})/(jh + j\omega - z_n), \qquad m = 0, 1, \ldots \tag{33}$$

where we define $D_{-1}^{(\ell)} = 0$ for all ℓ.

Proof: The proof is similar to that of the preceding lemma except that we start with the identity

$$(jh + j\omega - \lambda_n) \sum_{m=0}^{+\infty} D_m^{(n)} \prod_{i=0}^{m-1} (h - h_i) = \sum_{m=0}^{+\infty} D_m^{(n-1)} \prod_{i=0}^{m-1} (h - h_i) \tag{34}$$

and continue from there. □

5 Frequency Response Interpolation Algorithm

Step 1 Solve for X_0 in

$$(j(h_0 + \omega)I - A)X_0 = B \ , \tag{35}$$

and then solve recursively for $X_1, \ldots, X_k$ in

$$(j(h_m + \omega)I - A)X_m = -jX_{m-1} \ . \tag{36}$$

Step 2 Let $X_m^{(0)} = X_m$ and define

$$X_m^{(\ell)} = (jh_m + j\omega - \lambda_\ell)X_m^{(\ell-1)} + jX_{m-1}^{(\ell-1)} \ , \qquad 0 \le m \le k \ , \tag{37}$$

with $X_{-1}^{\ell} \equiv 0$ for $0 \le \ell \le n$.

Step 3 Let $X_m^{(n)(0)} = X_m^{(n)}$ and define

$$X_m^{(n)(\ell)} = (X_m^{(n)(\ell-1)} - jX_{m-1}^{(n)(\ell)})/(jh_m + j\omega - z_\ell) \ , \qquad 0 \le m \le k \ , \tag{38}$$

with $X_{-1}^{\ell} \equiv 0$ for $0 \le \ell \le l$.

Step 4 Form the coefficient matrices $G_0, \ldots, G_k$ via

$$G = CX_m^{(n)(l)} \ . \tag{39}$$

Step 5

$$\begin{aligned} G(\omega + h) &= (G_0 + G_1(h - h_0) + \cdots + G_k(h - h_0) \cdots (h - h_{(k-1)})) \\ &\quad \cdot \frac{\prod_{i=1}^{l}(j\omega + jh - z_i)}{\prod_{m=1}^{n}(j\omega + jh - \lambda_m)} \end{aligned} \tag{40}$$

Remark
The method used to solve the recursive linear systems in the first step of the algorithm depends on the initial structure of the LTI system being investigated. If the system has an exploitable structure such as sparsity, an algorithm that exploits that particular structure will be used. If no such structure exists, an initial similarity transformation, most likely to upper Hessenberg form, will be applied to the system.

6 Interpolation Point Selection

The placement of the interpolation points is of great importance in getting a good approximation to the frequency response. We have tested three simple methods to place the interpolation points: linear, loglinear, and Chebyshev. We have also tested placement using the *a priori* information of the pole locations.

Since frequency response is usually plotted against frequency on a log scale, the use of linearly spaced interpolation points does not usually perform well. It places too many points at the end of an interval. Both the loglinear and the Chebyshev interpolation point placements have shown promise. The loglinear placement technique usually gives an excellent approximation in the beginning to the middle of an interval, but sometimes can fail miserably at the end of an interval. The Chebyshev interpolation points (see [8]) spread the approximation error fairly evenly across the interval. However, several times the error of the Chebychev selection, although acceptable, is larger than that of the acceptable range of a loglinear interpolation of the same size. Currently, we are investigating possible hybrid techniques to exploit the best of both placement schemes.

In the cases where we have tried placing interpolation points with the knowledge of the poles and zeros of the system the results have been mixed in comparison to the two previously mentioned techniques. What has been learned is that under no circumstances should the interpolation points be the same as the resonant frequency of a resonant pole or zero. However, placing an interpolation point near the resonant frequency improves the approximation significantly.

7 Conclusion

In this paper we have presented a rational interpolation method for computing the frequency response of a system. A significant computational savings can be achieved over several of the current methods for computing a frequency response. An error analysis for the method, together with other details, can be found in [3].

The method presented in this paper avoids the numerical problem of subtraction of near equal quantities in the difference terms by using the resolvent identity of Theorem 1. Also, simple pole and zero cancellation techniques significantly increase the accuracy of the algorithm.

We are currently writing a software package to implement the algorithm in this paper. In addition, we are extending this algorithm for use with descriptor systems.

References

[1] Forsythe, G.E., M.A. Malcolm, and C.B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[2] Grace, A., A.J. Laub, J.N. Little, and C. Thompson, *Control System Toolbox, User's Guide*, The MathWorks, Natick, MA, Oct. 1990.

[3] Kenney, C.S, S.C. Stubberud, and A.J. Laub, "Frequency Response Computation Via Rational Interpolation," *Proceedings of the 1992 IEEE Symposium on Computer-Aided Control System Design*, pp. 188–195, Napa, Calif, March 1992, IEEE Control Systems Society.

[4] Laub, A.J., "Efficient Multivariable Frequency Response Computations," *IEEE Trans. Auto. Control*, AC-26 (1981), pp. 407–408.

[5] Lee, E.A., *Control Analysis Program for Linear Systems for MS-DOS Personal Computers, User Manuals*, CAPLIN Software, Redondo Beach, Calif., April 1990.

[6] Meier, W.A., "Sparse Matrix Applications in Computer-Aided Control System Design," *Proceedings of the 1992 IEEE Symposium on Computer-Aided Control System Design*, pp. 196–203, Napa, Calif, March 1992, IEEE Control Systems Society.

[7] Osterby, O. and Z. Zlatev, *Direct Methods for Sparse Matrices*, Springer-Verlag, Berlin, 1983.

[8] Rivlin, T.J, *The Chebyshev Polynomials*, John Wiley and Sons, New York, 1974.

[9] Schendel, U., *Sparse Matrices: Numerical Aspects with Applications for Scientists and Engineers*, Ellis Horwood, Chichester, 1989.

[10] Walker, R.A., *Computing the Jordan Form for Control of Dynamic Systems*, PhD thesis, Stanford University, Department of Aeronautics and Astronautics, March 1981.