Combined State Order and Model Order Formulations in the Unified Matrix Polynomial Method (UMPA)

R.J. Allemang, A.W. Phillips, D.L. Brown
Structural Dynamics Research Laboratory
School of Dynamic Systems
College of Engineering and Applied Science
University of Cincinnati
Cincinnati, OH 45221-0072 USA
Email: Randall.Allemang@UC.EDU

ABSTRACT

The unified matrix polynomial (coefficient) method (UMPA) has been used by the authors to provide a single, educational framework that encompasses most commercial and research methods used to estimate modal parameters from measured input-output data (normally frequency response functions). In past publications of this methodology, the issue of state order has not clearly been identified in the formulation of the UMPA model. State order refers to the order of the base vector that is an elementary part of the basic UMPA model and has been a part of the modal parameter estimation development since the Ibrahim Time Domain methods in the mid 1970s. The UMPA model is restated to clearly identify the role of base vector order and the relationship between base vector (state) order and polynomial model order. This relationship provides a mechanism for explaining a number of modal parameter estimation methods that have not previously been identified and helps to explain the sensitivity of different modal parameter estimation methods to noise.

Nomenclature

N_i = Number of inputs.
N_o = Number of outputs.
N_S = Short dimension size.
N_L = Long dimension size
N = Number of modal frequencies.
λ_r = S domain polynomial root.
λ_r = Complex modal frequency (rad/sec).
$\lambda_r = \sigma_r + j\ \omega_r$
σ_r = Modal damping.
ω_r = Damped natural frequency.
z_r = Z domain polynomial root.
$\{\psi_r\}$ = Modal vector.
$\{\phi_r\}$ = Pole weighted base vector (state vector).
$[A_r]$ = Residue matrix, mode r.
[I] = Identity matrix.
t_i = Discrete time (sec).
ω_i = Discrete frequency (rad/sec).
s_i = Generalized frequency variable.
$x(t_i)$ = Response function vector $(N_o \times 1)$.
$X(\omega_i)$ = Response function vector $(N_o \times 1)$.
$f(t_i)$ = Input function vector $(N_i \times 1)$
$F(\omega_i)$ = Input function vector $(N_i \times 1)$.
$[h(t_i)]$ = IRF matrix $(N_o \times N_i)$.
$[H(\omega_i)]$ = FRF matrix $(N_o \times N_i)$.
$[\alpha]$ = Denominator polynomial matrix coefficient.
$[\beta]$ = Numerator polynomial matrix coefficient.
m = Model order for denominator polynomial.
n = Model order for numerator polynomial.
v = Model order for base vector.
r = Mode number.

1. Introduction

During recent research concerning the development of autonomous modal parameter estimation methods, the authors noted that most autonomous methods utilize the concept of correlated complex modal frequency, modal vector and/or modal scaling as mechanisms for sorting out solutions that are physically and statistically relevant. One useful way to combine all of these modal parameters in a single correlation is to synthetically form an extended state vector (high order base vector) for each solution so that these modal parameters are coupled in the correlation procedure. Using higher order, pole weighted modal vectors yields a spatial-temporal vector which is a senstive correlation vector involving both the complex frequency and the (possibly scaled) modal vector. The use of the extended state vector allows similar modal parameters to be sorted out from a larger set of pole weighted vectors that represent other physical modal vectors or vectors of essentially noise. This extended state vector is the same pole weighted modal vector that has been used successfully in previous work to provide an alternate form of the consistency diagram [1-2].

Some existing modal parameter estimation algorithms already involve the state vector in the model. The primary examples of two algorithms that already use an extended base vector are the first order versions of the Eigensystem Realization Algorithm (ERA-1) and the Polyreference Frequency Domain Algorithm (PFD-1). Both of these methods also have alternate forms that utilize the conventional zeroth order base vector (ERA-2 and PFD-2) which appear to give relatively equivalent solutions. For the purpose of sorting correlated solutions in an autonomous modal parameter estimation method, state order above order one is attractive in order to get a longer (spatially) extended state vector which also may have heightened sensitivity to the complex modal frequency differences. Work has already been reported on higher state order methods in the past but this issue has not been discussed [5]. When investigating these models further, it is apparent that the Unified Matrix Polynomial Algorithm (UMPA) framework that has been utilized to describe most modal parameter estimation algorithms is not rigorous enough to describe the solutions covered by extended state order formulations without some additional clarification.

2. Background: Modal Parameter Estimation

All modern, commercial algorithms for estimating modal parameters from experimental input-output data can be developed or explained in terms of polynomial based models. For this reason, with minor implementation differences, all of these algorithms can take advantage of the consistency diagram as an aid in identifying the correct modal frequencies from the large number of poles that are found. This section quickly overviews the development of the polynomial models for both the time or frequency domains so that the model order variation options, that are involved in the consistency diagram, can be discussed. This background is detailed more fully in several references [6-7]. The algorithms that commonly use an implementation of the consistency diagram for identifying modal parameters are summarized in Table 1.

2.1 Polynomial Modal Identification Models

Rather than using a physically based mathematical model, the common characteristics of different modal parameter estimation algorithms can be more readily identified by using a matrix coefficient polynomial model. One way of understanding the basis of this model can be developed from the polynomial model used historically for the frequency response function. Note the nomenclature in the following equations regarding measured frequency ω_i versus generalized frequency s_i. Measured input and response data are always functions of measured frequency but the generalized frequency variable used in the model may be altered to improve the numerical conditioning as is done with most frequency domain methods (normalized frequency) and specifically with the polyreference least squares complex frequency (PLSCF) method (complex Z transform of frequency). The commercial implementation of the PLSCF

method is known as PolyMAX ®.

Therefore, the multiple input, multiple output (MIMO) FRF model is:

$$\sum_{k=0}^{m}\left[[\alpha_k](s_i)^k\right]\left[H(s_i)\right]=\sum_{k=0}^{n}\left[[\beta_k](s_i)^k\right][I] \tag{1}$$

Equation (1) is evaluated at many frequencies (ω_i) until all data are utilized or a sufficient overdetermination factor is achieved. Note that both positive and negative frequencies are required in order to accurately estimate conjugate modal frequencies. This allows for the coefficients of a matrix coefficient, characteristic polynomial to be identified for a given model order m. The roots of this polynomial can be used to find the modal parameters.

For the general multiple input, multiple output case:

$$\sum_{k=0}^{m}[\alpha_k]\{x(t_{i+k})\}=\sum_{k=0}^{n}[\beta_k]\{f(t_{i+k})\} \tag{2}$$

If the discussion is limited to the use of free decay or impulse response function data, the previous time domain equations can be simplified by noting that the forcing function can be assumed to be zero for all time greater than zero. If this is the case, the $[\beta_k]$ coefficients can be eliminated from the equations.

$$\sum_{k=0}^{m}[\alpha_k]\left[h(t_{i+k})\right]=0 \tag{3}$$

Additional equations can be developed by repeating Equation (3) at different time shifts into the data (t_i) until all data are utilized or a sufficient overdetermination factor is achieved. Note that at least one time shift is required in order to accurately estimate conjugate modal frequencies. This allows for the coefficients of the matrix coefficient, characteristic polynomial to be identified for a given model order m. The roots of this polynomial can be used to find the modal parameters.

The models represented by Equation (1) and Equation (3) are referred to as **Unified Matrix Polynomial Approach (UMPA)** models. Both equations yield a matrix coefficient, characteristic polynomial (the $[\alpha]$ polynomial in these models). Equation (3) corresponds to a time domain AutoRegressive-Moving-Average (ARMA(m,n)) model, or more properly an AutoRegressive with eXogenous inputs (ARX(m,n)) model, that is developed from a set of discrete time equations. Since both the frequency and time domain models are based upon functionally similar matrix coefficient, characteristic polynomials, the UMPA(m,n) terminology is used for models in both domains to reflect the order of the denominator polynomial (m) and the order of the numerator polynomial (n). In Section 2.2, this notation will be extended to UMPA(m,n,v) to reflect the order v of the base vector involved in the basic UMPA formulation.

In light of the above discussion, it is now apparent that most of the modal parameter estimation processes available could have been developed by starting from a general matrix polynomial formulation that is justifiable based upon the underlying matrix differential equation. The general matrix polynomial formulation yields essentially the same form of matrix coefficient, characteristic polynomial equation, for both time and frequency domain data.

For the frequency domain data case, this yields:

$$\left| [\alpha_m]\, s^m + [\alpha_{m-1}]\, s^{m-1} + [\alpha_{m-2}]\, s^{m-2} + \ldots\ldots\ldots + [\alpha_0] \right| = 0 \tag{4}$$

where:

$$s_r = \lambda_r \qquad \lambda_r = \sigma_r + j\ \omega_r \tag{5}$$

For the time domain data case, this yields:

$$\left| [\alpha_m]\, z^m + [\alpha_{m-1}]\, z^{m-1} + [\alpha_{m-2}]\, z^{m-2} + \ldots\ldots\ldots + [\alpha_0] \right| = 0 \tag{6}$$

where:

$$z_r = e^{\lambda_r \Delta t} \qquad \lambda_r = \sigma_r + j\ \omega_r \tag{7}$$

$$\sigma_r = \mathrm{Re}\left[\frac{\ln z_r}{\Delta t}\right] \qquad \omega_r = \mathrm{Im}\left[\frac{\ln z_r}{\Delta t}\right] \tag{8}$$

Once the matrix coefficients ($[\alpha]$) have been found, the modal frequencies (λ_r or z_r) can be found as the roots of the matrix coefficient polynomial (Equation (4) or (6)) using any one of a number of numerical techniques, normally involving the companion matrix associated with the matrix coefficient polynomial.

Therefore, the roots of the matrix characteristic equation can be found as the eigenvalues of the associated companion matrix. The companion matrix can be formulated in one of several ways. A common formulation is as follows:

$$[C]_{mN_s \times mN_s} = \begin{bmatrix} -[\alpha]_{m-1} & -[\alpha]_{m-2} & \cdots & \cdots & \cdots & -[\alpha]_1 & -[\alpha]_0 \\ [I] & [0] & \cdots & \cdots & \cdots & [0] & [0] \\ [0] & [I] & \cdots & \cdots & \cdots & [0] & [0] \\ [0] & [0] & \cdots & \cdots & \cdots & [0] & [0] \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ [0] & [0] & \cdots & \cdots & \cdots & [0] & [0] \\ [0] & [0] & \cdots & \cdots & \cdots & [0] & [0] \\ [0] & [0] & \cdots & \cdots & \cdots & [I] & [0] \end{bmatrix} \tag{9}$$

Note that the form of the companion matrix shown in Equation (9) is one of four, equivalent forms where the coefficient matrices appear in the first or last rows or columns. The form above will be used as a reference for any further discussion.

The companion matrix is used in the following eigenvalue formulation to determine the modal frequencies for the original matrix coefficient equation:

$$[C]\{X\} = \lambda\,[I]\,\{X\} \tag{10}$$

When the modal frequencies are estimated from the eigenvalue-eigenvector problem that is associated with solving this matrix coefficient polynomial equation, a unique estimate of the unscaled modal vector is identified at the same time. The length or dimension of this unscaled modal vector is equal to the dimension of the square alpha coefficients which, in general, is equal to the row dimension of the FRF data matrix in order for the matrix coefficient polynomial equation to be conformal. Normally, this row dimension associated with the FRF or IRF data matrix is assumed to be connected with the number of outputs (N_o) that were measured.

The eigenvector that is found, associated with each eigenvalue, is of length model order m times matrix coefficient size, N_i or N_o. In fact, the unique (meaningful) portion of the eigenvector is of length equal to the size of the coefficient matrices, N_i or N_o, and is repeated in the eigenvector m times.

For each repetition, the unique portion of the eigenvector is repeated, multiplied by a different complex scalar which is a successively larger, integer power of the associated modal frequency. Therefore, the eigenvectors of the companion matrix have the following form:

$$\{\phi\}_r = \begin{Bmatrix} \lambda_r^{m-1}\{\psi\}_r \\ \vdots \\ \lambda_r^{2}\{\psi\}_r \\ \lambda_r^{1}\{\psi\}_r \\ \lambda_r^{0}\{\psi\}_r \end{Bmatrix}_r \tag{11}$$

Since the data matrix (FRF or IRF) is considered to be symmetric or reciprocal, the data matrix can be transposed, switching the effective meaning of the row and column index with respect to the physical inputs and outputs.

$$[\,H(\omega_i)\,]_{N_o \times N_i} = [\,H(\omega_i)\,]^T_{N_i \times N_o} \tag{12}$$

Since many modal parameter estimation algorithms are developed on the basis of either the number of inputs (N_i) or the number of outputs (N_o), assuming that one or the other is larger based upon test method, some nomenclature conventions are required for ease of further discussion. In terms of the modal parameter estimation algorithms and the ultimate matrix coefficient, characteristic polynomial equation, it is more important to recognize whether the algorithm develops the square matrix coefficient on the basis of the larger (N_L) of N_i or N_o or the smaller (N_S) of N_i or N_o. For this reason, the terminology of *long* (larger of N_i or N_o) dimension or *short* (smaller of N_i or N_o) dimension is easier to understand without confusion. Using this approach, PTD, RFP and PLSCF are all short dimension methods where the vector found as part of the solution for poles is very small while ERA and PFD are long dimension methods where the vector found as part of the solution for poles is of full length, based upon measurement locations.

To eliminate possible confusion, in recent explanations of modal parameter estimation algorithms, the nomenclature of the number of outputs (N_o) and number of inputs (N_i) has been replaced by the length of the long dimension of the data matrix (N_L) and the length of the short dimension (N_S) regardless of

which dimension refers to the physical output or input. This means that the above reciprocity relationship can be restated as:

$$[\ H(\omega_i)\ ]_{N_L \times N_S} = [\ H(\omega_i)\ ]^T_{N_S \times N_L} \tag{13}$$

Note that the reciprocity relationships in Equation (12) and (13) are a function of the common degrees of freedom (DOFs) in the short and long dimensions. If there are no common DOFs, there are no reciprocity relationships. Nevertheless, the importance of Equation (12) and (13) comes from the idea that the dimensions of the FRF matrix can be transposed and this affects the size of the square alpha coefficients in the matrix coefficient polynomal equation.

Finally, once the modal frequencies and unscaled modal vectors are estimated via the eigenvalue-eigenvector problem, the residues (numerators) of the partial fraction model of the FRF data matrix are used to estimate the final, scaled modal vectors and modal scaling. Note that the unscaled modal vector found in the eigenvalue-eigenvector problem is available to be used as a weighting vector in the estimation of the residues and, therefore, the final scaled modal vectors and modal scaling. Also note that this weighting vector may be of length equal to the long or short dimension, depending on the modal parameter estimation algorithm being used.

$$[\ H(\omega_i)\ ]_{N_L \times N_S} = \sum_{r=1}^{N} \frac{[\ A_r\ ]_{N_L \times N_S}}{j\omega_i - \lambda_r} + \frac{[\ A_r^*\ ]_{N_L \times N_S}}{j\omega_i - \lambda_r^*} = \sum_{r=1}^{2N} \frac{[\ A_r\ ]_{N_L \times N_S}}{j\omega_i - \lambda_r} \tag{14}$$

This process means that most modern modal parameter estimation algorithms are implemented in a two stage procedure that has three steps as follows:

Stage 1, Step 1

- Load Measured Data into Over-Determined Linear Equation Form.
  - Utilize Matrix Coefficient Polynomial Based Model (Equation 1, 2 or 3).
  - Find Scalar or Matrix Coefficients ($[\alpha_k]$ and $[\beta_k]$).
  - Implement for Various Model Orders (Consistency/Stability Diagram).

Stage 1, Step 2

- Solve Matrix Coefficient Polynomial for Modal Frequencies (Equation 4 or 6).
  - Formulate Eigenvalue-Eigenvector Problem.
  - Eigenvalues Determine the Modal Frequencies (λ_r).
  - Eigenvectors Determine the Unscaled Modal Vectors ($\{\psi_r\}$) of dimension N_S or N_L.

Stage 2, Step 3

- Load Measured Data Into Over-Determined Linear Equation Form (Equation 14).
  - Determine Modal Vectors and Modal Scaling from Residues.

The most commonly used modal identification methods can be summarized as shown in Table 1. The

high order model is typically used for those cases where the system is undersampled in the spatial domain. For example, the limiting case is when only one measurement is made on the structure. For this case, the left hand side of the general linear equation corresponds to a scalar polynomial equation with the order equal to or greater than the number of desired modal frequencies. The low order model is used for those cases where the spatial information is complete. In other words, the number of physical coordinates is greater than the number of desired modal frequencies. For this case, the order of the lefthand side of the general linear equation is equal to two. The zero order model corresponds to the case where the temporal information is neglected and only the spatial information is used. These methods directly estimate the eigenvectors as a first step. In general, these methods are programmed to process data at a single temporal condition or variable. In this case, the method is essentially equivalent to the single-degree-of-freedom (SDOF) methods which have been used with frequency response functions. In others words, the zeroth order matrix polynomial model compared to the higher order matrix polynomial models is similar to the comparison between the SDOF and MDOF methods used historically in modal parameter estimation.

	Domain		Matrix Polynomial Order			Coefficients	
Algorithm	Time	Freq	Zero	Low	High	Scalar	Matrix
Complex Exponential Algorithm (CEA)	•				•	•	
Least Squares Complex Exponential (LSCE)	•				•	•	
Polyreference Time Domain (PTD)	•				•		$N_S \times N_S$
Ibrahim Time Domain (ITD)	•			•			$N_L \times N_L$
Multi-Reference Ibrahim Time Domain (MRITD)	•			•			$N_L \times N_L$
Eigensystem Realization Algorithm (ERA-1, ERA-2)	•			•			$N_L \times N_L$
Polyreference Frequency Domain (PFD-1, PFD-2)		•		•			$N_L \times N_L$
Simultaneous Frequency Domain (SFD)		•		•			$N_L \times N_L$
Multi-Reference Frequency Domain (MRFD)		•		•			$N_L \times N_L$
Rational Fraction Polynomial (RFP)		•			•	•	$N_S \times N_S$
Orthogonal Polynomial (OP)		•			•	•	$N_S \times N_S$
Polyreference Least Squares Complex Frequency (PLSCF)		•			•	•	$N_S \times N_S$
Rational Fraction Polynomial-Z Domain (RFP-Z)		•			•	•	$N_S \times N_S$
Complex Mode Indication Function (CMIF)		•	•				$N_L \times N_S$

TABLE 1. Summary of Modal Parameter Estimation Algorithms

How the different modal parameter estimation algorithms fit into the UMPA (m,n) model when the extended state order is used requires some clarification. This clarification explains the current state vector methods that utilize a base vector of model order one (ERA-1 and PFD-1) as well as another family of possible modal parameter estimation algorithms that are yet to be formally described. These potential algorithms yield higher order state vectors naturally which would be useful when an autonomous modal parameter estimation procedure is developed. This clarification is discussed in detail in the following section.

3. Clarification: Extended State (Base) Vector Order

When formulating the basic UMPA(m,n) equation for various model orders, the resulting matrix coefficient polynomial involves coefficient matricies which are sized based upon the short or long dimension of the data matrix. Once this matrix coefficient polynomial is chosen, the set of unscaled modal vectors, or base vectors is effectively chosen where the length of each vector matches this dimension.

For model orders (m) equal to one, however, this matrix model will not be able to estimate complex conjugate solutions without at least one time shift (time domain implementation) or one derivative (frequency domain implementation) when the model is formulated. When the solution is formed in this fashion, the base vector is a state vector, or what has alternately been referred to as a pole weighted modal vector, for this system. Instead of the base vector being of zeroth order as for all other UMPA cases, the base vector will be of order one. If successive time shifts or derivatives are included, the base vector can be of higher order. The form of the base vector is as follows where v is the order of the base vector:

$$\{\bar{\psi}\}_r = \begin{Bmatrix} \lambda_r^v\{\psi\}_r \\ \cdot \\ \cdot \\ \cdot \\ \lambda_r^2\{\psi\}_r \\ \lambda_r^1\{\psi\}_r \\ \lambda_r^0\{\psi\}_r \end{Bmatrix}_r \tag{15}$$

This means that the notation for the UMPA(m,n) model does not completely define the UMPA formulation. A more correct formulation would be UMPA(m,n,v). For most commercial algorithms, v is normally zero and the base vector is zeroth order. For the first order version of the Eigensystem Realization Algorithm (ERA-1) and the first order version of the Polyreference Frequency Domain (PFD-1) algorithm, v is one and the base vector is, therefore, first order. While no commercial algorithm utilizes v greater than one, there is no reason to restrict the UMPA(m,n,v) formulation to zeroth and first order base vectors. Allowing the base vector to take on higher orders lengthens the vector while adding a sensitivity to complex modal frequency that will be evaluated for correlation or consistency among all of the possible solutions. This is an extremely useful concept when developing autonomous modal parameter estimation procedures.

3.1 Impact of Base Vector Choice

While the concept of using additional time shifts or derivatives in the formulation of the matrix coefficient polynomial model, the form of the model and of the $[\alpha]$ coefficients changes. As additional time shifts or derivatives are added to the model formulation, much of the solution for the coefficients remains the same but the size of the $[\alpha]$ changes as a function of the UMPA(m,n,v) notation, specifically the v notation. This gives some insight as to what is changing and how it affects the solution from a practical viewpoint.

As an example, let's look at increasingly higher order base vector models of what is essentially an Eigensystem Realization Algorithm (ERA-1) in terms of the $[\alpha]$ coefficients that are found from the measured data. If the long dimension of the measured data space is N_L, the ERA method must involve one time shift at a minimum to achieve a solution. Therefore, the base vector in this case is of length $2 \times N_L$ and the companion matrix, assuming that the leading coefficient matrix is normalized to the identity matrix is as follow:

Case 1:

$$[C] = \begin{bmatrix} -[\alpha]_0 \end{bmatrix}_{2*N_L \times 2*N_L} \tag{16}$$

The above model would be referred to as an UMPA(1,n,1) model using the new notation. The number of unknowns found in the $[\alpha]$ coefficient would be $(2N_L)^2$.

If an additional time shift, involving additional information, is used to form the coefficient matrix, the new form is:

Case 2:

$$[C] = \left[-[\alpha]_0 \right]_{3*N_L \times 3*N_L} \tag{17}$$

The above model would be referred to as an UMPA(1,n,2) model using the new notation. The number of unknowns found in the $[\alpha]$ coefficient would be $(3N_L)^2$.

With each additional time shift, the square $[\alpha]$ coefficient matrix increases by a factor of N_L in each dimension. The base vector for each case also grows by N_L with additional weighting provided by raising the complex modal frequency (or its Z transform equivalent) to the next successive power.

This approach will yield alternate solution models that have an increasingly larger companion matrix and that also have higher ordered base vectors. The practical cost of these methods is that the memory required to solve for the $[\alpha]$ will be a function of $(v+1) \times N_L$. The solution time for solving the companion matrix will also grow. In light of modern computers, this may not be an issue and large (5th - 10th) state order problems can still be formed and solved in available desktop computing situations.

The alternate, and more natural, approach to involving additional time shifts or derivatives within the UMPA framework would be to leave the base vector at model order zero ($v = 0$) and increase the model order of the $[\alpha]$ matrix coefficient polynomial (m).

With respect to the previous example, the equivalent model to Case 1 would be the equivalent UMPA(2,n,0) model which would have a base vector of length N_L and a companion matrix of the form:

Case 3:

$$[C] = \begin{bmatrix} -[\alpha]_1 & -[\alpha]_0 \\ [I] & [0] \end{bmatrix}_{2*N_L \times 2*N_L} \tag{18}$$

The $[\alpha]$ coefficients in above model would result from exactly the same measured data as used in Case 1. The number of unknowns found in the $[\alpha]$ coefficients would be $2 \times N_L{}^2$ a much smaller memory footprint than Case 1.

If another time shift is involved in the solution, as in Case 2 in the above example, the UMPA(3,n,0) model would be formed. This model would also have a base vector of length N_L and a companion matrix of the following form:

Case 4:

$$[C] = \begin{bmatrix} -[\alpha]_2 & -[\alpha]_1 & -[\alpha]_0 \\ [I] & [0] & [0] \\ [0] & [I] & [0] \end{bmatrix}_{3*N_L \times 3*N_L} \tag{19}$$

The $[\alpha]$ coefficients in above model would result from exactly the same measured data as used in Case 2. The number of unknowns found in the $[\alpha]$ coefficients would be $3 \times N_L{}^2$ a much smaller memory footprint than Case 2.

Note that in these two alternate approaches (Case 1-2 versus Case 3-4), while the base vector model order and the polynomial model order are alternatively changed to yield the exact same number of eigenvalues and eigenvectors of the companion matrix (and the same length of eigenvectors), the cost in terms of memory footprint and somewhat in solution times based upon the number of unknowns being found will increase dramatically as base vector model order v is raised compared to the equivalent models formed by raising the order of the $[\alpha]$ polynomial m.

While the the use of higher base vector model orders is possible and theoretically sound, assuming memory and compute time is not a significant issue, the question remains as to whether this approach to model formulation is beneficial in some other way (e.g. numerically). The next section will prove that the apparently different model formulations that involve the same measured data, but different numbers of unknowns, are theoretically no different from one another.

3.2 Base Vector Order v versus $[\alpha]$ Coefficient Order m

The simplest comparison that will demonstrate the relationship between the order of the base vector and the order of the $[\alpha]$ coefficients in the UMPA formulation is to compare the ERA-1 case to the ERA-2 case. The ERA-1 algorithm is an UMPA(1,n,1) model and the ERA-2 is an UMPA(2,n,0) model. Both models generate the same size companion matrix and the same number of eigenvalue-eigenvector solutions. The following equation manipulations will show that the data in each companion matrix must be identical, and therefore yield identical eigenvalue-eigenvector results. This proof is formulated on the basis of normalizing the leading coefficient of the matrix coefficient polynomial to the identity matrix but the result is the same for any other normalization and for any other higher order polynomial algorithm when comparing the same size companion matrices formed by the two alternate approaches.

Basic Equation for ERA-1:

$$\left[[\bar{\alpha}_0]\right]\begin{bmatrix} [h(t_0)] & [h(t_1)] & [h(t_2)] & \cdots & [h(t_i)] \\ [h(t_1)] & [h(t_2)] & [h(t_3)] & \cdots & [h(t_{i+1})] \end{bmatrix} = -\begin{bmatrix} [h(t_1)] & [h(t_2)] & [h(t_3)] & \cdots & [h(t_{i+1})] \\ [h(t_2)] & [h(t_3)] & [h(t_4)] & \cdots & [h(t_{i+2})] \end{bmatrix} \quad (20)$$

Basic Equation for ERA-2:

$$\left[[\alpha_0]\,[\alpha_1]\right]\begin{bmatrix} [h(t_0)] & [h(t_1)] & [h(t_2)] & \cdots & [h(t_i)] \\ [h(t_1)] & [h(t_2)] & [h(t_3)] & \cdots & [h(t_{i+1})] \end{bmatrix} = -\begin{bmatrix} [h(t_1)] & [h(t_2)] & [h(t_3)] & \cdots. & [h(t_{i+1})] \end{bmatrix} \quad (21)$$

In order to simplify the notation of the above two equations, let:

$$\left[\overline{h(t_0)}\right] = \begin{bmatrix} [h(t_0)] & [h(t_1)] & [h(t_2)] & \cdots. & [h(t_i)] \end{bmatrix} \quad (22)$$

$$\left[\overline{h(t_1)}\right] = \begin{bmatrix} [h(t_1)] & [h(t_2)] & [h(t_3)] & \cdots. & [h(t_{i+1})] \end{bmatrix} \quad (23)$$

$$\left[\overline{h(t_2)}\right] = \begin{bmatrix} [h(t_2)] & [h(t_3)] & [h(t_4)] & \cdots. & [h(t_{i+2})] \end{bmatrix} \quad (24)$$

Rewriting Equations (20) and (21),

Basic Equation for ERA-1:

$$\left[[\bar{\alpha}_0]\right]\left[\begin{array}{c}[\overline{h(t_0)}]\\ [\overline{h(t_1)}]\end{array}\right] = -\left[\begin{array}{c}[\overline{h(t_1)}]\\ [\overline{h(t_2)}]\end{array}\right] \tag{25}$$

Basic Equation for ERA-2:

$$\left[[\alpha_0]\,[\alpha_1]\right]\left[\begin{array}{c}[\overline{h(t_0)}]\\ [\overline{h(t_1)}]\end{array}\right] = -\left[\begin{array}{c}[\overline{h(t_1)}]\end{array}\right] \tag{26}$$

Now solve for the $[\alpha]$ coefficient matrices for each case using a psuedo-inverse method.

$[\alpha]$ Coefficient Solution for ERA-1:

$$\left[[\bar{\alpha}_0]\right] = -\left[\begin{array}{c}[\overline{h(t_1)}]\\ [\overline{h(t_2)}]\end{array}\right]\left[\begin{array}{c}[\overline{h(t_0)}]\\ [\overline{h(t_1)}]\end{array}\right]^{+} \tag{27}$$

$[\alpha]$ Coefficient Solution for ERA-2:

$$\left[[\alpha_0]\,[\alpha_1]\right] = -\left[\begin{array}{c}[\overline{h(t_1)}]\end{array}\right]\left[\begin{array}{c}[\overline{h(t_0)}]\\ [\overline{h(t_1)}]\end{array}\right]^{+} \tag{28}$$

Now note, by definition, the following identity relationships:

$$\left[\begin{array}{c}[\overline{h(t_0)}]\\ [\overline{h(t_1)}]\end{array}\right]\left[\begin{array}{c}[\overline{h(t_0)}]\\ [\overline{h(t_1)}]\end{array}\right]^{+} = [\,I\,] = \left[\begin{array}{cc}[I] & [0]\\ [0] & [I]\end{array}\right] \tag{29}$$

>From the off diagonal terms in Equation (29), the following relationships must be true:

$$[\overline{h(t_0)}]\,[\overline{h(t_1)}]^{+} = [0] \qquad\qquad [\overline{h(t_1)}]\,[\overline{h(t_0)}]^{+} = [0] \tag{30}$$

Therefore, the following relationship holds:

$$\left[\begin{array}{c}[\overline{h(t_0)}]\\ [\overline{h(t_1)}]\end{array}\right]^{+} = \left[\begin{array}{c}[\overline{h(t_0)}]^{+}\,[\overline{h(t_1)}]^{+}\end{array}\right] \tag{31}$$

Substituting Equation (31) into Equations (27) and (28), the equivalence between the two approaches can be seen by inspection:

[α] Coefficient Solution for ERA-1:

$$\left[[\alpha_0]\right] = -\begin{bmatrix} [\overline{h(t_1)}] \\ [\overline{h(t_2)}] \end{bmatrix}\begin{bmatrix} [\overline{h(t_0)}] \\ [\overline{h(t_1)}] \end{bmatrix}^{+} = -\begin{bmatrix} [\overline{h(t_1)}] \\ [\overline{h(t_2)}] \end{bmatrix}\begin{bmatrix} [\overline{h(t_0)}]^{+} & [\overline{h(t_1)}]^{+} \end{bmatrix} = -\begin{bmatrix} [\overline{h(t_1)}][\overline{h(t_0)}]^{+} & [\overline{h(t_1)}][\overline{h(t_1)}]^{+} \\ [\overline{h(t_2)}][\overline{h(t_0)}]^{+} & [\overline{h(t_2)}][\overline{h(t_1)}]^{+} \end{bmatrix} \quad (32)$$

Simplifying, based upon Equations (30),

$$\left[[\alpha_0]\right] = -\begin{bmatrix} [0] & [I] \\ [\overline{h(t_2)}][\overline{h(t_0)}]^{+} & [\overline{h(t_2)}][\overline{h(t_1)}]^{+} \end{bmatrix} \quad (33)$$

[α] Coefficient Solution for ERA-2:

$$\left[[\alpha_0]\,[\alpha_1]\right] = -\begin{bmatrix} [\overline{h(t_1)}] \end{bmatrix}\begin{bmatrix} [\overline{h(t_0)}] \\ [\overline{h(t_1)}] \end{bmatrix}^{+} = -\begin{bmatrix} [\overline{h(t_1)}] \end{bmatrix}\begin{bmatrix} [\overline{h(t_0)}]^{+} & [\overline{h(t_1)}]^{+} \end{bmatrix} = -\begin{bmatrix} [\overline{h(t_2)}][\overline{h(t_0)}]^{+} & [\overline{h(t_2)}][\overline{h(t_1)}]^{+} \end{bmatrix} \quad (34)$$

Finally,

$$\left[[\alpha_0]\,[\alpha_1]\right] = -\begin{bmatrix} [\overline{h(t_2)}][\overline{h(t_0)}]^{+} & [\overline{h(t_2)}][\overline{h(t_1)}]^{+} \end{bmatrix} \quad (35)$$

By comparing Equation (35) with Equation (33), it is clear that the coefficient information in the two companion matrix cases must be exactly the same to within any numerical round-off errors in the zero and identity matrix areas of the companion matrix.

The conclusion, therefore must be that the base vector of order zero in the original UMPA formulation is complete and that no advantage is available when higher order base vectors are used to form alternate models. In fact, models that utilize a base vector model order of zero are better in terms of utilizing a minimal memory footprint and cannot be affected by numerical round-off that occurs in forming the associated companion matrix.

3.2.1 Companion Matrix Structure - Random Data

In order to demonstrate the inherent structure of the companion matrix that results from higher order base vector formulations, a simple numerical example was formed using random data. This random data was purely random data resulting from a random number generator and represents no underlying structural model. The following result was formed by using a first order Rational Fraction Polynomial model with a base vector model order of three. There are block coefficients of size 3×3 reflecting the three references in the artificial data. The model for this case would be an UMPA(1,n,3) model and it would be equivalent to an UMPA(4,n,0) model in terms of companion matrix size.

UMPA(1,n,3) Model:

$$[C] = \left[-[\alpha]_0 \right]_{12\times 12} \quad (36)$$

UMPA(4,n,0) Model:

$$[C] = \begin{bmatrix} [0] & [I] & [0] & [0] \\ [0] & [0] & [I] & [0] \\ [0] & [0] & [0] & [I] \\ -[\alpha]_0 & -[\alpha]_1 & -[\alpha]_2 & -[\alpha]_3 \end{bmatrix}_{12\times 12} \tag{37}$$

$$[C] = \begin{bmatrix} -[\alpha]_3 & -[\alpha]_2 & -[\alpha]_1 & -[\alpha]_0 \\ [I] & [0] & [0] & [0] \\ [0] & [I] & [0] & [0] \\ [0] & [0] & [I] & [0] \end{bmatrix}_{12\times 12} \tag{38}$$

The following numerical examples parallel the formulations represented by Equations 36-38.

UMPA(1,n,3) Model (Numerical Random Data Case - Equation 36):

6.49e-16	-8.28e-16	1.39e-15	1.00e+00	-4.85e-15	9.19e-16	2.33e-16	-4.78e-16	2.91e-16	7.89e-16	-1.85e-15	5.73e-16
4.27e-16	-1.18e-15	-8.33e-16	-2.52e-15	1.00e+00	7.18e-17	2.06e-16	-4.60e-16	-2.50e-16	-9.50e-16	-9.93e-16	1.27e-16
-7.52e-16	-5.59e-16	-4.57e-16	-1.74e-15	2.61e-15	1.00e+00	-2.24e-16	3.64e-17	-2.66e-16	-6.39e-16	1.00e-15	-4.67e-16
1.83e-15	6.05e-16	-3.61e-16	1.99e-16	-7.13e-16	2.77e-16	1.00e+00	1.30e-16	0.00e+00	3.10e-17	-8.09e-16	-5.31e-18
-3.12e-15	-5.36e-16	4.80e-15	-1.05e-14	6.44e-15	-4.42e-15	-9.88e-16	1.00e+00	1.58e-15	-3.44e-15	2.37e-15	-2.12e-15
1.78e-15	-1.48e-15	1.78e-15	7.52e-15	3.00e-15	-3.11e-15	9.16e-16	-6.38e-16	1.00e+00	2.81e-15	1.55e-15	-1.19e-15
-2.72e-15	-6.39e-16	1.75e-15	-4.88e-15	-9.24e-16	-3.40e-15	-1.07e-15	1.59e-16	1.09e-15	1.00e+00	-6.09e-16	-1.73e-15
-5.79e-16	1.72e-15	9.99e-16	9.38e-15	2.75e-15	-2.24e-15	-3.43e-16	8.33e-16	2.78e-16	3.37e-15	1.00e+00	-1.22e-15
3.27e-15	1.79e-15	-2.10e-15	2.67e-15	-5.80e-15	-3.03e-15	1.03e-15	3.68e-16	-4.23e-16	5.43e-16	-2.26e-15	1.00e+00
-2.43e+00	-1.77e-02	-9.69e-02	-1.64e-04	6.87e-02	-1.54e-02	-3.97e+00	-2.55e-03	-6.29e-02	4.06e-04	2.30e-02	-1.57e-02
-4.98e-03	-2.35e+00	-2.32e-02	-7.61e-02	5.01e-03	1.17e-01	4.20e-03	-3.87e+00	-2.07e-02	-2.74e-02	2.95e-03	6.00e-02
-8.98e-02	-2.93e-02	-2.34e+00	8.26e-03	-1.31e-01	-3.24e-03	-6.10e-02	-2.56e-02	-3.89e+00	1.18e-02	-6.83e-02	-3.35e-03

TABLE 2. Companion Matrix Numerical Example - Random Data - UMPA(1,n,3)

UMPA(4,n,0) Model (Numerical Random Data Case - Equation 37):

0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00
-2.43e+00	-1.77e-02	-9.69e-02	-1.64e-04	6.87e-02	-1.54e-02	-3.97e+00	-2.55e-03	-6.29e-02	4.06e-04	2.30e-02	-1.57e-02
-4.98e-03	-2.35e+00	-2.32e-02	-7.61e-02	5.01e-03	1.17e-01	4.20e-03	-3.87e+00	-2.07e-02	-2.74e-02	2.95e-03	6.00e-02
-8.98e-02	-2.93e-02	-2.34e+00	8.26e-03	-1.31e-01	-3.24e-03	-6.10e-02	-2.56e-02	-3.89e+00	1.18e-02	-6.83e-02	-3.35e-03

TABLE 3. Companion Matrix Numerical Example - Random Data - UMPA(4,n,0)

UMPA(4,n,0) Model (Numerical Random Data Case - Equation 38):

4.06e-04	2.30e-02	-1.57e-02	-3.97e+00	-2.55e-03	-6.29e-02	-1.64e-04	6.87e-02	-1.54e-02	-2.43e+00	-1.77e-02	-9.69e-02
-2.74e-02	2.95e-03	6.00e-02	4.20e-03	-3.87e+00	-2.07e-02	-7.61e-02	5.01e-03	1.17e-01	-4.98e-03	-2.35e+00	-2.32e-02
1.18e-02	-6.83e-02	-3.35e-03	-6.10e-02	-2.56e-02	-3.89e+00	8.26e-03	-1.31e-01	-3.24e-03	-8.98e-02	-2.93e-02	-2.34e+00
1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.00e+00	0.00e+00	0.00e+00	0.00e+00

TABLE 4. Companion Matrix Numerical Example - Random Data - UMPA(4,n,0)

Table 2 shows the numerical round-off error that is slightly contaminating the form of the companion matrix for the higher order base vector methods when compared to Tables 3 and 4. The information in the three matrices is exactly the same otherwise. Note that Equation 38 and Table 4 is presented simply to show the alternate forms of the companion matrix that result from manipulating the solution procedure used to determine the $[\alpha]$ coefficients. If the numerical round-off is sufficiently large enough or similar in magnitude to the data portions of the companion matrix, this could degrade the solution. Since the structure of the companion matrices is based upon forming the solution from time shifts and/or derivatives, allowing this slightly degenerate form is essentially allowing velocity to not be the derivative of displacement.

4. Summary and Future Work

In this paper, the UMPA framework has been extended to encompass the concept of a higher order base vector. Historically, most modal parameter estimation algorithms, except the first order versions of the ERA and PFD algorithms (ERA-1 and PFD-1), have utilized a base vector of zeroth order, that is a structure that corresponds directly to the vector basis of the model (either N_L or N_S.) While developing an automated technique for identifying valid modal parameters, it was observed that having higher order base vectors was advantageous in discriminating between the physical and the computational poles. Several papers were published which discussed this extended basis approach, but only as an extension of the traditional first order methods. As a result, just like ERA-1 and PFD-1, these higher order base vector methods produced a single $[\alpha_0]$ coefficient which was itself the companion matrix.

However, while developing a general autonomous modal parameter estimation methodology, it was recognized that other intermediate formulations were possible. These forms were initially believed to produce a new family of modal parameter estimation algorithms. However, in the course of investigation, the equivalence of the first-order state extended methods and the traditional higher-order methods was demonstrated. Although the thought process used to arrive at these first order methods was different from the higher-order formulation, the final matrix structural form is in fact identical within a computer numerical round-off which has nothing to do with the actual informational content as shown by the identical companion matrix structure resulting from using purely random, unrelated information. This suggests that there is no real advantage in computing these larger, extended basis coefficients. That, at best, doing so simply requires more computer memory and more computational time calculating a-priori known zero and identity quantities. In fact, it appears that if the numerical round-off on these

computed zero and identity terms becomes sufficiently large, the quality of the solution may actually be degraded.

5. References

[1] Phillips, A.W., Allemang, R.J., "Data Presentation Schemes for Selection and Identification of Modal Parameters", Proceedings, International Modal Analysis Conference (IMAC), 10 pp., 2005.

[2] Phillips, A.W., Allemang, R.J., "Additional Mechanisms for Providing Clear Stabilization (Consistency) Diagrams", Proceedings, International Conference on Noise and Vibration Engineering (ISMA), 15 pp., 2008.

[3] Phillips, A.W., Allemang, R.J., Brown, D.L., "Autonomous Modal Parameter Estimation: Methodology", Proceedings, International Modal Analysis Conference (IMAC), 22 pp., 2011.

[4] Allemang, R.J., Brown, D.L., Phillips, A.W., "Survey of Modal Techniques Applicable to Autonomous/Semi-Autonomous Parameter Identification", Proceedings, International Conference on Noise and Vibration Engineering (ISMA), 2010.

[5] Brown, D. L., Phillips, A. W., Allemang, R. J., "A First Order, Extended State Vector Expansion Approach to Experimental Modal Parameter Estimation", Proceedings, International Modal Analysis Conference (IMAC), 11 pp., 2005.

[6] Allemang, R.J., Phillips, A.W., "The Unified Matrix Polynomial Approach to Understanding Modal Parameter Estimation: An Update", Proceedings, International Conference on Noise and Vibration Engineering (ISMA), 2004.

[7] Allemang, R.J., Brown, D.L., "A Unified Matrix Polynomial Approach to Modal Identification", ***Journal of Sound and Vibration,*** Vol. 211, No. 3, pp. 301-322, April 1998.