# Dominant Modes Identification of Linear Systems via Geometrical Search

Adam Semlyen, *Life Fellow, IEEE*, Abner Ramirez, *Senior Member, IEEE*, Bjørn Gustavsen, *Fellow Member, IEEE*, Reza Iravani, *Fellow Member, IEEE*

***Abstract*—This paper presents a novel approach, based on the theory of hyperplanes, for mode identification of linear systems. The proposed approach can operate on either a set of ordinary differential equations (converted to diagonal form, if needed) or a set of partial fractions derived from a synthesized transfer function of the system under analysis. For either format, the linear system is structured to have as unknown variable a vector containing the residues. Singular value decomposition is initially used to identify an initial sparsity of the residue vector where the number of nonzero values corresponds to the pre-defined order of the dominant poles (eigenvalues) under search. An algorithm based on geometrical search of hyperplanes is used to optimize the selection of the nonzero residue locations, minimizing the residual of the zero residue hyperplanes. Finally, a recalculation of the residues is carried out by using the obtained optimal sparsity.**

***Index Terms*—Frequency-domain analysis, linear systems, reduced order systems, and singular value decomposition.**

## I. Introduction

The large number of components and apparatus being part of an electrical power network makes it desirable to have a methodology for its equivalent representation via a reduced-order model [1]-[5]. The main desirable characteristics of such a methodology are: closeness of dynamic response to that of the original complete system, robustness, and ease of computer implementation.

Also, the methodology should allow to identify dominant poles of the system to guarantee its fidelity from the controllability/observability point of view [6]. This corresponds to what is known in the literature as modal truncation or modal approximation [7]-[10]. Modal approximation basically decomposes a transfer function into a set of partial fractions retaining only those having their poles closest to the imaginary axis [8].

Regarding model order reduction techniques, two of the widely used methods for linear time invariant (LTI) systems are balanced truncation and moment matching [11]-[16]. The former method yields reduced-order stable models with uniform error bound. However, one of the main steps in such a technique is the calculation of the controllability and observability grammians, noting that for large systems such calculation is a major challenge due to its cubic computational complexity [1], [4]. The latter method has a lower computational complexity and is usually combined with numerically stable computation methods based on Krylov subspaces [13]. However, this technique lacks a uniform error bound since it matches moments at specific local frequency points only. In [16], a model order reduction technique based on dominant subspace projection is proposed; this technique provides a better approximation performance based on a new pair ($A^{-1}$, $A^{-1}B$) to form the Krylov subspace, compared to usage of the traditional pair ($A$, $B$).

Regarding modal approximation, i.e., identification of dominant poles (eigenvalues) of a transfer function, a major contribution is the subspace accelerated dominant pole algorithm (SADPA) [8]. The SADPA performs an iterative Newton-type computation of eigentriplets consisting of eigenvalue and right and left eigenvectors via a set of initial pole estimates (shifts). SADPA is enhanced by applying subspace acceleration, dominant pole selection strategy, and deflation to avoid convergence to already found eigentriplets. The SADPA technique in [8] is also extended to the multi-input multi-output (MIMO) case, i.e., SAMDP [9].

This paper proposes an alternative approach for dominant pole identification. The method is based on an initial selection of partial fractions provided by singular value decomposition (SVD) [17], followed by refinement using geometrical search of hyperplanes [18]. It is demonstrated that SVD initial selection does not always agree with dominant poles of the transfer function. Moreover, the residual given by non-chosen partial fractions should be minimum because closeness of the poles to the imaginary axis does not imply a dominant behavior of the frequency response.

The proposed approach can indistinctly be applied to modal truncation and model order reduction. Its mathematics are much simpler than those in dominant subspace projection-based methods mentioned above and curve-fitting methods, such as Vector Fitting [19], with comparable results.

Sections II-IV present the evolution of the proposed technique. Section V presents some numerical results and comparison between the different stages of Section II-IV. Section VI concludes the paper. The mathematical details of the proposed approach are shown for the single-phase case. The extension to the MIMO case is straightforward.

A. Semlyen and R. Iravani are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, M5S 3G4, Canada (e-mails: iravani@ecf.utoronto.ca, adam.semlyen@utoronto.ca).

B. Gustavsen is with SINTEF Energy Research, Trondheim N-7465 , Norway, (e-mail: bjorn.gustavsen@sintef.no).

A. Ramirez is with the Center for Research and Advanced Studies of Mexico (CINVESTAV) campus Guadalajara, Guadalajara, 45019, Mexico (e-mail: abner.ramirez@cinvestav.mx).

This is the accepted version of an article published in IEEE Transactions on Power Delivery
http://dx.doi.org/10.1109/TPWRD.2020.3038093

## II. SVD Initial Reduction

Consider the complete (full-order) LTI system of order *N*:

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

$$y = Cx\,, \tag{1b}$$

where $A \in \Re^{N\times N}$, $B \in \Re^{N\times 1}$, and $C \in \Re^{1\times N}$. Without loss of generality, the *D* term is omitted in (1b).

The state space model (1) comes either from the constitutive differential equations of the network or from the rational approximation of a transfer function. In the former case, matrix *A* contains the poles and is already assumed in diagonal form, obtained by a transformation matrix. As for the second case, the diagonal matrix *A* contains the poles provided by a rational fitting tool such as Vector Fitting [19], vector *C* has as elements the corresponding residues, and vector *B* contains "ones". In this case, the relation between the rational approximation of the transfer function and constitutive matrices/vectors in (1) is direct, as indicated in (2). The structure of system (*A*, *B*, *C*) for the multi-port case, as provided by VF, can be found in [20].

$$H(j\omega) = \sum_{k=1}^{N} \frac{q_k}{j\omega - a_k} = C(sI - A)^{-1}B\,, \tag{2}$$

where $a_k$ taken from *A* corresponds to the *k*th pole and $q_k$ taken from the product between *C* and *B* corresponds to the *k*th residue. A dominant pole in (2) is a pole $a_k$ with corresponding residue $q_k$ such that $|q_k| / |\mathrm{Re}(a_k)|$ is large compared to other poles [8].

By evaluating the central part of (2) for *F* frequencies and separating the real and imaginary parts, we obtain

$$v = Wq\,, \tag{3}$$

where $v \in \Re^{2F\times 1}$ contains the real and imaginary elements of *H*, $W \in \Re^{2F\times N}$ is a matrix of coefficients, and $q \in \Re^{N\times 1}$ contains the real and imaginary parts of residues transformed into a single residue variable, $q_k$, under a change of variable (Appendix A). This change of variable permits to halve the number of columns of matrix *W*. By applying SVD to *W* we obtain

$$v = U\Sigma V^T q\,, \tag{4a}$$

or

$$\Sigma V^T q = g\,, \tag{4b}$$

where $g = U^T v$ (*T* denotes transpose). Using only the *r* dominant singular values of *W*, system (4b) can be truncated to obtain a system of order *r* with $r < N$

$$g_r = \Sigma_r V_r^T q\,, \tag{4c}$$

where $g_r \in \Re^{r\times 1}$, $\Sigma_r \in \Re^{r\times r}$, and $V_r \in \Re^{r\times N}$.

The solution of the underdetermined system (4c) is a sparse vector *q*; this sparsity dictates the *r* partial fractions to be selected in (2) The sparsity of *q* is chosen such that the solution of the truncated system (4c) produces minimal error in the least squares sense, in this paper achieved by the use of Matlab's backslash "\" operator [21]. This choice may however not result either in an optimal selection of the partial fractions (dominant poles) in (2), or a minimum value of the residual in (4d), as shown in later sections.

$$\left\| \Sigma_{N-r} V_{N-r}^T q - g_{N-r} \right\|. \tag{4d}$$

Although closeness of the poles to the imaginary axis indicates dominance over other poles, the minimum residual criterion has to be guaranteed for dominant behavior of frequency response (which is related to location of system's resonances). This criterion can also be adopted for assessing accuracy of approximation between the original *N*th order system and the one obtained via dominant pole identification.

The number of frequencies and their distribution is still an open topic. Such number is associated to the phenomenon under analysis, the nature of the frequency-dependent network elements, among other factors. As for dynamic simulations, two points that must be satisfied are the Nyquist's criterion, involving maximum frequency, and the frequency-domain resolution to precisely account for resonance peaks. The proposed approach gives the flexibility to select the number of samples and their distribution (linearly or logarithmically spaced).

## III. Optimization by Brute Force Approach

To obtain an optimal reference solution in terms of both pole dominance and minimum residual, we first introduce a brute force method which searches through all possible solutions. Assume that *r* dominant modes from a complete system of order *N* are going to be identified. We divide (4b) as

$$\begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} [q] = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \tag{5}$$

with dimensions of $\Sigma_1$ being equal to *r*. Equation (5) constitutes two independent matrix-vector equations,

$$M_1 q = g_1\,, \tag{6a}$$

$$M_2 q = g_2\,. \tag{6b}$$

To obtain all possible sparsity patterns in *q*, a set of combinatorial numbers is generated. This set defines the *r* columns taken from $M_1$ such that *q* of dimensions *r*×1 is obtained from

$$q = M_{1r}^{-1} g_1\,. \tag{7a}$$

For illustration purposes, assume that *N* = 7 and *r* = 2; then, the number of combinations of *N* positions taken *r* at a time, is given by $n_c = N!/r!(N-r)!$ which results for this example in 21 (! denotes factorial). For all *q*'s obtained from (7a) by applying the set of combinations, the residuals $\|e_2\|$ in (7b) are calculated (considering only the corresponding *r* columns in $M_2$). The set of *r* columns taken from $M_1$, yielding the minimum residual, provides the optimal sparsity in terms of least squares error.

$$\|e_2\| = \|M_{2r}q - g_2\| . \tag{7b}$$

The brute force approach provides the optimum sparsity in the residue vector $q$ which, in turn, provides the dominant poles under search with minimum residual. However, an obvious large number of combinations results for large systems, making this approach unfeasible. A better approach is presented next.

## IV. Optimization Via Geometrical Search

### A. Hyperplanes Equations

In the last part of the SVD reduction procedure (Section II), the residue vector $q$ is forced to be sparse through the least squares solution provided by Matlab. This implies some arbitrariness which can result either in a non-optimal choice of partial fractions in (2) or in non-minimal residual in (4d). In the following we attempt to optimize the sparsity pattern of $q$.

For the complete system, (3) is exactly satisfied for any residue vector $q_o$, i.e.,

$$e_o = Wq_o - v \equiv 0 . \tag{8}$$

We choose an initial $q_o$ that corresponds to a vector of "ones". Now, we apply SVD to (3), to obtain the new residual:

$$e = \Sigma V^T q_o - g \cong 0 . \tag{9}$$

Partition of (9) gives

$$e_1 = \Sigma_1 V_1^T q_o - g_1 \cong 0 , \tag{10a}$$

$$e_2 = \Sigma_2 V_2^T q_o - g_2 \cong 0 , \tag{10b}$$

where the dimensions of the system in (10a) correspond to the $r$ desired dominant poles. Note that there exists the possibility of having very small singular values in $\Sigma_2$, e.g., $10^{-8}$; those can be set to $10^{-8}$ to avoid numerical issues and without affecting accuracy in computation of dominant poles. Besides, the number of equations in (9) should be reasonably small as we perform a combinatorial optimization, as explained next.

The objective is to calculate a new $q$, which is sparse, and satisfies:

$$e_1 = \Sigma_1 V_1^T q - g_1 = 0 , \tag{11a}$$

$$\|e_2\| = \|\Sigma_2 V_2^T q - g_2\| = \min . \tag{11b}$$

Let us write

$$q = q_o + \Delta q . \tag{12}$$

Then, substituting for $q$ from (12) into (11), taking (10) into account, gives

$$\Sigma_1 V_1^T \Delta q = 0 , \tag{13a}$$

$$\|\Sigma_2 V_2^T \Delta q\|^2 = \min . \tag{13b}$$

Expression (13a) is satisfied for any $\Delta q$ in the range of $V_2$ (since $V$ is orthonormal). Thus, we define

$$\Delta q = V_2 z , \tag{14}$$

where $z$ is arbitrary at this stage. Then, substituting for $\Delta q$ from (14) into (12), and accounting for (13b) gives

$$q = q_o + V_2 z , \tag{15}$$

$$\|\Sigma_2 z\|^2 = \min . \tag{16}$$

Using a new variable $y = \Sigma_2 z$ we obtain

$$q = q_o + V_2 (\Sigma_2)^{-1} y = q_o + By . \tag{17}$$

Expression (17) can be handled as a set of hyperplanes [18]. Note that, as $q$ is sparse, one subset of equations in (17) is equal to zero.

Sparsity of $q$ is optimized via two alternative methods described in Sections IV.B and IV.C, respectively. Both methods identify dominant poles and are accompanied by evaluation of residual criterion of the final sparsity, as defined by (4d).

### B. Geometrical Search Algorithm #1: Distance to the Origin

Assuming an initial sparsity in $q$ given by Matlab's backslash "\" operator, the geometrical search algorithm (denoted hereafter to as GS1) consists of the following steps:

- STEP 1. First, divide (17) into two groups, the lower one containing the non-zero residues, i.e., $q_{g2} \neq 0$,

$$\begin{bmatrix} q_{g1} \\ q_{g2} \end{bmatrix} = \begin{bmatrix} q_{og1} \\ q_{og2} \end{bmatrix} + \begin{bmatrix} B_{g1} \\ B_{g2} \end{bmatrix} y . \tag{18}$$

For illustration purposes, assume $N = 7$ and $r = 2$. Then, the sparsity of $[q_{g1} \; q_{g2}]^T$ is $[0\;0\;0\;0\;0\;\bullet\;\bullet]^T$ with the dots indicating non-zero values. Second, calculate the distance to the origin of the intersection of planes from group one, given by

$$y_o = \left\| -(B_{g1})^{-1} q_{og1} \right\| . \tag{19}$$

- STEP 2. Calculate distances to the origin of the planes from group two (see Fig. 1).

$$y_{g2,k} = \frac{\left| q_{g2,k} - q_{og2,k} \right|}{\left\| B_{g2,k} \right\|} . \tag{20}$$

In this example, $B_{g2,k}$ is the $k$-th row of matrix $B_{g2}$ with $k$ = 6, 7.

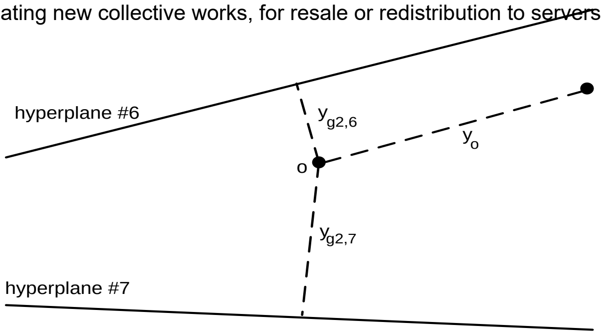


**Fig. 1**. Initial distance to the origin, $y_o$, and hyperplanes 6 and 7 for the initial iteration in the geometrical search algorithm.

- STEP 3. The plane of group two which is closest to the origin is the candidate to be part of group one (pull-in). The remaining planes keep their current positions in group two (push-out). As graphical interpretation for our example, based on Fig. 1 plane 6 is pulled-in and plane 7 is pushed-out.
- STEP 4. To find the candidate to replace the pulled-in plane in step 3 (plane 6), a sweeping procedure is applied to the planes in group one. This implies calculation of the following distances to the origin ($y_i$) (using an expression similar to (20)):

  (7, 1): planes # 2, 3, 4, 5, 6 → $y_1$
  (7, 2): planes # 1, 3, 4, 5, 6 → $y_2$
  (7, 3): planes # 1, 2, 4, 5, 6 → $y_3$
  (7, 4): planes # 1, 2, 3, 5, 6 → $y_4$
  (7, 5): planes # 1, 2, 3, 4, 6 → $y_5$

where the first number in parenthesis represents the pushed-out plane in step 3 and the second number corresponds to the plane candidate to be pushed-out.

The candidate yielding the minimum distance to the origin, if such a distance is smaller than $y_o$, is chosen. This distance becomes the new $y_o$ and we continue to step 5. Otherwise, most likely Matlab has given the right sparsity of $q$. Nevertheless, the computational algorithm is able to pull-in a different plane from group two and going back to step 3 in an iterative fashion. If none of the other planes give a distance smaller than $y_o$, the final sparsity is taken as that given by Matlab and the algorithm ends.

- STEP 5. Update the newly pushed-out planes. Assuming in the example that $y_5$ was the minimum distance in step 4, plane 5 is pushed-out, thus the new sparsity of $q$ becomes $[0\ 0\ 0\ 0 \bullet \bullet\ 0]^T$ and the updated hyperplanes are calculated as

$$q_{g2,5} = q_{og2,5} + B_{g2,5} y_5 , \qquad (21a)$$

$$q_{g2,6} = q_{og2,6} + B_{g2,6} y_5 . \qquad (21b)$$

- STEP 6. Using the new sparsity of $q$ and based on the new arrangement as in (18), go to step 2.

### *C. Geometrical Search Algorithm #2: Distance Between Two Planes*

The pull/push algorithm GS1, outlined in the previous section, may require a significant computing time for selecting the push-out plane, since that implies monitoring if a smallest norm is obtained. Below, we describe an alternative method (denoted hereafter to as GS2) which is substantially more efficient than GS1.

First, (17) is scaled by the inverse of the norm of the rows of $B$, such that the resultant matrix $P$ has all its rows with norm equal to 1. Then, by changing the sign of the resulting expression we get

$$-d + Py = 0 . \qquad (22)$$

Expression (22) represents a set of hyperplanes in the $y$-space where the elements of $d$ represent their distance to the origin. Finally, similar to (18), we split (22) into two groups

$$\begin{bmatrix} -d_{g1} \\ -d_{g2} \end{bmatrix} + \begin{bmatrix} P_{g1} \\ P_{g2} \end{bmatrix} y = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \qquad (23)$$

The GS2 algorithm pulls in, into group 1 and from group 2, the plane with the smallest distance to the origin (say plane #6). The criterion of which plane to push-out (say plane #5) is: “The union of planes 5&6 is a hyperplane of order by-one-less than either plane 5 or plane 6 alone and its distance to the origin is $D_{56}$. We choose plane 5 assuming that $D_{56}$ is maximal (compared to all other choices)”. Appendix B shows that this distance is given by

$$D_{56} = d_{56} / s_{56} . \qquad (24)$$

In particular, $s_{56}$ represents the sine of the angle between planes 5 and 6 and it is intuitively clear that the smaller the angle the larger the distance to the geometrical intersection of the two planes.

### *D. Further Residues Refinement*

The GS1 and GS2 algorithms outlined above can yield an optimized sparsity of the weighted residue vector $q$. As for dynamic simulation, a further improvement can be made by recalculating the values of the residues. This is done by keeping in a system of the form of (29) (Appendix A) the columns corresponding to the calculated sparsity. The final system is full-rank and can be accurately calculated. This process gives the final residues.

### *E. Model Order Reduction*

As mentioned in Section II, the original state space system (1) can be originated either from the network’s differential equations or from a rational approximation. Application of either GS1 or GS2 algorithm and residue refinement provides an $r$-order reduced state space system. This means, for instance, that one can utilize VF with an arbitrary approximation order $N$ and then identify an $r$-order model with mode optimal sparsity guaranteed by GS1 or GS2.

In fact, the reduced-order model has the possibility of being non-passive. If this is the case, a post-processing passivity enforcement algorithm can be applied to the reduced-order system, such as the residue perturbation method in [22], its enhanced version in [23], or via a pole selective technique [24].

## V. Numerical results

### A. RLCG Case

To introduce the proposed methods, the *RLCG* circuit of Fig. 2, with the transfer function

$$Y_{input} = \frac{1}{R_o} + \frac{1/L_1}{s + R_1/L_1} + \frac{s + G/C}{L_2 s^2 + (R_2 + L_2 G/C)s + (1 + R_2 G)/C} \quad , \quad (25)$$

is analyzed in this Section. The three poles corresponding to (25), using the values in Fig. 2, are:

| | |
|---|---|
| Real: | $-10^{5}$ |
| Complex: | $-5050 \pm 8688.93j$ |

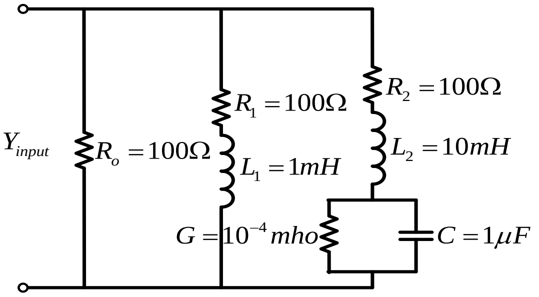


**Fig. 2**. *RLCG* circuit used as introductory example.

The magnitude of the input admittance obtained by direct evaluation of (25) and its rational approximation via VF are presented in Fig. 3. In VF, 2048 samples are used and an arbitrary order of approximation of 10 is defined, yielding $5\times10^{-18}$ as RMS approximation error. This arbitrarily chosen order can probably be selected by a common VF user, as no criterion exists so far for an optimal order. The poles by VF are arranged as complex followed by real poles (in ascending order in magnitude) and are listed in Table I where the ones corresponding to the analytical poles listed above are presented in bold-type.

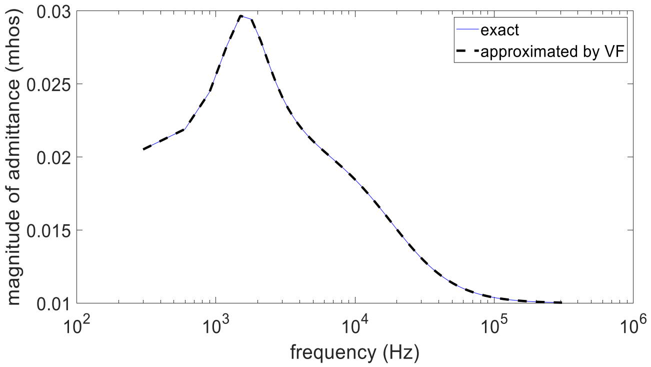


**Fig. 3**. Admittance of the *RLCG* circuit and its approximation via VF.

TABLE I

Poles given by VF Approximation

| ***Complex poles*** **(×10$^{6}$)** | ***Real poles*** **(×10$^{10}$)** |
|---|---|
| **− 0.005050 ± 0.008688*j*** | − 0.00000066 |
| − 0.516049 ± 1.633688*j* | − 0.00000999 |
| − 2.418143 ± 1.606464*j* | **− 0.00001000** |
| | − 7.97236069 |

Setting $r$ = 3, the (initial) sparsity given by applying SVD, as in Section II-A, is $\left[\bullet\ \bullet\ 0\ 0\ 0\ 0\ \bullet\ 0\ 0\ 0\right]^T$, where the dots indicate non-zero values. This sparsity indicates that the first pair of complex poles in Table I first column and the first real pole in Table I second column are selected. The pair of complex poles agree with the analytical one (–5050±8688.93*j*) but an incorrect real pole $-0.66\times10^{4}$ is selected by SVD.

The GS1 and GS2 schemes of Section IV are applied to this case study. These schemes provide the same sparsity, i.e., $\left[\bullet\ \bullet\ 0\ 0\ 0\ 0\ 0\ 0\ \bullet\ 0\right]^T$, which corresponds to the analytical poles, as can be verified in Table I. The brute force approach also provides the same sparsity as GS1 and GS2. Application of the dominant pole concept, given in Section II, to the set of 10 partial fractions given by VF, results in the dominances listed in Table II.

TABLE II

Dominance of Poles given by VF

| ***Dominance*** $\lvert q_k \rvert / \lvert \mathrm{Re}(a_k) \rvert$ |
|---|
| 0.011394 |
| 0.011394 |
| 0.000000 |
| 0.000000 |
| 0.000000 |
| 0.000000 |
| 0.000000 |
| 0.000000 |
| 0.010000 |
| 0.000000 |

The three largest values of the list above correspond to the analytical poles (bold-type in Table II) and to the sparsity given by GS1, GS2, and brute force approaches.

The residual metrics in (4d) is utilized in this paper. As for this example, the residuals given by SVD and GS1 (same for GS2 and brute force) are of 0.0702 and $2.47\times10^{-11}$, respectively.

Further experiments (not shown here) with distinct orders of approximation in VF yield sparsity corresponding to the analytical poles, as given by both GS1 and GS2. As an example, using $N$ = 14, SVD provides positions (1, 2, 11) as non-zero elements with residual of 0.072 while GS1 and GS2 approaches yield (1, 2, 13) positions corresponding to the analytical poles and residual of $2.3\times10^{-15}$. The higher orders are not reported as VF gives many close-to-zero residues. For completeness of this example, the singular values previously obtained are presented in Fig. 4.

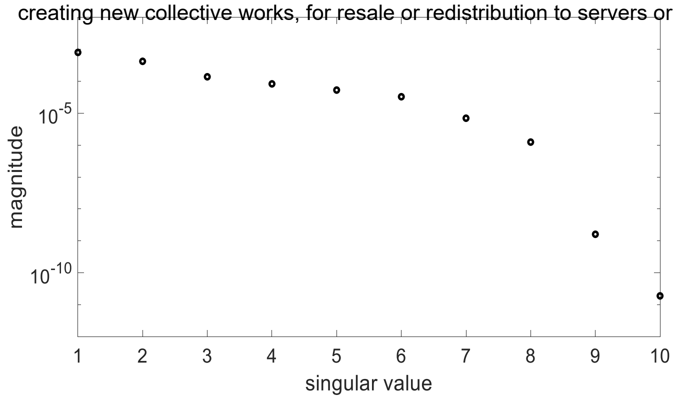


**Fig. 4**. Singular values of the 10th order approximation of *RLGC* circuit.

### B. Single-Phase System

This case study involves a two-dimensional network for which a typical cell is depicted in Fig. 5 and extends to the right and downward. In that network, each of the horizontal and vertical branches consists of series *R* and *L* elements; at each node, a *CG* load is connected. The input is the voltage and the output is the current shown at the left-top corner of the grid. The seven ordinary differential equations (ODEs) for a single cell are given as an illustrative example in Appendix C.

The total number of states for a 10-grid network (extending 10 cells to the right and 10 cells downward) is 279 and is initially reduced to 4 states (resulting in 4 complex pairs of poles). The state space system is diagonalized and expressed into partial fractions form, as in (2). As the preliminary data treatment, partial fractions with very small residues (in absolute value < $10^{-10}$) are removed, yielding a 100 order from the original 279 poles. The 100-order is considered as the full-order system.

The magnitudes and phases of the full- and reduced-order transfer functions and the absolute error among them are presented in Fig. 6; these are obtained either with SVD, GS1, or GS2 approaches. As for the 50-order system (recalling that real and imaginary parts of residues are compacted under a change of variable), the resultant sparsity by SVD means that the elements of *q*, (15, 16, 22, 50), are non-zero values. The same sparsity is obtained via the brute force approach and by both the geometrical search algorithms GS1 and GS2. The residual in (4d) for all methods is of 0.0124.

Table III lists the cpu-times of the four above mentioned approaches used for this example. Table III indicates that SVD and GS2 consume about the same cpu-time. The brute force approach requires sweeping among the $n_c$ = 230,300 combinations to obtain the optimal combination.

As the second experiment, the 10-grid network of order 100 (partial fractions with very small residues already removed) is reduced to 18 states (18 complex pairs of poles). Verification with the brute force approach is not practically possible owing to the 1.8053×$10^{13}$ combinations. Fig. 7 presents magnitude and absolute error of the 18-order approximation given by GS1. As a reference, Fig. 8 presents the pole map as given by the three methods.

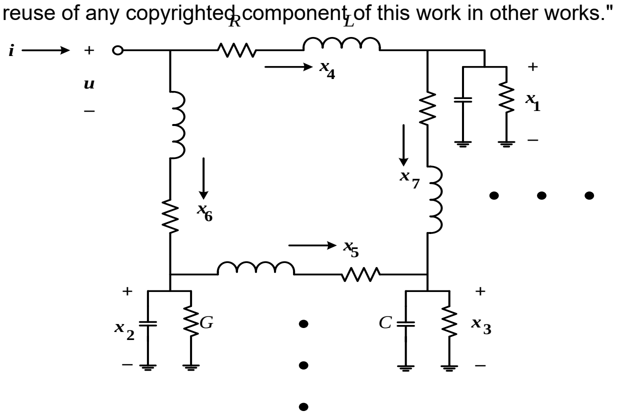
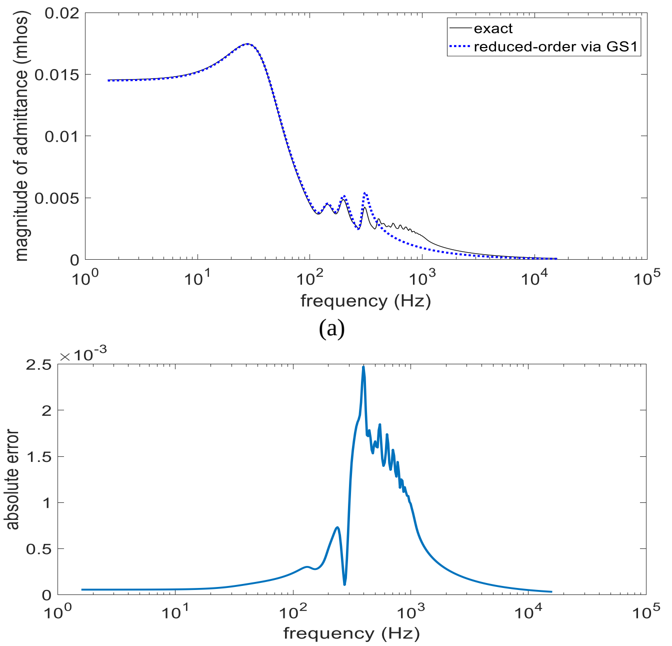


**Fig. 5**. Typical cell for the 2-D single-phase network.



(a)

(b)

**Fig. 6**. Transfer function and 4-order approximation, (a) magnitude and (b) absolute error.

TABLE III
CPU-TIMES

| | SVD | Brute force | GS1 | GS2 |
|---|---|---|---|---|
| Cpu-time (s) | 0.002 | 2.6 | 2.0 | 0.002 |

The sparsities (non-zero positions in vector *q* of expression (3)) given by SVD, GS1, and GS2 are given in Table IV. It is observed from Table IV that sparsities among the three methods are partly distinct. To assess their accuracy, residuals are calculated via (4d) and listed in Table V. The GS1 (distance to the origin) approach provides the smallest residual, although cpu-time is the largest, as seen in Table VI. GS2 yields a cpu-time comparable with SVD and with comparable residual as GS1.

TABLE IV
SPARSITIES

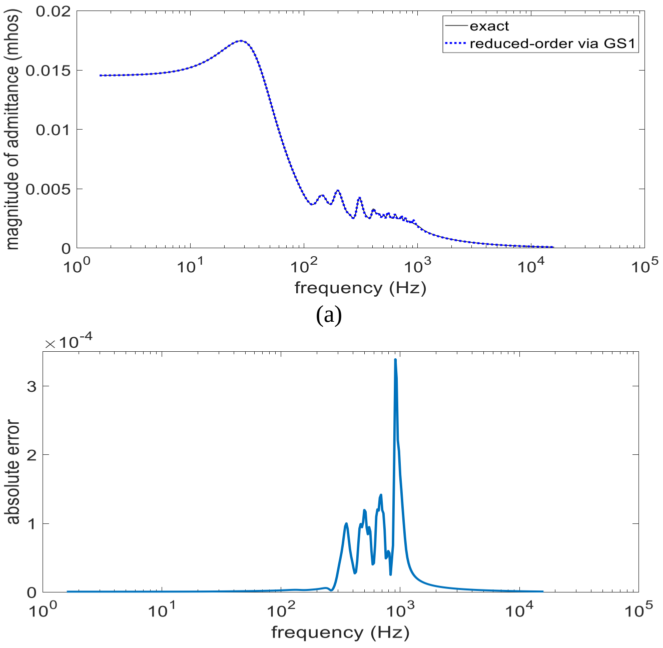
| | Sparsity |
|---|---|
| **SVD** | 14, 15, 16, 17, 19, 20, 22, 23, 26, 27, 29, 30, 35, 37, 38, 41, 45, 50 |
| **GS1** | 12, 15, 16, 17, 19, 20, 22, 23, 26, 29, 30, 34, 35, 37, 38, 41, 46, 50 |
| **GS2** | 14, 15, 16, 17, 19, 20, 22, 23, 25, 26, 29, 30, 35, 37, 38, 41, 45, 50 |

TABLE V
RESIDUALS

| | SVD | Brute force | GS1 | GS2 |
|---|---|---|---|---|
| **Residual** | 1.064×10$^{-3}$ | --- | 8.708×10$^{-4}$ | 1.033×10$^{-3}$ |

TABLE VI
CPU-TIMES

| | SVD | Brute force | GS1 | GS2 |
|---|---|---|---|---|
| **Cpu-time (s)** | 0.0038 | --- | 2.11 | 0.004 |



(a)

(b)

**Fig. 7**. Transfer function and 18-order approximation, (a) magnitude and (b) absolute error.

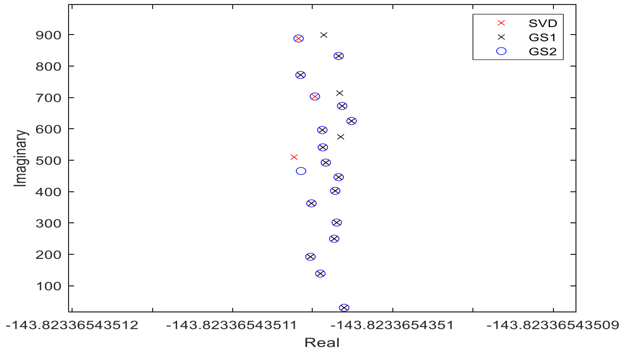


**Fig. 8**. Poles locations for 18-order approximation.

## C. *Three-Phase 469-Bus System*

This Section shows the capability of the methodologies based on hyperplanes theory to identify dominant poles of frequency dependent network equivalents (FDNEs). The system under analysis, taken from [25], is presented in Fig. 9. It consists of 469 nodes, 58 distributed parameters frequency-dependent overhead lines, 280 RLC branches, 105 ideal transformer units, and 16 generators modeled as constant voltage sources behind RL impedances. The FDNE for the external zone (region outside the study zone) is calculated with 400 logarithmically-spaced frequency samples, resulting in a 15 × 15 × 400 input admittance matrix *Y*.

An order of approximation of 130 is utilized in VF, resulting in a state space system of dimensions 1950 (three phases, five ports) with an RMS error of 2.8×10$^{-5}$. It is noted that passivity enforcement is applied to the fitted admittance.

Tables VII and VIII list the residuals and cpu-times, respectively, given by SVD, GS1, and GS2 approaches. An order of 40 is originally specified in SVD. For this specific case, GS1 yields the smallest residual but consumes the largest cpu-time (1 iteration required). On the other hand, GS2 (2 iterations required) yields smaller residual than SVD while comparable cpu-time.

Figure 10(a) presents the dominant poles identified by GS2 only; Table IX lists their corresponding frequencies.

To continue the application of GS-based techniques to model order reduction, a residue recalculation is applied based on the identified dominant poles via either the linear formulations (29) or (31) of Appendix A. The frequency domain spectra of elements (1, 1) and (1, 2) of the original input admittance matrix and those by the reduced-order via GS2 are presented in Figs. 10(b) and 10(c). The three identification methods provide 14 real poles and 26 pairs of complex poles. As for the GS2, the frequencies listed in Table IX agree with the main spikes of the frequency domain spectra of Figs. 10(b) and 10(c) (note that due to space limitations, not all elements are presented in this paper).

TABLE VII
RESIDUALS

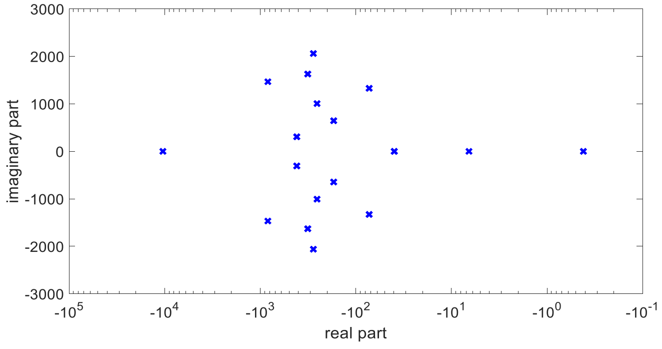
| | SVD | Brute force | GS1 | GS2 |
|---|---|---|---|---|
| **Residual** | 0.1944 | --- | 0.1929 | 0.1939 |

TABLE VIII
CPU-TIMES

| | SVD | Brute force | GS1 | GS2 |
|---|---|---|---|---|
| **Cpu-time (s)** | 2.32 | --- | 1470.00 | 3.93 |

TABLE IX
FREQUENCIES OF COMPLEX POLES (HZ) BY GS2

| Frequency (Hz) |
|---|
| 307.15 |
| 645.24 |
| 1006.44 |
| 1328.05 |
| 1466.87 |
| 1629.96 |
| 2060.67 |



(a)

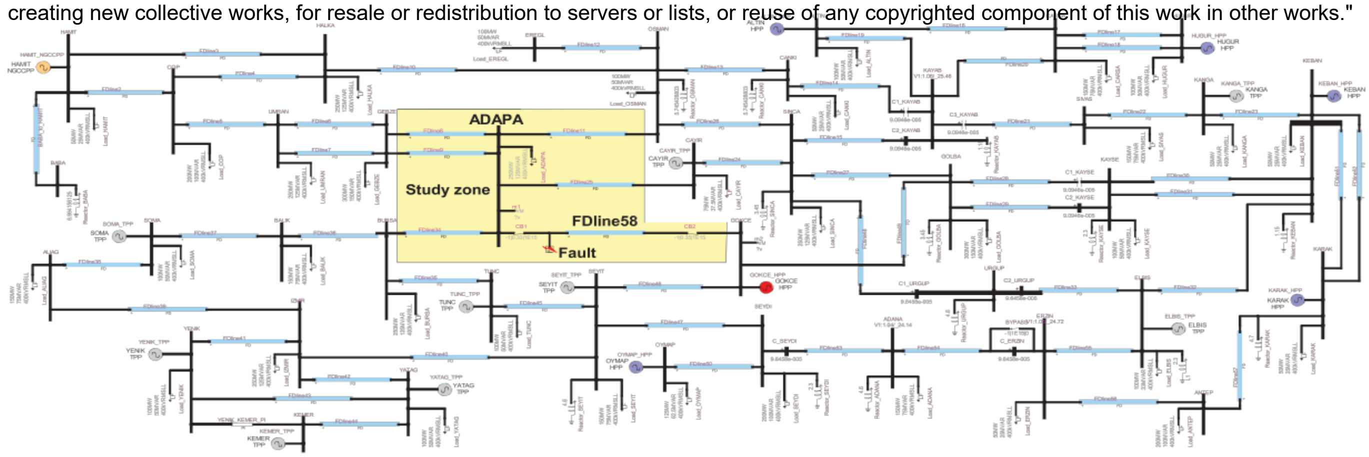
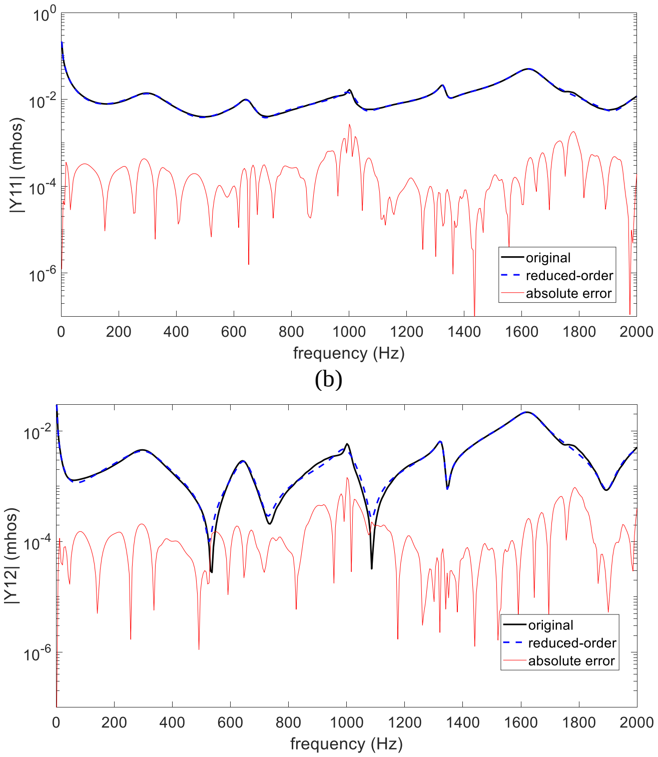


**Fig. 9**. 469-bus three-phase system, taken from [25].



(b)

(c)

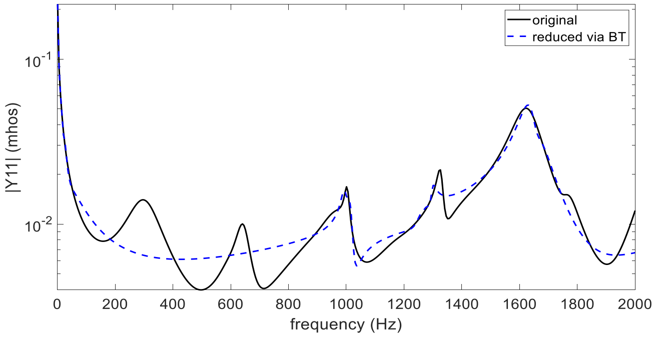
**Fig. 10**. 469-bus three-phase system (a) pole map and (b) frequency domain spectra of original and reduced-order systems, element (1, 1) and (c) element (1, 2).

### *D. Comparison with Balanced Truncation Method*

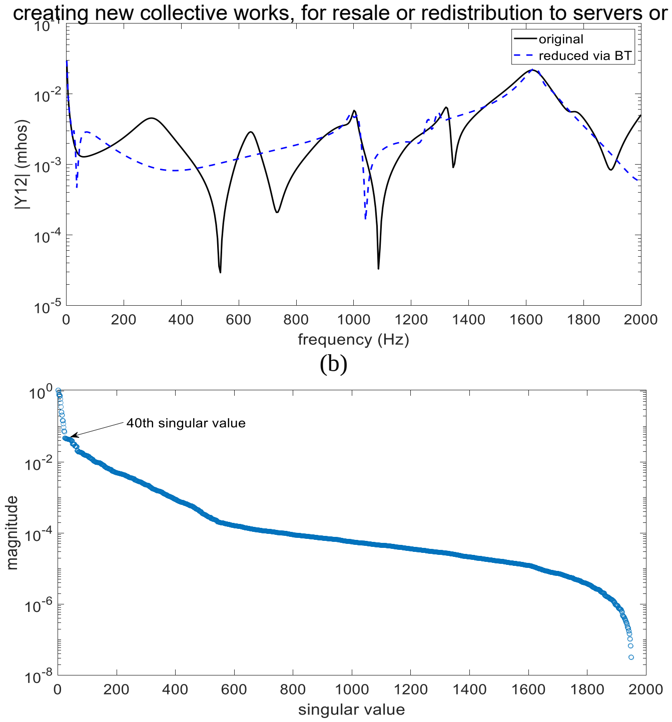
In the previous Section, dominant poles identification is taken to the limit by defining a very small order, i.e., 40, from a 1950-order system. For comparison purposes, the balanced truncation method, chosen due to its uniform error bound and well-defined approximation errors [11], is applied to the network in Fig. 9. Details of implementation of the balanced truncation method in Matlab environment can be found in [26].

The approximation results, including residue refinement, by the balanced truncation method (labeled as BT) corresponding to Figs. 10(b) and 10(c) are presented in Figs. 11(a) and 11(b), respectively (absolute errors not shown). The cpu-time taken by balanced truncation is of 31 s. It is observed from Figs. 11(a) and 11(b) a poor agreement between the original spectra and their approximation. This poor agreement can be attributed to the very small order chosen for reduction, as confirmed by the singular values provided by the balanced truncation technique and presented in Fig. 11(c). The first and 40th Hankel singular values, $\sigma$, have magnitudes of 1.058 and 0.043, respectively, giving a ratio of 0.041. The closeness between the full and truncated systems, which can be evaluated with (26), is equal to 0.085 which is highly correlated with the ratio of 0.041 [11], [12]. Note that if closer approximation is desired, we must truncate to a higher number of singular values, as shown in Fig. 11(c), increasing the order of approximation. Finally, the eigenvalues of the state matrix corresponding to reduced-order system are listed in Table X, showing that not all principal spikes in the spectra are identified and confirmed by the approximations in Figs. 11(a) and 11(b).

$$E_r = \left( \sum_{i=r+1}^{N} \sigma_i^2 \Bigg/ \sum_{i=1}^{r} \sigma_i^2 \right)^{1/2} . \qquad (26)$$



(a)



(b)

(c)

**Fig. 11**. Balanced truncation method applied to 469-bus three-phase system (a) frequency domain spectra of original and reduced-order systems, element (1, 1) and (b) element (1, 2), and (c) Hankel singular values.

TABLE X

FREQUENCIES OF COMPLEX POLES (HZ) BY BALANCED TRUNCATION

| **Frequency (Hz)** |
|---|
| 8.71 |
| 995.13 |
| 1014.53 |
| 1264.83 |
| 1297.68 |
| 1644.87 |
| 1653.85 |

## VI. CONCLUSIONS

The topic of dominant pole identification is addressed in this paper by singular value decomposition followed by two alternative methods based on geometrical search of hyperplanes. It is observed that the two techniques (GS1, GS2) based on hyperplanes provide an optimized pole identification in cases where SVD does not. The computation time of GS2 is similar to that of the initial SVD calculation, thereby offering an optimization refinement at little additional cost.

## APPENDIX A

Consider the transfer function in (2) with *m* complex poles and *n* real poles,

$$H(j\omega)=\sum_{k=1}^{m}\frac{x_k}{j\omega-a_k}+\sum_{k=1}^{n}\frac{y_k}{j\omega-b_k}\,. \qquad (27)$$

Evaluating (27) for *F* frequencies,

$$\begin{bmatrix} \frac{1}{j\omega_1-a_1} & \cdots & \frac{1}{j\omega_1-a_m} & \frac{1}{j\omega_1-p_1} & \cdots & \frac{1}{j\omega_1-p_n} \\ & \vdots & & & \vdots & \\ \frac{1}{j\omega_F-a_1} & \cdots & \frac{1}{j\omega_F-a_m} & \frac{1}{j\omega_F-p_1} & \cdots & \frac{1}{j\omega_F-p_n} \end{bmatrix}\begin{bmatrix} x_1\\ \vdots\\ x_m\\ y_1\\ \vdots\\ y_n \end{bmatrix}=\begin{bmatrix} H(j\omega_1)\\ \vdots\\ H(j\omega_F) \end{bmatrix}. \qquad (28)$$

By separating real and imaginary parts in (28) we have,

$$\begin{bmatrix} A_r & A_i & \cdots & C_r & \cdots \\ & \vdots & & \vdots & \\ B_r & B_i & \cdots & C_i & \cdots \\ & \vdots & & \vdots & \end{bmatrix}\begin{bmatrix} x_r\\ x_i\\ \vdots\\ y_1\\ \vdots \end{bmatrix}=\begin{bmatrix} H_r(j\omega_1)\\ \vdots\\ H_r(j\omega_F)\\ H_i(j\omega_1)\\ \vdots\\ H_i(j\omega_F) \end{bmatrix}, \qquad (29)$$

where:

$$A_r=-a_r\left[\frac{1}{a_r^2+(\omega_1-a_i)^2}+\frac{1}{a_r^2+(\omega_1+a_i)^2}\right],$$

$$A_i=\frac{\omega_1-a_i}{a_r^2+(\omega_1-a_i)^2}-\frac{\omega_1+a_i}{a_r^2+(\omega_1+a_i)^2}\,,$$

$$B_r=-\frac{\omega_1-a_i}{a_r^2+(\omega_1-a_i)^2}-\frac{\omega_1+a_i}{a_r^2+(\omega_1+a_i)^2}\,,$$

$$B_i=-a_r\left[\frac{1}{a_r^2+(\omega_1-a_i)^2}-\frac{1}{a_r^2+(\omega_1+a_i)^2}\right],$$

$$C_r=-\frac{b}{p^2+\omega_1^2}\,,\qquad C_i=-\frac{\omega_1}{p^2+\omega_1^2}\,.$$

In (29), subscripts *r* and *i* represent the real and imaginary parts, respectively. The same notation holds for poles, residues, and *H*.

To reduce dimensions in (29), a change of variable is applied to both real and imaginary parts of the complex residues as follows:

$$\begin{bmatrix} A_r & A_i \end{bmatrix}\begin{bmatrix} x_r\\ x_i \end{bmatrix}=\begin{bmatrix} A_r & A_i \end{bmatrix}w\begin{bmatrix} u_r\\ u_i \end{bmatrix}=(A_r u_r+A_i u_i)w\,. \qquad (30)$$

A similar change of variable is applied to real residues. Thus, we finally obtain

$$\begin{bmatrix} A_r u_r+A_i u_i & \cdots & C_r v & \cdots \\ \vdots & & \vdots & \\ B_r u_r+B_i u_i & \cdots & C_i v & \cdots \\ \vdots & & \vdots & \end{bmatrix}\begin{bmatrix} w_1\\ \vdots\\ w_{m/2}\\ w_1'\\ \vdots\\ w_n' \end{bmatrix}=\begin{bmatrix} H_r(j\omega_1)\\ \vdots\\ H_r(j\omega_F)\\ H_i(j\omega_1)\\ \vdots\\ H_i(j\omega_F) \end{bmatrix}. \qquad (31)$$

Note that (31) has half columns compared with (29). Finally, (31) becomes (3).

## APPENDIX B

Since *P* is a normalized version of *B*, we have:

$$P_{g1}^T P_{g2}=P_{g2}^T P_{g1}=\cos(\alpha)=c_{56}\,, \qquad (32)$$

so that

$$\sin(\alpha)=s_{56}=\sqrt{1-c_{56}}\,. \qquad (33)$$

It can be shown that

$$d_{56} = \sqrt{d_5^2 + 2c_{56}d_5d_6 + d_6^2}\ , \tag{34}$$

where $d_5$ and $d_6$ are calculated with (20).

## Appendix C

A typical cell of the network is shown in Fig. 5. This cell can be repeated to the right and below to form a larger network. The system of ODEs obtained for one cell is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \end{bmatrix} = \begin{bmatrix} -G/C & 0 & 0 & 1/C & 0 & 0 & -1/C \\ 0 & -G/C & 0 & 0 & -1/C & 1/C & 0 \\ 0 & 0 & -G/C & 0 & 1/C & 0 & 1/C \\ -1/L & 0 & 0 & -R/L & 0 & 0 & 0 \\ 0 & 1/L & -1/L & 0 & -R/L & 0 & 0 \\ 0 & -1/L & 0 & 0 & 0 & -R/L & 0 \\ 1/L & 0 & -1/L & 0 & 0 & 0 & -R/L \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/L \\ 0 \\ 1/L \\ 0 \end{bmatrix} u\ . \tag{35}$$

## Acknowledgment

This work was originated based on discussions with Adam Semlyen, Professor Emeritus at the University of Toronto. The co-authors would like to dedicate this work to Prof. Semlyen in appreciation of his contributions and mentorship in the field of power systems analysis.

## References

[1] A.C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, series on Advances in Design and Control, SIAM, USA, 2005.

[2] U.D. Annakkage, N.K.C. Nair, Y. Liang, A.M. Gole, V. Dinavahi, B. Gustavsen, T. Noda, H. Ghasemi, A. Monti, M. Matar, R. Iravani, and J.A. Martinez, “Dynamic system equivalents: A survey of available techniques,” *IEEE Trans. Power Del.*, vol. 27, no. 1, pp. 411-420, Jan. 2012.

[3] P. Benner, V. Mehrmann and D.C. Sorensen. Dimension Reduction of Large-Scale Systems. *Lecture Notes in Computational Science and Engineering*, vol. 45, Springer-Verlag, Jun. 2005.

[4] W.H.A. Schilders, H.A. van der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, vol. 13, *Springer*, Berlin, 2008.

[5] P. Vorobev, P. Huang, M. Al Hosani, J. L. Kirtley and K. Turitsyn, “High-fidelity model order reduction for microgrids stability assessment,” *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 874-887, Jan. 2018.

[6] J.R. Smith, J.F. Hauer, and D.J. Trudnowski, “Transfer function identification in power system applications,” *IEEE Trans. on Power Systems*, vol. 8, no. 3, pp. 1282-1290, Aug. 1993.

[7] N. Martins and P.E.M. Quintao, “Computing dominant poles of power system multivariable transfer functions,” *IEEE Trans. on Power Systems*, vol. 18, no. 1, pp. 152-159, Feb. 2003.

[8] J. Rommes and N. Martins, “Efficient computation of transfer function dominant poles using subspace acceleration,” *IEEE Trans. on Power Systems*, vol. 21, no. 3, pp. 1218-1226, Aug. 2006.

[9] J. Rommes and N. Martins, “Efficient computation of multivariable transfer function dominant poles using subspace acceleration,” *IEEE Trans. on Power Systems*, vol. 21, no. 4, pp. 1471-1483, Nov. 2006.

[10] A. Lopez and A.R. Messina, “An optimal modal approximation method for model reduction of linear power system models,” *Int. J. Electrical Power and Energy Systems*, vol. 44, no. 1, pp. 293-300, Jan. 2013.

[11] B. Moore, “Principal component analysis in linear systems: Controllability, observability, and model reduction,” *IEEE Trans. Autom. Control*, vol. AC-26, no. 1, pp. 17-32, Feb. 1981.

[12] A. Ramirez, A. Mehrizi-Sani, D. Hussein, M. Matar, M. Abdel-Rahman, J. Jesus Chavez, A. Davoudi, and S. Kamalasadan, “Application of balanced realizations for model-order reduction of dynamic power system equivalents,” *IEEE Trans. Power Del.*, vol. 31, no. 5, pp. 2304-2312, Oct. 2016.

[13] G. Troullinos and J.F. Dorsey, “Application of balanced realizations to power system equivalents,” *IEEE Trans. Autom. Control*, vol. AC-30, no. 4, pp. 414-416, Apr. 1985.

[14] J. Liesen and Z. Strakos, *Krylov Subspace Methods: Principles and Analysis*. Oxford University Press, 2013.

[15] Z. Zhu, G. Geng, and Q. Jiang, “Power system dynamic model reduction based on extended Krylov subspace method*,” IEEE Trans. Power Syst*., vol. 31, no. 6, pp. 4483–4494, January 2016.

[16] G. Shi and C.J.R. Shi, “Model-Order Reduction by Dominant Subspace Projection: Error Bound, Subspace Computation, and Circuit Applications”, *IEEE Trans. on Circuits and Systems* –I: Regular Papers, vol. 52, no. 5, pp. 975-993, May 2005.

[17] V.C. Klema and A.J. Laub, “The singular value decomposition: Its computation and some applications,” *IEEE Trans. on Automatic Control*, vol. AC-25, no. 2, pp. 164-176, April 1980.

[18] A. Bjorck, *Numerical Methods in Matrix Computations*, Springer, Switzerland, 2015.

[19] B. Gustavsen and A. Semlyen, “Rational approximation of frequency domain responses by vector fitting,” *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1052-1061, July 1999.

[20] B. Gustavsen and A. Semlyen, “A robust approach for system identification in the frequency domain,” *IEEE Transactions on Power Delivery*, vol. 19, no. 3, pp. 1167-1173, July 2004.

[21] The MathWorks, Inc. Matlab v. R2019b, USA.

[22] B. Gustavsen and A. Semlyen, “Enforcing passivity for admittance matrices approximated by rational functions”, *IEEE Trans. on Power Systems*, vol. 16, no. 1, pp. 97-104, Feb. 2001.

[23] B. Gustavsen, “Passivity enforcement by residue perturbation via constrained non-negative least squares,” *IEEE Trans. on Power Delivery*, early access, doi: 10.1109/TPWRD.2020.3026385

[24] E. Medina, A. Ramirez, J. Morales, and K. Sheshyekani, “Passivity enforcement of FDNEs via perturbation of singularity test matrix,” *IEEE Trans. on Power Delivery*, vol. 35, no. 4, pp. 1648-1655, Aug. 2020.

[25] J. Morales, J. Mahseredjian, A. Ramirez, K. Sheshyekani and I. Kocar, “A Loewner/MPM–VF combined rational fitting approach,” *IEEE Trans. on Power Delivery*, vol. 35, no. 2, pp. 802-802, Apr. 2020.

[26] A. Ramirez, Vector fitting-based calculation of frequency-dependent network equivalents by frequency partitioning and model-order reduction,” *IEEE Trans. Power Del.*, vol. 24, no. 1, pp. 410–415, Jan. 2009.