5-2017

Applications of Krylov Subspace and Balanced Truncation Model Order Reduction in Power Systems

Sebastian Emanuel Garrido
University of Arkansas, Fayetteville

Applications of Krylov Subspace and Balanced Truncation Model Order Reduction in Power Systems

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering

by

Sebastian Emanuel Garrido Sanchez
Universidad Tecnológica de Panamá
Bachelor of Science in Electrical Engineering, 2013

May 2017
University of Arkansas

This thesis is approved for recommendation of the Graduate Council.

Dr. Roy McCann
Thesis Director

Dr. Juan Carlos Balda
Committee Member

Dr. Simon Ang
Committee Member

ABSTRACT

Dynamic representations of power systems usually result in the order of hundreds or even thousands of buses. Therefore, reduction of these dynamic representations is convenient. Two applications of model order reduction in power systems are discussed in this thesis. First, Krylov subspace-based method is applied to the IEEE-123 Node Test Feeder in the context of distribution-level power systems simulation. Second, a Balanced Truncation-based model reduction is implemented in the 3-Machine 9-Bus system for designing a power system controller in the context of generation- and transmission-level power systems.

First, for the IEEE-123 Node Test Feeder, a two-sided Arnoldi algorithm is proposed to compute the basis of the Krylov subspace-based model reduction. The two-sided Arnoldi algorithm was found to decrease the deviation between the reduced and full-order model.

Second, for the 3-Machine 9-Bus, a linear quadratic regulator (LQR) controller is designed based on the reduced model. The selection method of $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices is adopted from [1]. The resulting controller is shown to damp the oscillations of the open-loop system.

ACKNOWLEDGEMENTS

I want to acknowledge God all mighty who stayed with me in the most challenging times. I also want to thank my family and my wife Doris who always gave me support.

I want to specially acknowledge the time and effort my academic advisor Dr. Roy McCann dedicated in guiding up to the accomplishment of this work.

I am extremely thankful with the office of Sponsored Student who always ensure I never had to worry about my scholarship status.

Finally, I want to thank the Fulbright scholarship program and the Secretaría Nacional de Ciencia, Tecnología e Innovación (SENACYT) in Panama for giving me the opportunity to pursue my dream of study abroad.

DEDICATION

Dedicado a mis maravillosos padres Roberto y Lastenia y a mi amada esposa Doris quienes siempre estuvieron conmigo.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation for this study

Power systems are complex and constituted by several elements. For instance, in [4] a representation of the Danish power system formed by 470 buses is used to study the integration of wind power, in [5] the Indian southern power system is represented by a system with 246 buses, and in [6] power system stabilizers are designed for the 1262 buses Northwest China Grid. Thus, dynamic representations of power systems usually result in the order of hundreds or even thousands; for example, the order of the open-loop representation for the later system is 1643.

The increasingly integration of non-linear power electronic-based sources increments the order of the system and introduces more complexity to it. For the case of distribution level power systems, the inclusion of distributed generation, often based on voltage source converters, transforms the system into a bi-directional power flow network. This creates phenomena not seen before in the network, and the study of electromagnetic transients acquires more attention. In addition, the inclusion of power electric-based equipment increases the computation time, for each switching implies a new network configuration. For instance, [7] reports that the simulation of a given distribution feeder without penetration of power electronic-based sources took 80 seconds, while the simulation of the same feeder with high penetration of this sources took approximately two hours.

Moreover, the real-time simulation of power system requires the solution of the network in a determined time step in actual clock time. Thus, with a constant computing power, the larger the system the bigger the time step. For the case of power-electronic-based simulation, a small-time step is required to accurate reproduce the phenomena involved, which limits the amount of nodes that can be simulated [8]. Hence, when the computing power limit is reached, three options may be available: increase the computing power which may be costly, increase the time step which may be unacceptable for the accuracy of the solution, or simplify the representation of the system at expense of inaccuracy issues also.

In addition, the application of traditional control theory concepts to power system is complicated due to the size of the system. The representation of power systems often generates large and sometimes ill-conditioned matrices, and the system is often treated as a multiple-input multiple-output system. Many control design methods generate gains of the same order of the system that is controlled. This makes complex and computationally demanding the implementation of control theory to power systems.

The aforementioned issues motivate in this research to study the applications of model order reduction techniques in power systems. A number of model order reduction methods are available in the literature. However, the main idea of these methods remains the same, to reduce the order of the system while keeping, as far as possible, its original properties. The specific purpose of this study is expanded in the next section.

1.2 Purpose of this study

The purpose of this study is to apply model order reduction methods to a given power system. Specifically, two methods are emphasized: Krylov subspace-based and Balance truncation. A Krylov subspace-based method is applied to improve the simulation in the context of distribution-level power system; specifically, the IEEE-123 Node Test Feeder presented in [2]. On the other hand, the Balanced Truncation method is implemented for designing a power system controller. This time, the reduction method is applied in the context of generation and transmission network; specifically, the 3-Machine 9-Bus power system presented in [9] . The objective of the power system controller is to damp the oscillations present in the uncontrolled system.

The simulation for the accomplishment of the above objectives are carried out using MATLAB™ code editor and MATLAB/SIMULINK™ SimPower System™ package.

1.3 Outline of this thesis

This thesis is outlined as follows:

Chapter 2: This chapter condensed the theory background regarding the two model order reduction methods implemented in this study: Krylov subspace and Balanced Truncation.

Chapter 3: The implementation of the Krylov subspace method for the reduction of the IEEE-123 Node Test Feeder is presented. The test feeder is described in detailed, and the assumptions adopted for the simulation of this feeder are listed. A SimPower System™ model is developed as well as a state-space representation of the system. The comparison of the reduced model and the full-order model is reported. In addition, a two-sided Arnoldi algorithm is implemented to reduce the deviation between the full-order and reduced model.

Chapter 4: The implementation of the Balanced Truncation for the controller design of the 3-Machine 9-Bus system is presented in this chapter. A linearized model of the system is obtained, and its reduced model is used to simplify the design and implementation of the LQR controller.

Chapter 5: Conclusions and future work suggestions regarding this work are presented in this chapter.

CHAPTER 2

MODEL ORDER REDUCTION IN POWER SYSTEM

This chapter discusses the theory background concerning model order reduction methods. Special emphasis is made on Balance Truncation and Krylov subspace-based method since these methods were adopted for the development of this work. The aim of model order reduction methods is to reduce the size of the system while maintaining its properties. The reduction methods are developed considering the preservation of certain properties as: stability [10] or passivity [7]. Besides the preservation of these properties, keeping the input-output behavior is crucial; therefore, the amount of reduction that can be applied to the original system is going to be determined by the acceptable deviation between the output of the original and reduced system.

2.1 Model order reduction applications

Model order reduction methods have been applied in several disciplines. One of these disciplines is the design and simulation of integrated circuits, where the order of the systems is of several thousands. Another area of application for model order reduction methods is the analysis of electromagnetic phenomena in telecommunications [11]. Whenever the size of the system represents an inconvenient, there may be an opportunity for applying model order reduction methods. Power system model reduction is a good implementation as well. The accurate representation of actual power systems can easily surpass the order of hundreds or even thousands since each bus of the system contributes with information of the behavior of the system. For instance, in [4] a representation of the Danish power system formed by 470 buses is used to study the integration of wind power, in [5] the Indian southern power system is represented by a system with 246 buses, [12] implements a 111 nodes representation of a distribution feeder from Moscow; moreover, the IEEE Distribution Test Feeder Working Group presents in [2] a 8500 nodes system based on a typical North American distribution feeder.

The more detailed and intricated the representation is, the higher the order of the model. The model of the IEEE-123 Node Test Feeder developed in the following chapter has an order of 110. However, a representation of the same test feeder reported in [7] has an order of 916. The difference in the size of the system underlined in the model of the distribution lines and the methods for representing the system.

2.2 Model order reduction classification

Many model order reduction methods are reported in the literature. They can be accommodated in categories based on the mathematical principles in which they are found. A fair classification of model order reduction methods is presented in [13]. In this classification, the reduction methods are grouped based on the type of domain or system representation in which they have been developed (frequency and time domain), and based on the theoretical foundation in which they were developed (polynomial approximation, state truncation and parameter approximation). According to [13], polynomial approximation-based methods are suitable for system representation in the frequency domain, while parameter approximations can be applied either for problems formulated in the time or frequency domain. In the case of polynomial approximation, the coefficient of the transfer function of the original system is approximated by a reduced order transfer function; for instance, moment matching methods can be allocated in this category. State truncation-based methods use control theory concepts to accommodate the state representation, and systematically eliminate the less relevant states for the dynamic of the system. In this category, balanced truncation methods can be mentioned. Finally, parameter approximation methods apply optimization methods to minimize the error between parameters in the original and reduced system.

The separation of model order reduction methods may not be relevant anymore because, in recent years, model reduction techniques combining concepts of the methods presented before have been reported. For instance, in [14], a polynomial approximation-based method is implemented to speed up the process of creating a balance realization of the system in a state truncation-based method. In addition, [15] presents a method that uses parameter approximation concepts to find the optimal points around which a polynomial approximation of the original system is found.

2.3 The concept of projection and its application on model order reduction

Despite the classification of model order reduction methods presented in the previous section, many of these methods lay on the concept of projection onto a subspace. Thus, it is convenient to present these concepts next.

The projection theory establishes that a projector $\boldsymbol{P}$ can be defined by using two subspaces $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{L}}$ such that [16]

$$\boldsymbol{\mathcal{M}} \cap \boldsymbol{\mathcal{L}}^{\perp} = \boldsymbol{0} \tag{2-1}$$

where $\boldsymbol{\mathcal{M}}$ is the subspace spanned by the range of $\boldsymbol{P}$ and $\boldsymbol{\mathcal{L}}$ is the subspace orthogonal to the null subspace of $\boldsymbol{P}$. The projector $\boldsymbol{P}$ can define the projection $\boldsymbol{u}$ of the any vector $\boldsymbol{x}$ over the subspace $\boldsymbol{\mathcal{M}}$ and the residual vector $\boldsymbol{r}$ over the subspace $\boldsymbol{\mathcal{L}}^{\perp}$ as follows

$$\boldsymbol{u} = \boldsymbol{P}\boldsymbol{x}, \tag{2-2}$$

$$\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}. \tag{2-3}$$

Given the linear system

$$\begin{aligned} \boldsymbol{E}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u, \\ y &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{2-4}$$

and the matrices $\boldsymbol{V}$ being basis for the subspace $\boldsymbol{\mathcal{M}}$ and $\mathbf{W}$ basis for the subspace $\boldsymbol{\mathcal{L}}$, the projection techniques find an approximate solution $\boldsymbol{z}$ that is element of the subspace $\boldsymbol{\mathcal{M}}$ and a residual vector $\boldsymbol{r}$ orthogonal to the subspace $\boldsymbol{\mathcal{L}}$ (Petrov-Galerki condition) as follows [17]

$$\boldsymbol{x} = \boldsymbol{V}\boldsymbol{z}. \tag{2-5}$$

The residual of the system is then defined as

$$\boldsymbol{r} = \boldsymbol{b} + (\boldsymbol{A} - \boldsymbol{s}\boldsymbol{E})\boldsymbol{x}, \tag{2-6}$$

$$\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{r} = \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{b} + \boldsymbol{W}^{\boldsymbol{T}}(\boldsymbol{A} - \boldsymbol{s}\boldsymbol{E})\boldsymbol{V}\boldsymbol{z} = \boldsymbol{0}. \tag{2-7}$$

The matrices of the system projected on $\boldsymbol{\mathcal{M}}$ and orthogonal to $\boldsymbol{\mathcal{L}}$ are

$$\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{E}\boldsymbol{V}, \qquad \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{A}\boldsymbol{V}, \qquad \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{b}, \qquad \boldsymbol{C}\boldsymbol{V}. \tag{2-8}$$

The problem of model order reduction methods based on projection is then summarized as finding the reduced basis matrices $\boldsymbol{W^T}$ and $\boldsymbol{V}$ that project the original system onto a smaller subspace. Two of these methods, Krylov subspace and Balanced Truncation, are presented next.

2.4 Krylov subspace-based model order reduction

The Krylov subspace-based model order reduction methods are moment matching methods. The priciple in these methods is to replicate the moments of the original system on the moments of the reduced system. The moments of a system are defined as the coefficients of the Taylor series expansion of its transfer function [10], [18]–[20] as follows, borrowing the notation from [18]:

Given the transfer function of a system, where $\boldsymbol{B}$ is the input matrix for the cases of multiple inputs, be defined as

$$\boldsymbol{G}(s) = \boldsymbol{C}(s\boldsymbol{E} - \boldsymbol{A})^{-\mathbf{1}}\boldsymbol{B} \tag{2-9}$$

the moments of the system around the interpolation point s=0 are

$$\boldsymbol{M}_i = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{E})^i\boldsymbol{A}^{-\mathbf{1}}\boldsymbol{B} \quad i = 0{,}1 \ldots \tag{2-10}$$

The moments or the coefficients of the Taylor series can be calculated around different values of s, or in other words, can be calculated around different interpolation points. Using the moments of the system calculated around s=0 will tend to emulate the original system at low frequencies [18], and using the moments calculated around s=∞ will tend to emulate the original system at high frequencies [21].

The moments of the system around s=0 can be approximated from the Krylov subspace by using the appropriate initial vectors. This is desired since using the Krylov subspace, instead of using the explicit Taylor series elements, results in a more mathematically stable procedure [19]. Before continuing is appropriate to state the definition of Krylov subspace as follows

$$K(\boldsymbol{T},\boldsymbol{R}) = span(\boldsymbol{T},\boldsymbol{TR},\boldsymbol{T^2R} \dots .\boldsymbol{T^{r-1}R}). \tag{2-11}$$

By choosing $\boldsymbol{T} = \boldsymbol{A^{-1}E}$ and $\boldsymbol{R} = \boldsymbol{A^{-1}B}$, where *r* is the desired dimension of the reduced model, the elements of the Krylov subspace will reproduce the information of the first *r* elements of the Taylor series [19].

An orthonormal basis for $K(\boldsymbol{T},\boldsymbol{R})$ is required for an effective model order reduction because the columns in the Krylov subspace tent to point the same direction after a considerable amount of recurrences. These bases are defined in equation (2-8) as the orthonormal matrices $\boldsymbol{W}$ and $\boldsymbol{V}$. In the case of the system (2-4), the projection of the original system can be one-sided or two-sided projections. In one-sided projections, the matrix $\boldsymbol{V}$ is calculated so that constitutes the basis of the subspace $K(\boldsymbol{A^{-1}E},\boldsymbol{A^{-1}B})$ while $\boldsymbol{W}$ is just selected to allow the reduced matrix to be reversible; meanwhile, in two-sided projections, the matrix $\boldsymbol{W}$ is additionally calculated so that constitutes the basis of the subspace $K(\boldsymbol{A^{-T}E^T},\boldsymbol{A^{-T}C^T})$ [18], [19]. The one-sides projections can reproduce *r* moments of the original systems; in contrast, the two-sided projections reproduces 2*r* moments.

The resulting projection would be represented as follows

$$\boldsymbol{E_r} = \boldsymbol{W^TEV}, \boldsymbol{A_r} = \boldsymbol{W^TAV}\,, \boldsymbol{B_r} = \boldsymbol{W^TB}\,, \boldsymbol{C_r} = \boldsymbol{CV}. \tag{2-12}$$

There are two commonly used process to compute the orthonormal basis for the Krylov subspace: Arnoldi-based and Lanczos-based. The Lanczos-based algorithms are suitable for two-sided projections since the algorithms compute the orthogonal matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ simultaneously [19]. [22] suggests that Lanczos algorithms are especially useful for the case of non-symmetric system matrix. [18], on the other hand, proposes to use twice an Arnoldi-based algorithm to compute the basis corresponding to the input and output Krylov subspace.

The Arnoldi-based algorithm is described next since the algorithm developed for model reduction implemented in this work is based on this type of algorithms. A comprehensive explanation of Lanczos-based algorithms can be found in [19], [22].

2.4.1 Arnoldi algorithm

As mention before, after selecting the suitable vectors that form the Krylov subspace, an orthonormal basis of this subspace should be computed. The need for this orthonormal basis underlies in the fact that the vectors in the Krylov subspace conserve a certain level of dependency [19]. Arnoldi algorithms compute orthonormal bases by decomposing the system matrix, in the case of system (2-4) $\boldsymbol{T} = \boldsymbol{A}^{-\boldsymbol{1}}\boldsymbol{E}$, into an orthonormal matrix $\boldsymbol{Q}$ and an upper Hessenberg $\boldsymbol{H}$ matrix as follows [23]

$$\boldsymbol{T}\boldsymbol{Q}_{\boldsymbol{r}} = \boldsymbol{Q}_{\boldsymbol{r}+\boldsymbol{1}}\boldsymbol{H} \tag{2-13}$$

where *r* is the number of column of the matrix $\boldsymbol{Q}_{\boldsymbol{r}}$ and the order of the reduced system.

The matrix $\boldsymbol{Q}_{\boldsymbol{r}+\boldsymbol{1}}$ requires an extra calculation; however, this matrix is not required for the sake of model reduction since the matrix $\boldsymbol{Q}_{\boldsymbol{r}}$ works as the orthonormal basis of *r*th order reduced systems.

The decomposition process in the classic Arnoldi algorithm is executed using Gram-Schmidt process [22], [23] which is often referred as QR decomposition. Borrowing notation from [23], the steps of the Arnoldi algorithm can be stated as follows:

Considering the columns of the basis matrix $\boldsymbol{R} = \boldsymbol{A}^{-\boldsymbol{1}}\boldsymbol{B} = (\boldsymbol{r_1 r_2 r_3} \dots \boldsymbol{r_r})$

$$\boldsymbol{q_1} = \frac{\boldsymbol{r_1}}{\|\boldsymbol{r_1}\|_2}, \tag{2-14}$$

$$\overline{\boldsymbol{q}_{\boldsymbol{\imath}+\boldsymbol{1}}} = \boldsymbol{Tq_i} - \sum_{\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{i}} \boldsymbol{q_j} h_{ji} \tag{2-15}$$

where h_{ji} is the inner product of the vectors $\boldsymbol{q_j}$ and $\boldsymbol{Tq_i}$ and elements of the matrix $\boldsymbol{H}$

$$h_{ji} = \boldsymbol{q_j T q_i}\ , \tag{2-16}$$

normalizing the rest of the elements of $\boldsymbol{Q_r}$

$$\boldsymbol{q}_{\boldsymbol{i}+\boldsymbol{1}} = \frac{\overline{\boldsymbol{q}_{\boldsymbol{\imath}+\boldsymbol{1}}}}{\|\overline{\boldsymbol{q}_{\boldsymbol{\imath}+\boldsymbol{1}}}\|_2}. \tag{2-17}$$

The steps presented up to equation (2-17) provide the required basis for the projection of the original system and the consequent model reduction. The rest of the steps for the completion of the Arnoldi decomposition of the matrix $\boldsymbol{T}$ are stated in [23].

The Arnoldi algorithm can be implemented in MATLAB™ code editor taking advantage of some predefined functions in this software.

2.5 Balanced truncation based model order reduction

The main idea of the model order reduction methods based on balanced truncation is to re-arrange the system in its balanced realization form. This means that the states of the system are arranged such that, the more controllable and observable modes of the system are at the beginning of the system's realization. Consequently, the original system can be truncated where its modes become irrelevant for representing its behavior.

Borrowing the notation from [21], given a system of the form (2-18)

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu}\,, \boldsymbol{y} = \boldsymbol{Cx}. \tag{2-18}$$

A system is in its balanced realization form if its controllability $\boldsymbol{W_c}$ and observability $\boldsymbol{W_o}$ Gramians are equal; however, most likely, this is not the case of the original system (2-18). Thus, the balanced realization theory establishes that there exists a similarity transformation $\boldsymbol{x}$=$\boldsymbol{Tz}$ such that the condition above is met. Moreover, the eigenvalues of the product $\boldsymbol{W_c}$ $\boldsymbol{W_o}$ does not change over any coordinate change [13]. This means that these values, called Hankel values, have the same meaning for the original and transformed system.

The problem of balanced truncation model reduction is narrow to the calculation of the matrix $\boldsymbol{T}$, and this calculation can be carry out as follows [17].

The controllability and observability Gramians can be computed from the Lyapunov equations (2-19) and (2-20) respectively.

$$\boldsymbol{AW_c} + \boldsymbol{W_cA^T} + \boldsymbol{BB^T} = \boldsymbol{0}, \tag{2-19}$$

$$\boldsymbol{A^TW_o} + \boldsymbol{W_oA} + \boldsymbol{C^TC} = \boldsymbol{0}. \tag{2-20}$$

The Cholesky factor of $\boldsymbol{W_c}$ and $\boldsymbol{W_o}$ are computed

$$\boldsymbol{W_c} = \boldsymbol{L_cL_c^T}, \tag{2-21}$$

$$\boldsymbol{W_o} = \boldsymbol{L_o^TL_o}. \tag{2-22}$$

The singular value decomposition of the product $\boldsymbol{L_o^T L_c}$ is calculated.

$$\boldsymbol{L_o^T L_c = U\Sigma V^T} \tag{2-23}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix containing the invariant Hankel values of the system.

Finally, the matrix $\boldsymbol{T}$ can be computed as

$$\boldsymbol{T = L_c^T U\Sigma^{-1/2}}. \tag{2-24}$$

The balanced realization of the system (2-18) can be obtained by projection:

$$\boldsymbol{\widehat{A} = T^{-1}AT}\,,\boldsymbol{\widehat{B} = T^{-1}B}\,,\boldsymbol{\widehat{C} = CT}. \tag{2-25}$$

The matrices of the balanced system above can be decomposed as follows

$$\boldsymbol{\widehat{A}} = \begin{bmatrix} \boldsymbol{\widehat{A}_{11}} & \boldsymbol{\widehat{A}_{12}} \\ \boldsymbol{\widehat{A}_{21}} & \boldsymbol{\widehat{A}_{22}} \end{bmatrix}, \boldsymbol{\widehat{B}} = \begin{bmatrix} \boldsymbol{B_1} \\ \boldsymbol{B_2} \end{bmatrix}, \boldsymbol{\widehat{C}} = [\boldsymbol{C_1} \quad \boldsymbol{C_2}] \tag{2-26}$$

where the first blocks of the matrices are related with the first *r*th more controllable and observable states. The reduced system can be established by extracting the matrices from (2-26) or by projecting the original system over the first *r*th columns of the matrix $\boldsymbol{T}$.

$$\boldsymbol{A_r = T_r^T A T_r}\,,\boldsymbol{B_r = T_r B}\,,\boldsymbol{C_r = C T_r^T}. \tag{2-27}$$

The balance truncation method preserves the original modes of the system that have not been truncated; therefore, the reduced model presents a good representation of the original system in a wide variety of frequencies [20], [21]. However, the calculation of $\boldsymbol{\Sigma}$ and the two Lyapunov is computationally demanding since requires computing the singular value decomposition of the system, and is even more difficult for high order systems [21] as distribution level power systems.

2.6 Conclusion

The applications of model order reduction methods are spread over many disciplines. Power systems is one of these disciplines since the dynamic representation of power system often requires systems of high order.

Model order reduction methods can be classified considering the domain they are applied or their mathematical foundations. However, several of these methods work by projecting the original system onto a smaller subspace. Two of these projection-based methods are Krylov subspace-based and Balanced truncation.

Krylov subspace seeks to replicate the *r*th moments of the original system on the moments of the reduced system. Balanced truncation, on the other hand, rearranges the system in its balanced realization form in order to truncate the modes of the system that are less controllable and less observable.

CHAPTER 3

SIMULATION OF DISTRIBUTION-LEVEL NETWORK USING KRYLOV SUBSPACE MODEL ORDER REDUCTION

This chapter presents the application of a model order reduction method based on Krylov subspace on a distribution level network. The reduction is applied to the IEEE-123 Node Test Feeder. In order to perform the model reduction, a Modified State Space representation is adopted from [3], [7], and the results are compared with an equivalent MATLAB/SIMULINK™ model. Next, the model order reduction is carried out by implementing an Arnoldi algorithm, as presented in the previous chapter. The model order reduction produced using the approach presented in [7] is further improved by using a two-sided Arnoldi algorithm. The reduced model is used to simulate the transient of a PV sources connected to the system.

3.1 Test Feeder selection criteria

The IEEE-123 Node Test Feeder has been selected on this study as a distribution network benchmark for demonstrating the application of model order reduction methods. This test feeder, together with other test feeders, was presented by the IEEE Distribution Test Feeder Working Group as a benchmark for software developers who needed to test power flow algorithms. The IEEE-123 Node Test Feeder was selected considering that includes characteristics of typical distribution level feeders:

- Three- and single-phase branches.

- Three- and single-phase loads.
- Considerable unbalance in power flow.
- Undervoltage issues.

The size of the test feeder was also considered. The test feeder has an appropriate number of nodes, which facilitates both the implementation and the evaluation of model order reduction methods. The topology of the test feeder was considered as well. The topology of the IEEE 123 nodes test feeder includes large three-phase trunk branches suitable for simulating the addition of many medium-voltage generation, and clusters of short single-phase branches adequate for simulating residential-level distributed generation.

The information required to create the test feeder is presented in [24]. Fig. 3-1 shows the layout of the test feeder as presented in [2]. The information about the spot loads, line segments' configuration, and capacitor banks is organized in tables like those shown in Fig. 3-2.

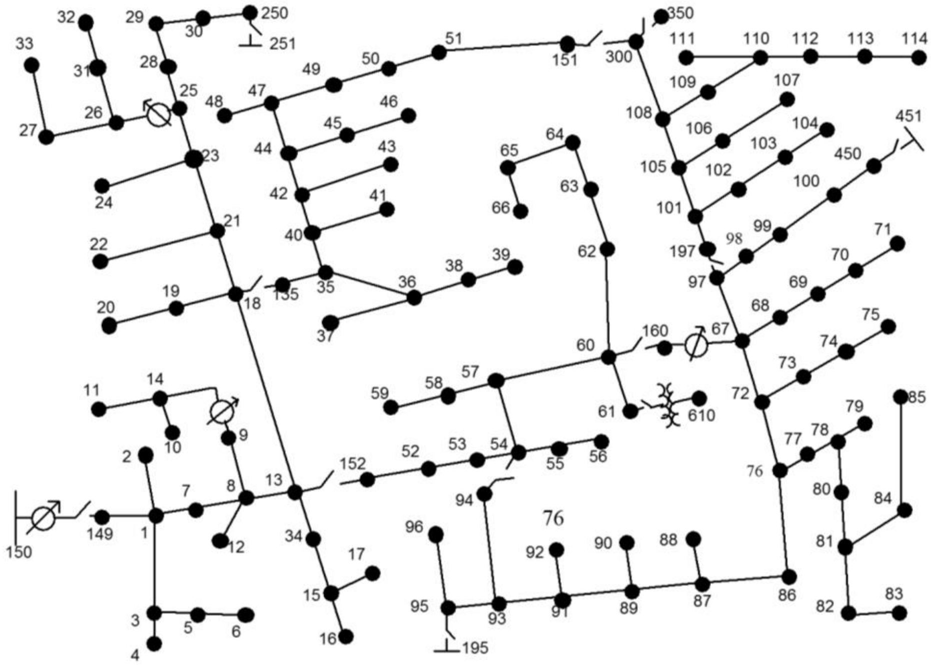
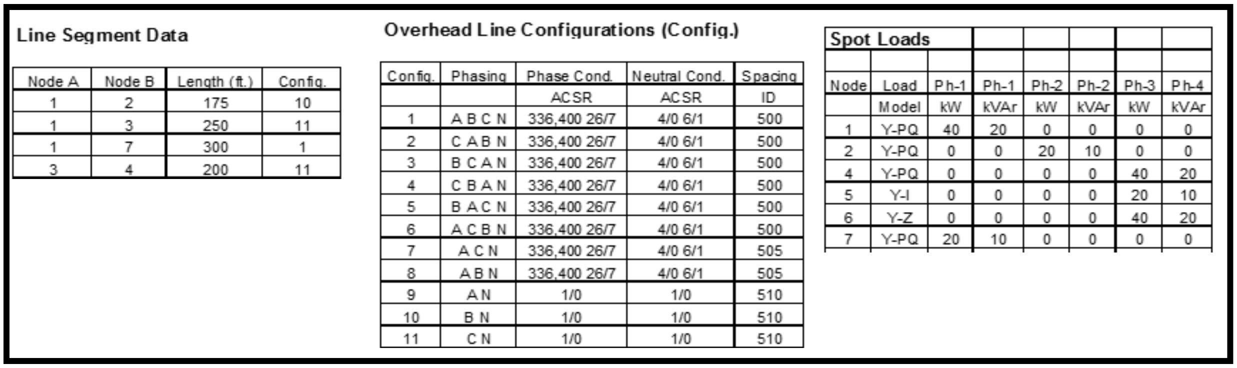


Fig. 3-1. IEEE-123 Node Test Feeder [2].



Line Segment Data

Node A	Node B	Length (ft.)	Config.
1	2	175	10
1	3	250	11
1	7	300	1
3	4	200	11

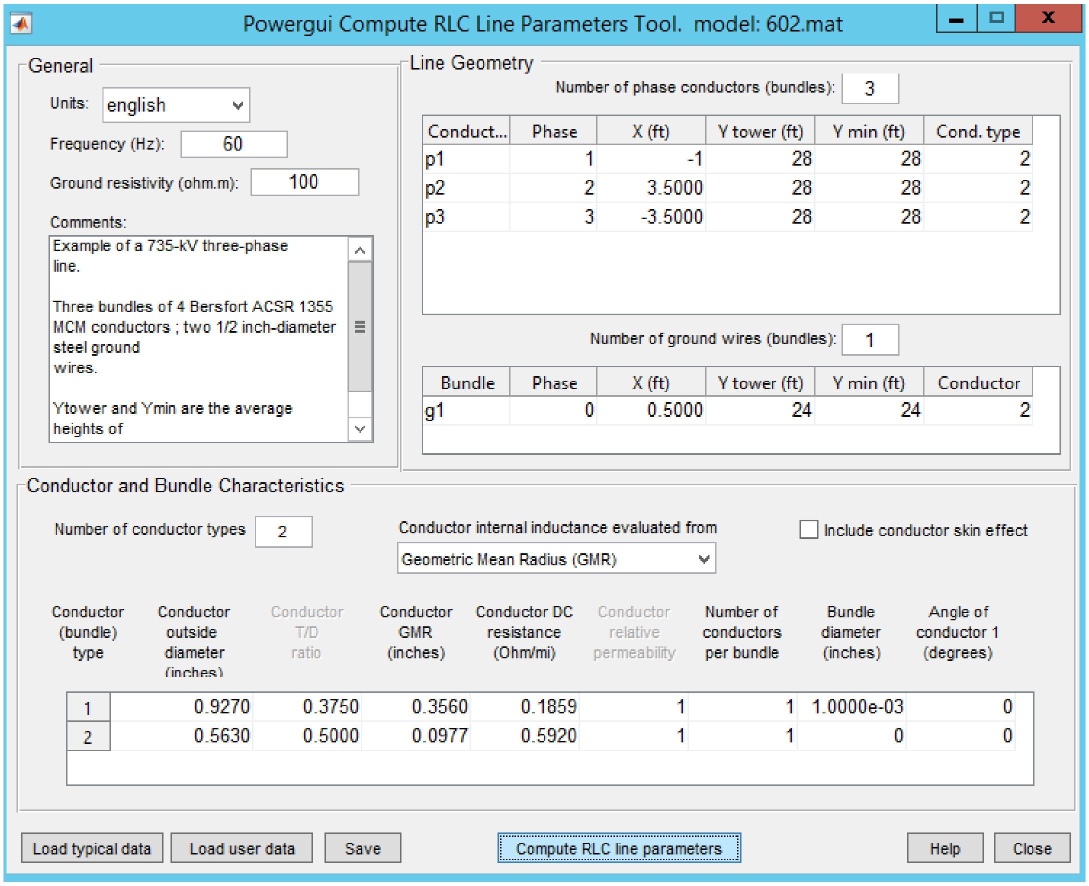
Overhead Line Configurations (Config.)

Config.	Phasing	Phase Cond.	Neutral Cond.	Spacing
		ACSR	ACSR	ID
1	A B C N	336,400 26/7	4/0 6/1	500
2	C A B N	336,400 26/7	4/0 6/1	500
3	B C A N	336,400 26/7	4/0 6/1	500
4	C B A N	336,400 26/7	4/0 6/1	500
5	B A C N	336,400 26/7	4/0 6/1	500
6	A C B N	336,400 26/7	4/0 6/1	500
7	A C N	336,400 26/7	4/0 6/1	505
8	A B N	336,400 26/7	4/0 6/1	505
9	A N	1/0	1/0	510
10	B N	1/0	1/0	510
11	C N	1/0	1/0	510

Spot Loads

Node	Load	Ph-1	Ph-1	Ph-2	Ph-2	Ph-3	Ph-4
	Model	kW	kVAr	kW	kVAr	kW	kVAr
1	Y-PQ	40	20	0	0	0	0
2	Y-PQ	0	0	20	10	0	0
4	Y-PQ	0	0	0	0	40	20
5	Y-I	0	0	0	0	20	10
6	Y-Z	0	0	0	0	40	20
7	Y-PQ	20	10	0	0	0	0

Fig. 3-2. Sample of table presented in [2].



Powergui Compute RLC Line Parameters Tool. model: 602.mat

General

Units: english

Frequency (Hz): 60

Ground resistivity (ohm.m): 100

Comments:

Example of a 735-kV three-phase line.

Three bundles of 4 Bersfort ACSR 1355 MCM conductors ; two 1/2 inch-diameter steel ground wires.

Ytower and Ymin are the average heights of

Line Geometry

Number of phase conductors (bundles): 3

Conduct...	Phase	X (ft)	Y tower (ft)	Y min (ft)	Cond. type
p1	1	-1	28	28	2
p2	2	3.5000	28	28	2
p3	3	-3.5000	28	28	2

Number of ground wires (bundles): 1

Bundle	Phase	X (ft)	Y tower (ft)	Y min (ft)	Conductor
g1	0	0.5000	24	24	2

Conductor and Bundle Characteristics

Number of conductor types 2

Conductor internal inductance evaluated from: Geometric Mean Radius (GMR)

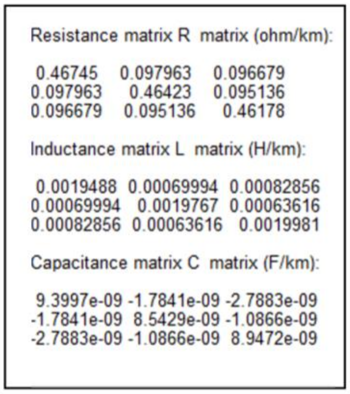
□ Include conductor skin effect

Conductor (bundle) type	Conductor outside diameter (inches)	Conductor T/D ratio	Conductor GMR (inches)	Conductor DC resistance (Ohm/mi)	Conductor relative permeability	Number of conductors per bundle	Bundle diameter (inches)	Angle of conductor 1 (degrees)
1	0.9270	0.3750	0.3560	0.1859	1	1	1.0000e-03	0
2	0.5630	0.5000	0.0977	0.5920	1	1	0	0

Fig. 3-3. RLC Line Parameter Tool graphical interface.

In [24] a guidance for interpreting the information presented in [2] is available. For example, the spacing for each line segment configuration is presented in [24], as well as information regarding the type of spot loads.

The impedance of each line segment configuration was calculated using the RLC Line Parameter Tool. The RLC Line Parameter Tool is part of the software package SimPower System™ of MATLAB/SIMULINK™, and can be found inside the block “powergui” or be called using the MATLAB™ command “power_lineparam”. The graphical interface of the RLC Line Parameter Tool is shown in Fig. 3-3. The entries for this tool are separated in three main sections: General, Line geometry, and Conductor and Bundle characteristics. In the General section, a frequency of 60Hz and a ground resistivity of 100ohms was used, as can be observed in Fig. 3-3. For the Line geometry, the distance between conductors was retrieved from [24], per each line segment configuration. Finally, the Conductors and Bundles characteristics were adopted from [25], and the conductor internal inductance is calculated from the geometric mean radius, since this is the information available in [25]. No conductor skin effect was considered.



```
Resistance matrix R  matrix (ohm/km):

 0.46745    0.097963    0.096679
0.097963     0.46423    0.095136
0.096679    0.095136     0.46178

Inductance matrix L  matrix (H/km):

 0.0019488  0.00069994  0.00082856
0.00069994   0.0019767  0.00063616
0.00082856  0.00063616   0.0019981

Capacitance matrix C  matrix (F/km):

 9.3997e-09 -1.7841e-09 -2.7883e-09
-1.7841e-09  8.5429e-09 -1.0866e-09
-2.7883e-09 -1.0866e-09  8.9472e-09
```

Fig. 3-4. Matrices resulting from RLC Line Parameter Tool.

The outcome of this tool is a series of square matrices containing the value of resistance, capacitance and inductance for each line segment configuration. Fig. 3-4 shows an example of resistance, inductance and capacitance matrices calculated using the RLC Line Parameter Tool corresponding to the line segment configuration #2. The matrices of resistance, inductance and capacitance for all line segment configurations, in per distance unit of feet, are compiled in Appendix A.1.

3.1.1 Assumptions adopted for simplification

Some assumptions were adopted for the sake of simplification when comparing the MATLAB/SIMULINK™ model discussed in this section and the state-space model developed in later sections. These assumptions seek to avoid interactions that may be neglected for the case of distribution-level feeders or interactions that may be complex to represent in a mathematical model. For instance, the line segments' mutual inductances are neglected since this phenomenon is insignificant in short lines like distribution lines. To disregard the mutual inductance, the elements off the main diagonal of the resistance, inductance and capacitance matrices are set to zero, as can be observed in Appendix A.1. In terms of modeling complexity, for instance, all loads are considered constant impedance, although the original test feeder combines constant power and constant current loads as well. The reason for this assumption will be clearer when discussing the process of creating a mathematical model of the system in the following sections.

A summary of assumptions and simplification are listed below:

- The model is simulated from the 4.16kV side; therefore, the power transformer connected between node 150 and 149 is neglected.
- No voltage regulator is included.

- Line segments are assumed with no mutual inductance.
- Capacitor banks are considered as constant impedances.
- All loads were simulated using series three- or single-phase load blocks.
- All loads were modeled as constant impedances.

The resulting circuit in MATLAB/SIMULINK™ is shown in Fig. 3-5. Three-phase loads are colored blue, single phase loads are colored orange and yellow, capacitor banks are colored sky blue, and the voltage source is colored magenta. A load flow report was generated using the Load Flow Tool provided in the “powergui” block. The summary of the report is shown in Table 3-1. The numbers in this report are not expected to replicate the original load flow reported in [2] because several assumptions were adopted as mentioned before. However, the most loaded phase is phase A followed by Phase C and then Phase B, which agrees with the order presented in [2]. This behavior is reasonable since no change in the topology has been made.

The voltage profiles for each phase was also analyzed, and they are shown in Fig. 3-6 to 3-8. The voltage is higher for those nodes nearer the source; as it moves away from the source, the voltage decreases. Slight increments can be observed where capacitor banks are connected; for instance, in node 83 a three-phase capacitor bank is connected and this is reflected in the three phases. In addition, the overall voltage profile is higher for phase-B. This behavior is expected since phase-B is the least loaded as was indicated before.

Table 3-1. Summary of load flow report.

	Phase A		Phase B		Phase C		Total	
V (pu deg.)	1.0436	-0.02	1.0437	-120.02	1.0436	119.98		
Gen (kW kvar)	1399.01	688.94	912.63	454.00	1133.30	516.02	3444.94	1658.96

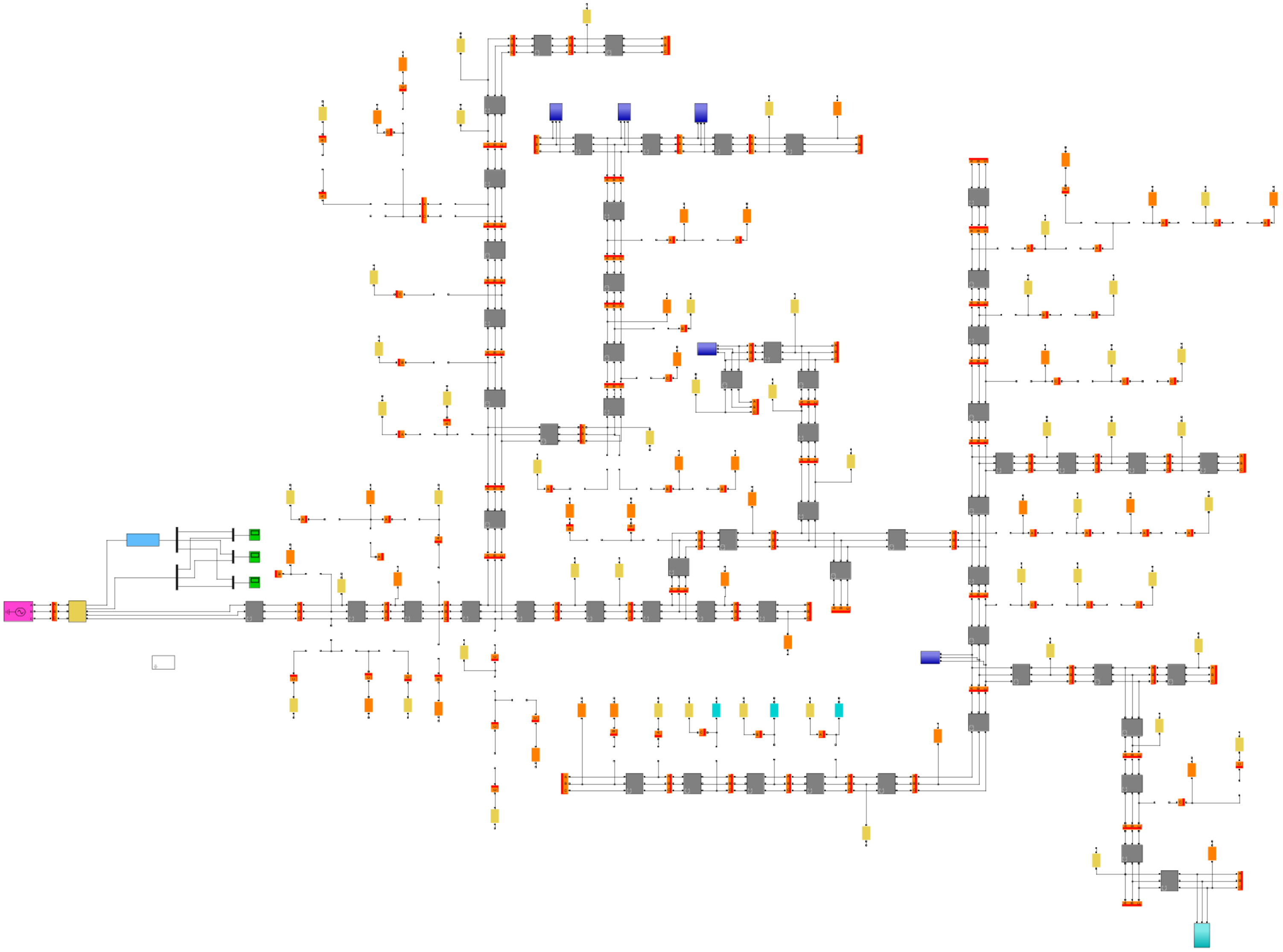


Fig. 3-5. MATLAB/SIMULINK™ model of IEEE-123 Node Test Feeder.

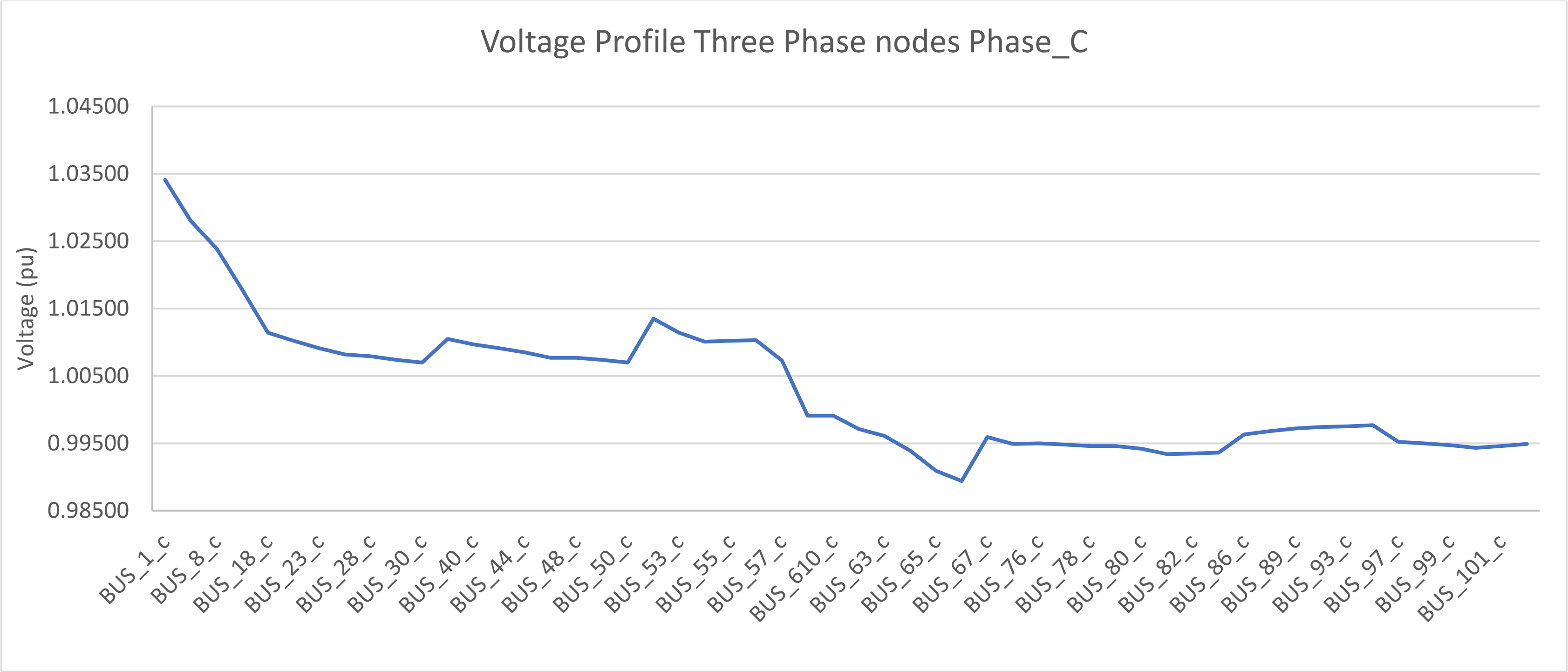


Fig. 3-6. Voltage profile three-phase nodes phase C.

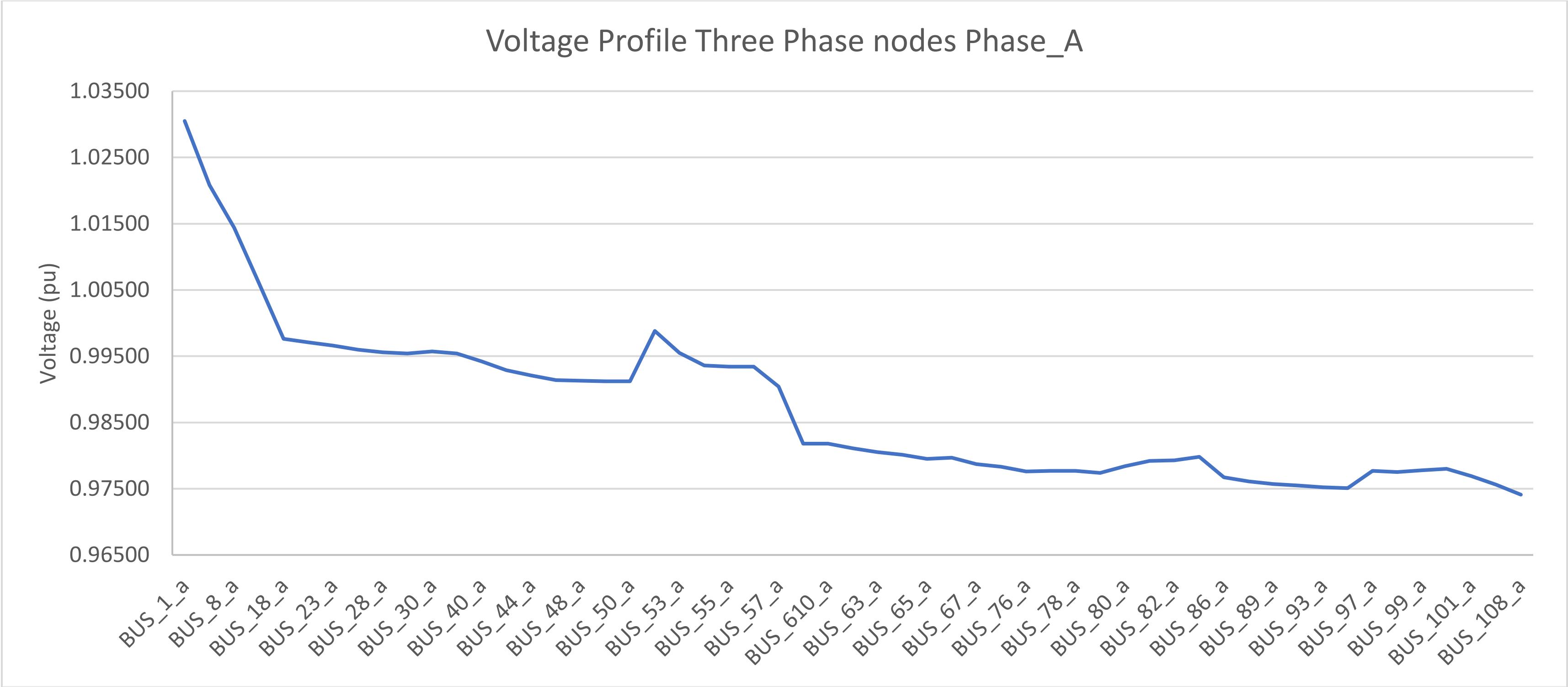


Fig. 3-7. Voltage profile three-phase nodes of phase A.

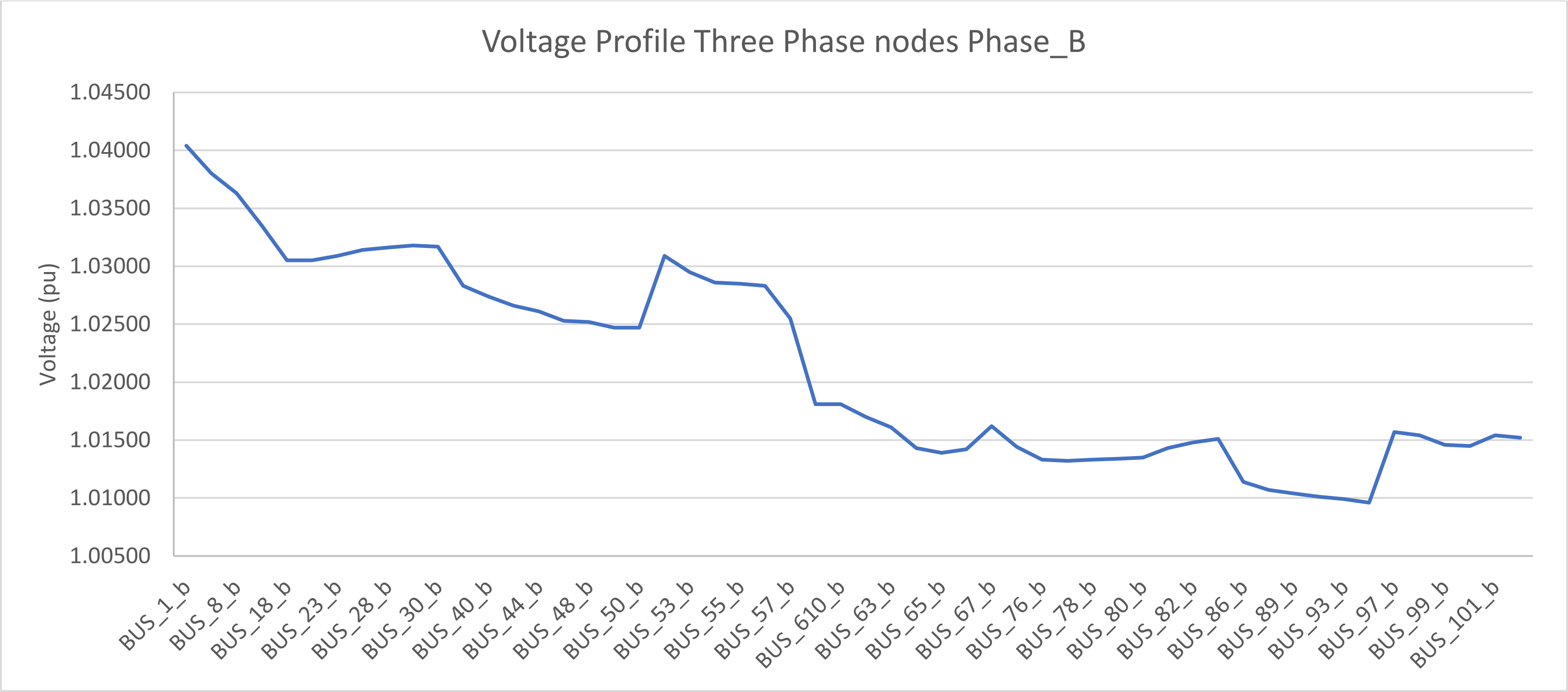


Fig. 3-8. Voltage profile three-phase nodes of phase B.

Although the model of the test feeder simulated cannot be directly compared with the results presented in [2] due to the assumptions made, the results of the load flow and voltage profile suggest the model is being simulated appropriately and is behaving as expected for a radial distribution-level feeder.

3.2 Modified State Space model formulation

In this section, the method adopted for formulating the dynamic representation of the IEEE-123 Node Test Feeder is presented.

In [7], a Modified State Space representation is developed. This representation uses the so-called node incidence matrix to gather information related to the topology of the system. In addition, a general branch model is presented to simulate the voltage-current expression of each branch. The representation proposed in [3] is better adapted for power system implementation in [7]. In [7], this adapted representation is also applied to the IEEE-123 Node Test Feeder. We are going to adopt this representation for the sake of this study and a brief description of its mathematical foundation is presented next.

As mentioned before, Modified State Space representation involves the establishment of the node incidence matrix. In order to build this matrix, a row in the matrix is designated per each node in the system; this can apply for any number of phases. Thus, if the node is a three-phase node, three rows should be designated, one per each phase. A column in the matrix is designated per each branch in the system. Like in the case of nodes, if the branch is a three-phase branch, three columns should be designated. As a matter of example the node incidence matrix for the circuit shown in Fig. 3-9 is developed next.

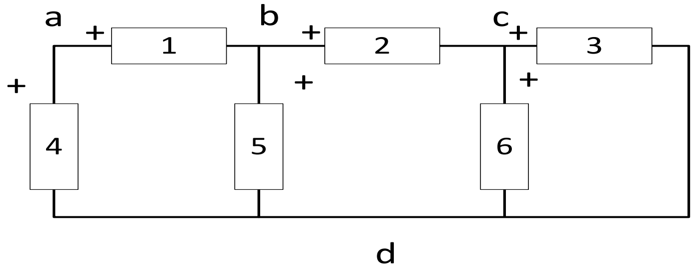


Fig. 3-9. Circuit for node incidence matrix sample.

In the circuit shown in Fig. 3-9, the nodes are assigned letters while the branches are numbered. In addition, the assumed polarity is marked for each branch. The resulting incidence matrix is shown below; the notations are borrowed from [3]

$$\boldsymbol{A_c} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 & -1 & -1 \end{bmatrix} \end{array}. \tag{3-1}$$

A positive one is allocated in the row corresponding to the node where the polarity mark of the branch in connected; a negative one is allocated in the row corresponding to the node where the non-polarity terminal of the branch is connected. Thus, each column can have just one positive one and one negative one. The rows of $\boldsymbol{A_c}$ represent the Kirchhoff's current law for each node.

The matrix $\boldsymbol{A_c}$ can be re-ordered into its reduced row Echelon form [3]

$$\boldsymbol{\widehat{A}_c} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\widehat{A}} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{3-2}$$

where $\boldsymbol{I}$ is the identity matrix whose dimensions are *n*-1 x *n*-1, *n* being the number of nodes. $\boldsymbol{\widehat{A}}$ is the resulting matrix after the re-arrangement of $\boldsymbol{A_c}$ and its dimensions are *n*-1 x *b*-*n* +1, where *b* is the number of branches. The zero row at the bottom of matrix $\boldsymbol{\widehat{A}_c}$ must appear after the manipulation of $\boldsymbol{A_c}$, a non-zero row at the bottom may means that there is an error in the elements of $\boldsymbol{A_c}$; for instance, one branch may be represented as connected in three nodes.

The reduced row Echelon form of $\boldsymbol{A_c}$ can be computed using the MATLAB™ command "rref". The command "rref" perform the re-arrangement using Gauss-Jordan row elimination with partial pivoting [26]. The resulting matrix after applying "rref" is shown in equation (3-3)

$$\widehat{\boldsymbol{A}}_{\boldsymbol{c}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3-3}$$

The re-ordering of $\boldsymbol{A_c}$ is important since it allows for separation of independent currents from the dependent currents, and a first reduction is accomplished up to this point; in contrast, Modified Nodal Analysis does not include this reduction, resulting in a bigger system representation.

Separating the matrix $\widehat{\boldsymbol{A}}_{\boldsymbol{c}}$, and multiplying by a vector containing the current of each branch

$$\widehat{\boldsymbol{A}}_{\boldsymbol{kcl}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \end{bmatrix}. \tag{3-4}$$

From equation (3-4), it is evident that i_1, i_2 and $i_3$can be expressed by a combination of i_4, i_5 or i_6. Extracting the non-identity matrix from $\widehat{\boldsymbol{A}}_{\boldsymbol{c}}$

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \tag{3-5}$$

Filling in with the corresponding identity matrix and changing the sign of $\widehat{\boldsymbol{A}}$

$$\boldsymbol{B_b^T} = \begin{bmatrix} -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3-6}$$

where the superscript " $^{\boldsymbol{T}}$" indicates the transpose of the matrix. If $\boldsymbol{B_b^T}$ is multiplied by the independent currents i_4, i_5 or i_6 , the resulting vector is the vector of all the branch currents as is shown in equation (3-6)

$$\boldsymbol{B_b^T i_x} = \boldsymbol{i_{br}}. \tag{3-7}$$

The non-transposed matrix $\boldsymbol{B_b}$**,** known as basic loop matrix, contains the information of the mesh analysis of the network. In other word, if this matrix is multiplied by a vector containing the voltages of the branches, it would be the equivalent to the Kirchhoff's Voltage Law. This can be better visualized in equation (3-8)

$$\boldsymbol{\widehat{A}_{kvl}} = \begin{bmatrix} -1 & -1 & -1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \boldsymbol{0}. \tag{3-8}$$

From equation (3-6), it is evident that the circuit can be analyzed using three meshes or loops. These loops are shown in Fig. 3-10.

Equation (3-6) can be re-written as

$$\boldsymbol{B_b v_{br}} = \boldsymbol{0}. \tag{3-9}$$

The next step for building the Modified State Space model is to stablish the equation that represent the voltage-current expression of each branch. The general branch presented in [3] is shown in Fig. 3-11. In contrast, [7] simplifies the branch model not including switches and current sources since, in [7], all the circuit breakers are considered to be in a fixed position and no current sources are included. The same assumptions are adopted in this work.

If the simplification presented in [7] is applied to the circuit in Fig. 3-10, the equation that represents the branch voltage-current expression would be as follows [7]

$$v_{br\_i} = r_i \, i_{br\_i} + L_i \frac{di_{br\_i}}{dt} + v_{Cbr\_i} + e_i. \tag{3-10}$$

Substituting (3-7) into (3-10) and multiplying by $\boldsymbol{B_b}$

$$\boldsymbol{B_b v_{br}} = \boldsymbol{B_b R \, B_b^T i_x} + \boldsymbol{B_b L B_b^T} \frac{d\boldsymbol{i_x}}{dt} + \boldsymbol{B_b v_{Cbr}} + \boldsymbol{B_b e} \tag{3-11a}$$

$$\boldsymbol{0} = \boldsymbol{B_b R \, B_b^T i_x} + \boldsymbol{B_b L B_b^T} \frac{di_x}{dt} + \boldsymbol{B_b v_{Cbr}} + \boldsymbol{B_b e}. \tag{3-11b}$$

The following quantities are going to be defined as in [7]. A matrix $\boldsymbol{W}$ is defined to related the voltage and current in the branches with the inputs and outputs of the model's representation. The dimensions of this matrix are *N* x *b*, where *N* is the number of ports and *b* the number of branches. W_{ij} is equal to 1 if the branch jth is connected to the ith port; if not, W_{ij} is equal to 0 [3], [7]. Thus, equations (3-12) and (3-13) can be written:

$$\boldsymbol{e} = \boldsymbol{W}^T \boldsymbol{v_p}, \tag{3-12}$$

$$\boldsymbol{i_p} = -\boldsymbol{W i_{br}}. \tag{3-13}$$

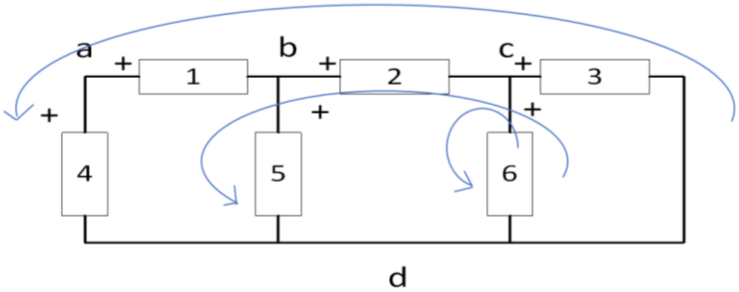


Fig. 3-10. Loops in the circuit sample.

The matrix $\boldsymbol{M}$ is introduced to establish the connection of capacitors in the system. The dimensions of this matrix are *b* x *nc,* where *nc* is the number of capacitors connected to the system. When the jth capacitor is connected to ith branch, a 1 is written in the element M_{ij}; if not, $M_{ij}$is equal to 0 [3], [7]. Thus, equation (3-14) can be written:

$$\boldsymbol{v_{Cbr}} = \boldsymbol{Mv_C}. \tag{3-14}$$

Where $\boldsymbol{v_{Cbr}}$ is the vector of the capacitor voltage in all the branches, and its dimensions are equal to the number of branches b. In contrast, $\boldsymbol{v_C}$ is the vector containing the values of the voltage of the capacitors connected, and its dimension is equal to the number of capacitors connected in the system.

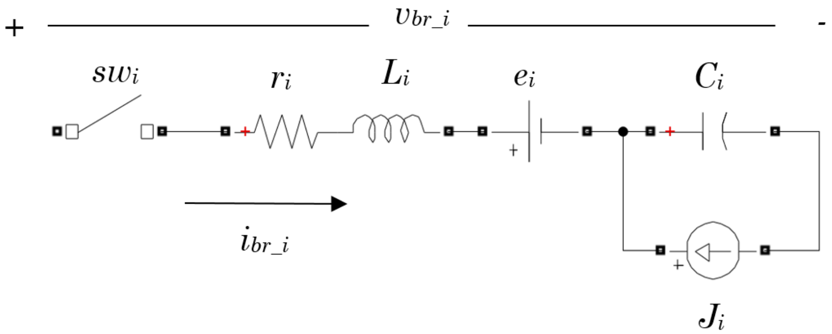


Fig. 3-11. General branch configuration [3].

Substituting equations (3-12) and (3-14) in (3-11) and re-ordering [7]:

$$\boldsymbol{B_b R\, B_b^T i_x} + \boldsymbol{B_b L B_b^T} \frac{d\boldsymbol{i_x}}{dt} + \boldsymbol{B_b M v_C} = -\boldsymbol{B_b W}^T \boldsymbol{v_p}. \tag{3-15}$$

The next equation can be derived from the equation for the capacitor charge.

$$q_C = Cu_C. \tag{3-16}$$

In order to introduce equation (3-16) in the system of equations, it needs to be modified as in equation (3-17):

$$\boldsymbol{q_C} = \boldsymbol{M}^T \boldsymbol{CMv_C}. \tag{3-17}$$

The charge of the capacitor can also be expressed in terms of the branch current as follows:

$$\boldsymbol{q_c} = \boldsymbol{M}^T \boldsymbol{B_b^T} \int \boldsymbol{i_x}. \tag{3-18}$$

Substituting equation (3-17) into (3-18) and deriving:

$$\boldsymbol{M}^T \boldsymbol{CM} \frac{d\boldsymbol{v_C}}{dt} - \boldsymbol{M}^T \boldsymbol{B_b^T i_x} = \boldsymbol{0}. \tag{3-19}$$

Equations (3-15) and (3-19) constitutes the Modified State Space representation of the system. Thus, the system is defined as follows:

$$\begin{bmatrix} \boldsymbol{M}^T\boldsymbol{CM} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B_b L B_b^T} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}}_c \\ \dot{\boldsymbol{i}}_x \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & -\boldsymbol{M}^T\boldsymbol{B_b^T} \\ \boldsymbol{B_b M} & \boldsymbol{B_b R\, B_b^T} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_c \\ \boldsymbol{i}_x \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{B_b W}^T \end{bmatrix} \boldsymbol{v_p}. \tag{3-20}$$

The number of nodes that usually contains distribution systems requires to automate the process for implementing equation (3-20). In the next section an algorithm for this purpose is presented.

3.3 Implementation of the Modified State Space representation using MATLAB™

In the later section, the Modified State Space representation presented in [3], [7] was described. However, the elaboration of this model for the IEEE-123 Node Test Feeder requires a level of automation; otherwise, the process would imply an unacceptable amount of time and may also be exposed to typing errors. Therefore, an algorithm for developing the Modified State Space representation of the IEEE-123 Node Test Feeder is required.

The algorithm is elaborated using MATLAB™ code editor. The algorithm has two main routines: the first routine is dedicated to build the node incidence matrix, and the second routine manipulates the node incidence matrix to create matrix $\boldsymbol{B_b}$ as was shown in equations (3-2) to (3-6). The second routine also gathers the computed matrices to condense them in the form of equation (3-20).

3.3.1 First routine: Node incidence matrix construction

The construction of the node incidence matrix is divided into four sub-node-incidence matrices: line segments, loads, capacitors, and sources. This subdivision is convenient since the information for each of these elements is presented in different tables, and the information in these tables is arranged slightly different. The resulting matrices are then merged into one node incident matrix for the overall system.

The information of the line segments presented in [2] is first charged into a variable named “SegLine”. A section of this table was presented in Fig. 3-2 as a matter of sample and is repeated in Table 3-2 for convenience of the reader.

The five matrices shown in Fig. 3-12 are declared. These matrices correspond to the non-three-phase line segment configuration, and they are going to be used to place the elements of the node incident matrix in the right position.

Next, the names of all nodes are extracted from the first and second column of “SegLine” and storage in the variable named “Nname”. This step is necessary since the nodes are not numbered following a specific pattern. Then, a position number is assigned to each node.

The positions are spaced 3 digits apart because all the nodes, at this point, are treated as three-phase nodes. For instance, the first node will be assigned the position 1, and the second node will be assigned the position 4. The names of the nodes and their position are merged in the matrix named “Npos”.

The next portion of the algorithm reads the number of nodes where the line segment is connected. Then, the position of these nodes is extracted from “Npos”. Next, the configurations of the line segments are evaluated, if the configuration corresponds to a three-, two- or single-phase node, a matrix of configuration, similar to the ones shown in Fig. 3-12, is allocated in the position determined by the extracted position number.

Table 3-2. Line Segment Data sample table [2].

Node A	Node B	Length (ft.)	Config.
1	2	175	10
1	3	250	11
1	7	300	1
3	4	200	11

A flow chart of the algorithm described above is shown in Fig. 3-13. The output of the algorithm is the matrix named “A1”. This matrix corresponds to the portion of the overall node incidence matrix, contributed by the line segments in the distribution system.

Similar approaches are used to build the matrices corresponding to the portion of the node incidence matrices contributed by the load and the capacitor banks. In addition, together with these matrices, the matrices of resistance, inductance and capacitance are built using the same algorithms. The script of the algorithm in MATLAB™, with detailed commentaries, is presented in Appendix B.1.

The last portion of the algorithm merges the node incidence matrices contributed by line segments, load, capacitor and sources into one over-all incidence matrix. Once this matrix is built, it is required to delete all-zero rows and columns. These all-zero rows and columns are present in the matrix since, at the beginning, all nodes were considered three-phase nodes, and the nodes that are in fact non-three-phase will add zeros to the matrix. The algorithm takes advantage of the MATLAB™ command “any” which suppresses the all-zero rows or columns. The main output of the algorithm is the node incidence matrix of the system.

$$Config7 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Config8 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$Config9 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$Config10 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$Config11 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 3-12. Non-three-phase configuration matrices.

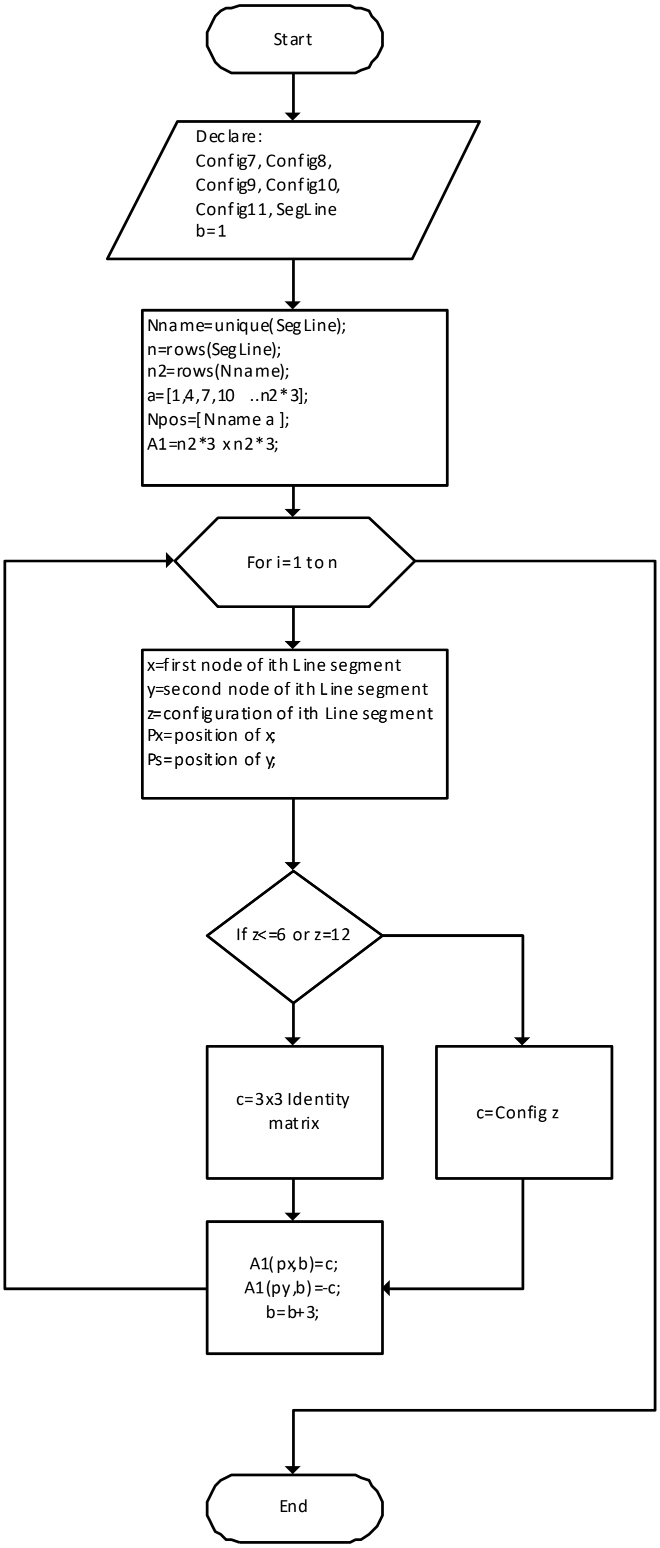


Fig. 3-13. Flow chart of the algorithm for node incident matrix corresponding to line segments.

Up to this point, the dimension of the system can be known from the dimension of the resulting node incident matrix. The output of the algorithm described in this section condensed all the matrices required to proceed with the composition of equation (3-20), and the process is explained in the following section.

3.3.2 Second routine: State space matrices construction

This second routine manipulates the node incidence matrix resulting from the first routine to create the basic loop matrix $\boldsymbol{B_b}$. Once this is completed, all the components for putting together the Modified State Space representation can be accommodated in the form of equation (3-20). After all the matrices in equation (3-20) are created, the MATLAB™ command "dss" is used to create a linear time invariant system out of these matrices. The script of the algorithm in MATLAB™, with detailed commentaries, is presented in Appendix B.2.

3.4 Comparison between the MATLAB/SIMULINK™ SimPower System™ model and the Modified State Space representation

The resulting Modified State Space model was implemented in MATLAB/SIMULINK™ using a linear time invariant block (LTI block). The SimPower System™ model developed in previous sections and the Modified State Space model are set in the same MATLAB/SIMULINK™ file. Fig. 3-14 shows the setup of the simulation. The input of the LTI block is the same voltage source connected to the same node as in the SimPower System™ model developed in section 3.1. The output of the LTI block is the three-phase current of the voltage source. The currents are plotted using the same scope.

Fig. 3-15 to 3-17 show the current waveforms of phase-A, phase-B and phase-C respectively. The currents are overlapping, which indicates there is a match between the two models. It can be concluded that the Modified State Space representation and the procedure applied to implement it are correct.

3.5 Krylov subspace-based model order reduction implementation in MATLAB™

Before describing the procedure, is pertinent to state that, the selected model order reduction method consists of a Block Krylov subspace method with the implementation of the Block Arnoldi algorithm for computing the basis of the subspace. The mathematical foundation of this method was presented in the previous chapter section 2.4; however, some of the equation are repeated here using the notation adopted during the derivation of the Modified State Space model in section 3.4.

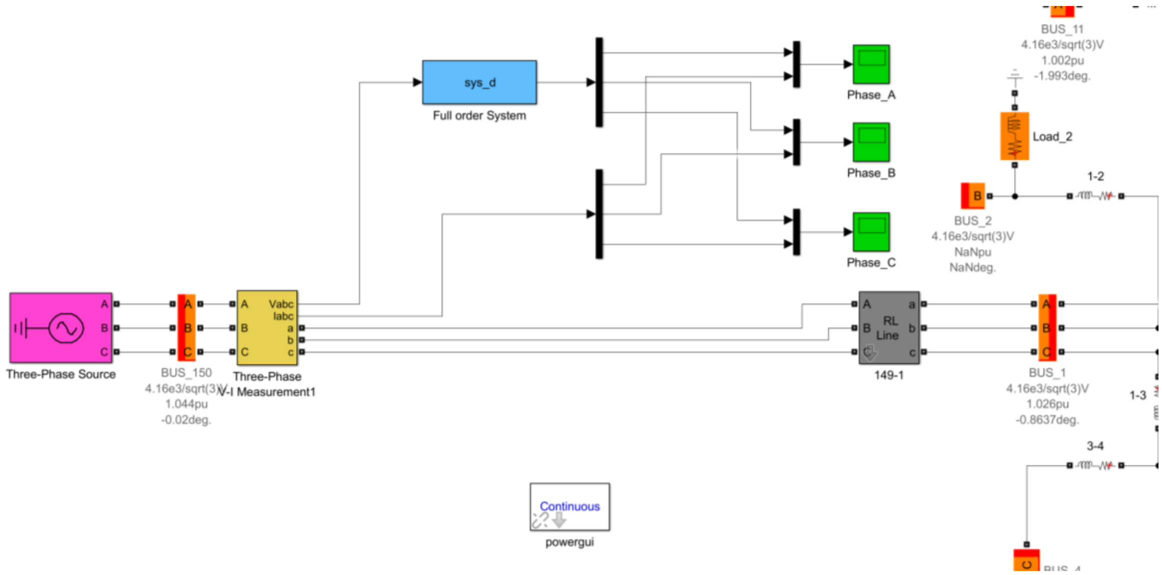


Fig. 3-14. SimPower System™ model and Modified state space model simulation setup.

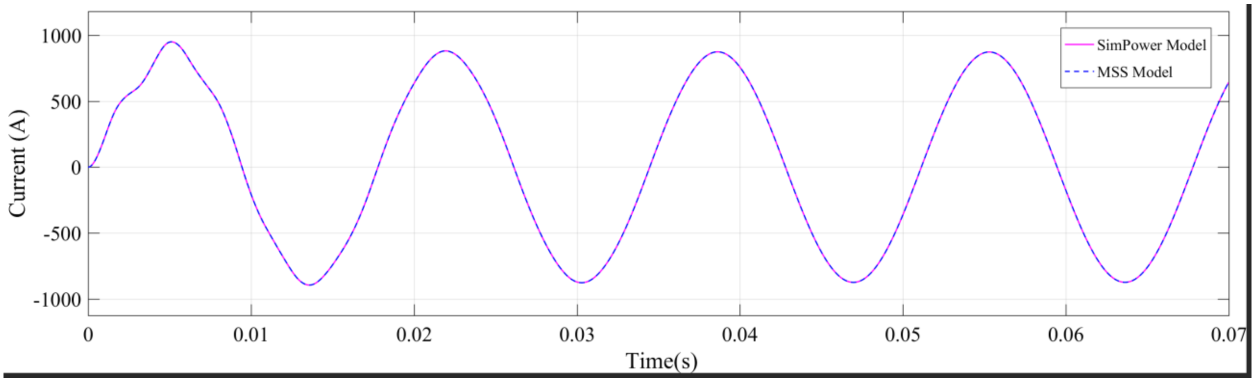


Fig. 3-15. SimPower System™ and MSS current waveforms corresponding to phase A.

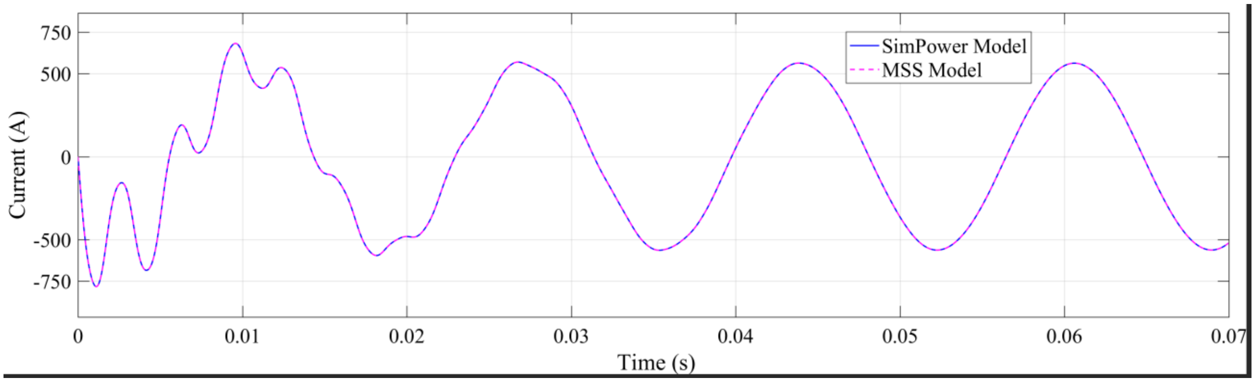


Fig. 3-16. SimPower System™ and MSS current waveforms corresponding to phase B.

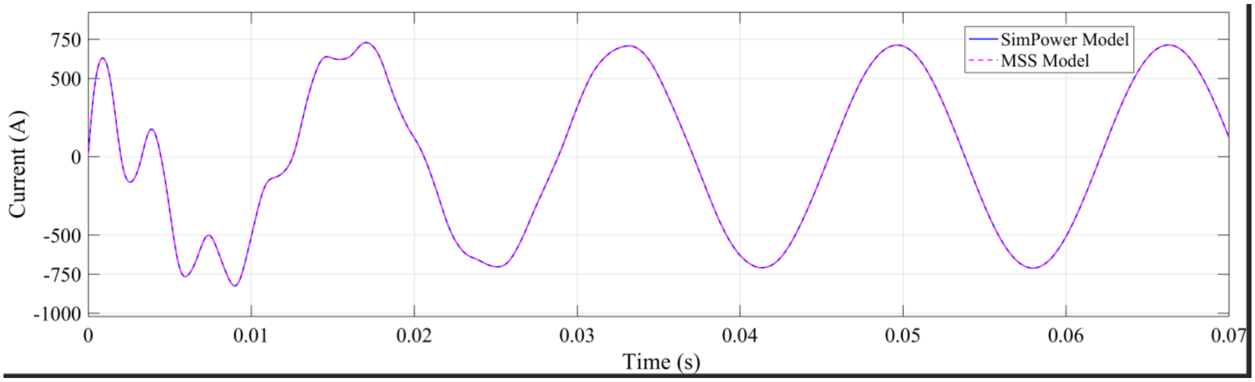


Fig. 3-17. SimPower System™ and MSS current waveforms corresponding to phase C.

The selected method is implemented to reduce the system (3-20). To avoid confusion, the system is re-stated with the nomenclature used in MATLAB™ script:

$$\boldsymbol{C}\dot{\boldsymbol{x}} + \boldsymbol{G}\boldsymbol{x} = \boldsymbol{B}\boldsymbol{u}\,; \boldsymbol{y} = \boldsymbol{L}^{\boldsymbol{T}}\boldsymbol{x}. \tag{3-21}$$

The steps in the algorithm are described as follows:

- Calculating the starting vector in the Krylov subspace

  ```
  R=G^-1*B;
  V0=orth(R);
  ```

  The MATLAB™ function "orth" calculated the orthonormal basis of $\boldsymbol{R}$.

- The desired order of the reduced model is stated

  ```
  q=30;
  [row,col]=size(B);
  n=q/col;
  ```

  The integer q is required to be a multiple of the number of column of the input or output matrices [18]. In this case, the number of columns of both matrices are equal. Thus, the resulting basis matrix is constituted by *n* matrices which dimensions are *N* x *col.*

- The matrix $\boldsymbol{Tq_i}$ is calculated

  ```
  for k=1:n
      eval(['X=C*V',int2str(k-1),';']);
      eval(['V0',int2str(k),'=G^-1*X;']);
  ```

  This step is equivalent to:

  $$\boldsymbol{V_{0k}} = \boldsymbol{G^{-1}Cq_{k-1}}. \tag{3-22}$$

- The inner product of the vectors $\boldsymbol{q_j}$ and $\boldsymbol{Tq_i}$, and the matrix $\sum_{\boldsymbol{j=1}}^{\boldsymbol{i}} \boldsymbol{q_j} h_{ji}$ are calculated:

  ```
  for j=1:k
  ```

```
        eval(['H=transpose(V',int2str(k-j),')*V',int2str(j-
1),int2str(k),';']);
        eval(['V',int2str(j),int2str(k),'=V',int2str(j-
1),int2str(k),'-V',int2str(k-j),'*H;']);
end
```

This step is equivalent to a Gram-Schmidt orthogonalization of the new columns with respect to all the previous ones

$$h = \boldsymbol{V}_{\boldsymbol{k-1}}^{\boldsymbol{T}} \boldsymbol{V}_{\boldsymbol{j-1},\boldsymbol{k}}, \tag{3-23}$$

$$\boldsymbol{V}_{\boldsymbol{j},\boldsymbol{k}} = \boldsymbol{V}_{\boldsymbol{j-1},\boldsymbol{k}} - \boldsymbol{V}_{\boldsymbol{k-j}} h. \tag{3-24}$$

- Normalizing the rest of the elements of $\boldsymbol{V}_{\boldsymbol{j},\boldsymbol{k}}$

```
eval(['V',int2str(k),'=orth(V',int2str(k),int2str(k),');']);
```

This step is equivalent to

$$\boldsymbol{V}_{\boldsymbol{k}} = \frac{\boldsymbol{V}_{\boldsymbol{kk}}}{\|\boldsymbol{V}_{\boldsymbol{kk}}\|_2}. \tag{3-25}$$

- Finally, the basis matrix is built from the calculated matrices $\boldsymbol{V}_{\boldsymbol{k}}$ (k=1, 2, 3…n)

```
V=V(:,1:q);
```

- The matrices of the reduced system are computed:

```
Cr30=V'*C*V;
Gr30=V'*G*V;
Br30=V'*B;
Lr30=V'*L;
sys_dred2_1=dss(-Gr30,Br30,Lr30',[0],Cr30);
[Asysred,Bsysred,Csysred,Dsysred]=ssdata(sys_dred2_1);
```

The complete MATLAB™ script can be found in Appendix B.3.

3.6 Model order reduction of the IEEE-123 Nodes Test Feeder

The reduced model resulting from the implementation of the MATLAB™ script described in the previous section was simulated in MATLAB/SIMULINK™. The output waveforms corresponding to the original and reduced systems were compared. Fig. 3-18 shows the setup used for the simulation. The original system was reduced to three sizes: 30-, 24- and 12-order. Each of these reduced systems were computed from the original system and not in cascade. Fig. 3-18 simultaneously shows the output current of phase-A of the original system and the three reduced systems. In a first glance, the reduced systems seem to accurately follow the behavior of the original; however, the box in the upper right corner of Fig. 3-19 shows a zoom-in of the first peak of the waveform.

The deviated lines correspond to the waveforms of the smaller 24- and 12-order systems. This higher deviation is expected since the smaller the size of the reduced system, the less moments of the original system are emulated by the reduced system.

Fig. 3-20 shows a deeper zoom-in of Fig. 3-19. In this figure, the deviation of the 30-order systems is better appreciated.

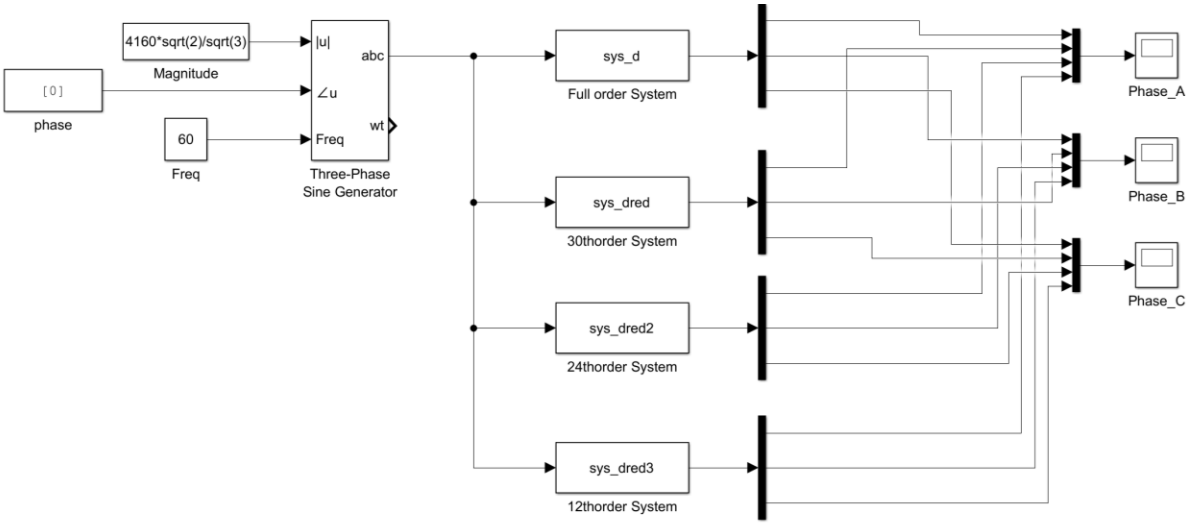


Fig. 3-18. MATLAB/SIMULINK™ setup for comparing reduced models.

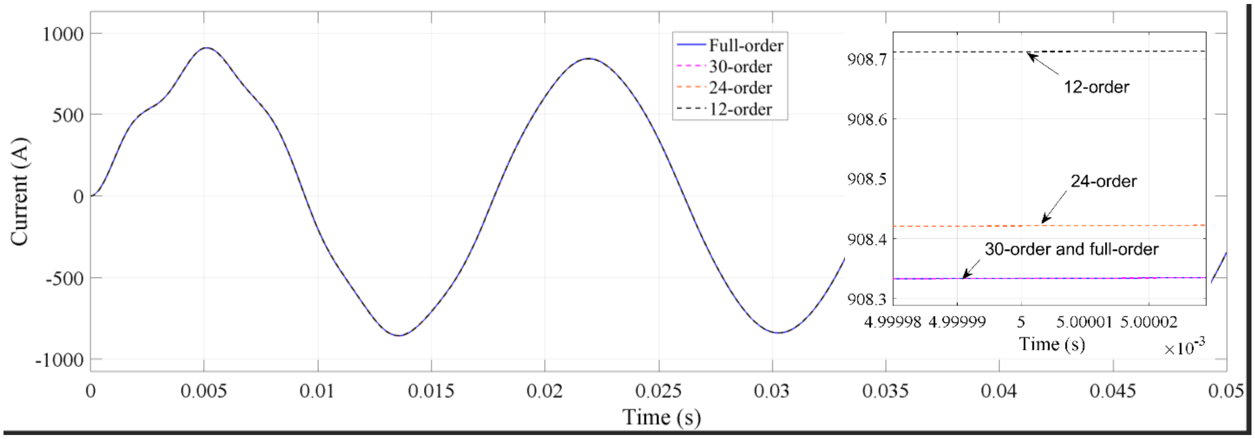


Fig. 3-19. Waveform of output Current of Phase A of full and reduced systems.

The waveform of the 30-order system closely follows the waveform of the full order system. Similar behavior can be observed for the case of phase-B and C. The corresponding waveforms can be found in Appendix C.1.

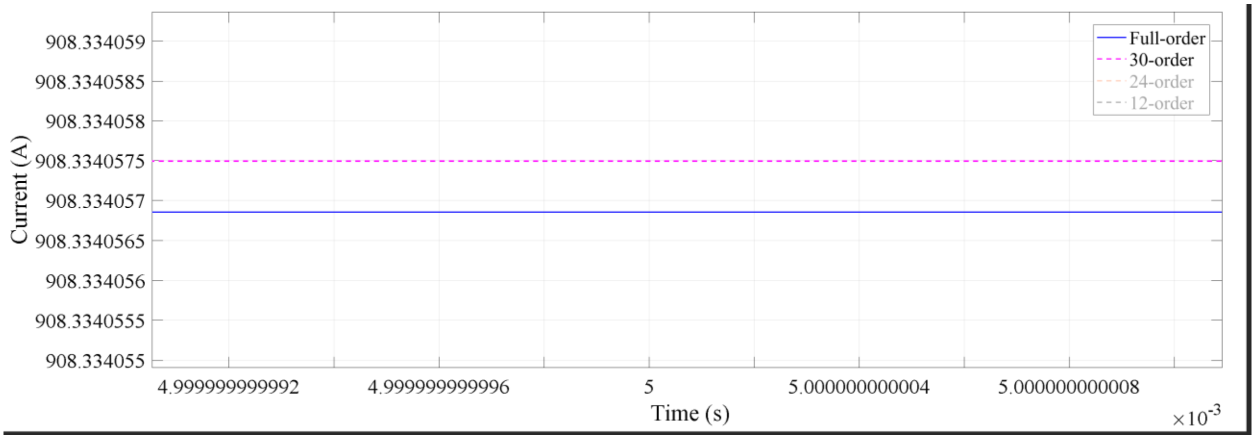


Fig. 3-20. Waveform of output Current of Phase A of full- and 30-order systems.

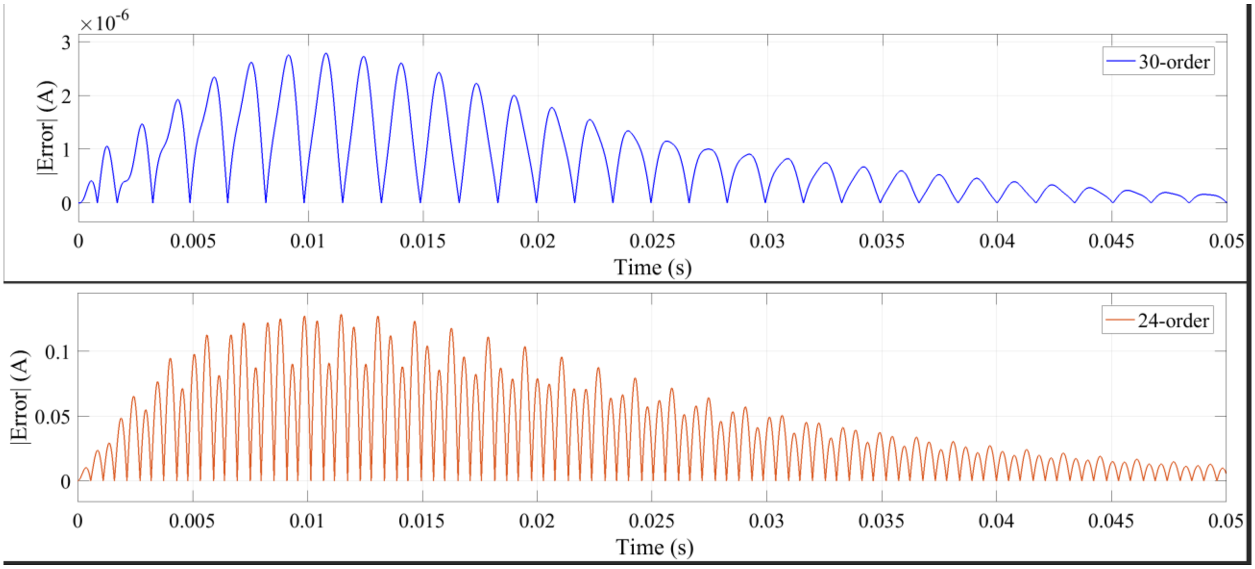


Fig. 3-21. Absolut value of error between full-order system and 30- and 24-order systems.

Table 3-3 shows the reduction in the computation time for the full-order, 30-order and the 12-order system. The simulation was ran using a fixed step of 1e-5s time step and the ode23 solver.

The absolute value of the errors of the 30- and 24-order models is shown in Fig. 3-21. As can be observed in this figure, the error from the 30-order system is in the range of 10^{-6}, and the error from the 24-order system is in the range 0.1Amps, as mention before. For both reduced systems, the error starts decreasing while the system reaches steady state values. This behavior is expected since the simulation was run with initial conditions set to zero; therefore, the oscillations of the waveforms in the firsts cycles of the simulation. Once the storage elements in the system are charged, the oscillations decrease and the reduced system follows closer the pure sinusoidal behavior of the full-order system.

Table 3-3. Computation time reduction with ideal voltage source.

Order	Full-order	30-order	12-order
Computation time (s)	31.21	27.61	26.58
Reduction (%)	-	11.53	14.83

Another interesting behavior of the reduced systems is their frequency response. The magnitude response of the full and reduced systems are shown in Fig. 3-22. Because the systems are multiple inputs multiple outputs (MIMO), the transfer function of one input against one output should the used to plot the magnitude response. In the case of Fig. 3-22, the magnitude response of phase A voltage, as input, against phase A current, as output, is shown.

It is evident that the 12-order system poorly performs since it continues decreasing in a straight line missing the resonant mode around 6000 rad/s present in the full-order system. The magnitude response of the ratio between the transfer function of the full-order system and the transfer functions of the reduced order systems is presented in Fig. 3-23. This figure helps to visualize the deviation of the reduced systems with respect to the full-order system over a range of frequencies. It can be observed that all reduced systems deviate from the full-order system about 6000 rad/s where a resonant mode is present, and, as mentioned before, the 12-order system is the lest accurate.

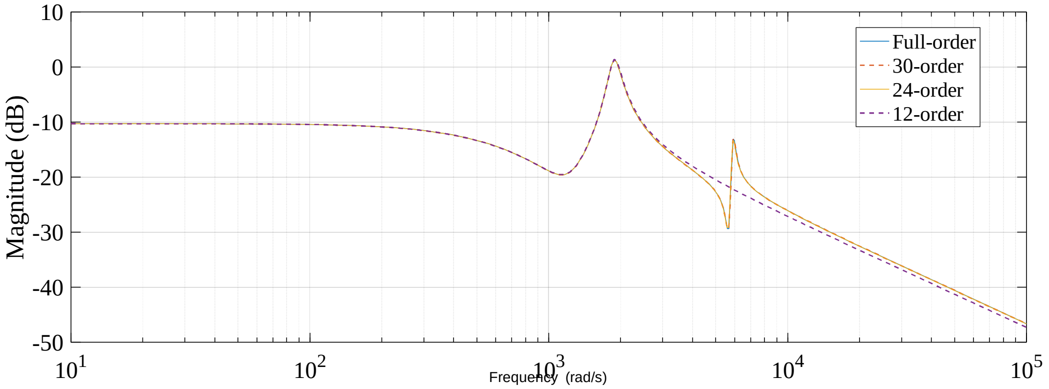


Fig. 3-22. Magnitude response of full and reduced systems for Phase A.

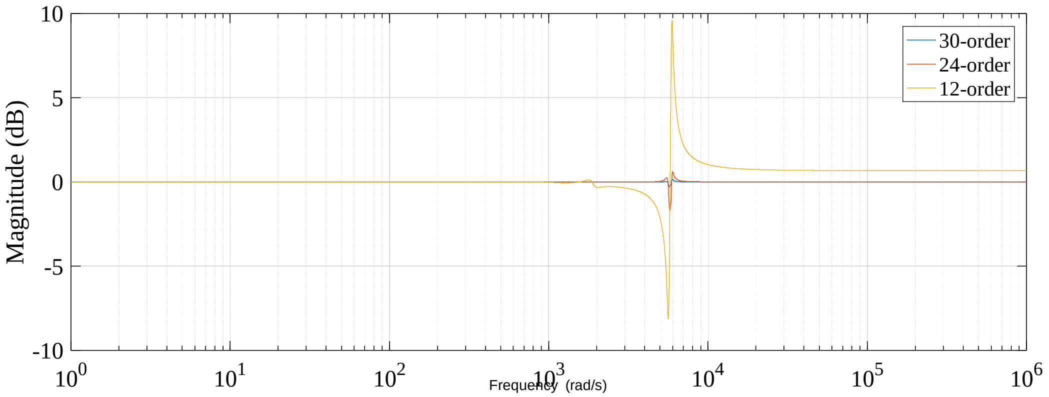


Fig. 3-23. Magnitude response difference between full- order and reduced systems.

The deficient performance of the reduced systems at high frequencies is expected since, during the derivation of the Krylov subspace basis, the moments of the system were calculated around s=0 rad/s. Thus, the performance of the reduced systems is expected to be better at low frequencies than at high frequencies. In addition, the smaller reduced systems deviate more because they do not match as much moments as with bigger reduced systems. These two phenomena can be observed comparing the eigenvalues of the 24-order and 12-order system listed in Table. 3-4.

The eigenvalues of the 24-order system are shown on the left of Table 3-4. The 24-order system has six eigenvalues about 6000 rad/s (highlighted in yellow) that the 12-order system derived around s=0 rad/s (on the right top of Table 3-4) does not have.

On the other hand, the 12-order reduced system was calculated, this time around s=6000 rad/s. This new derivation is calculated replacing $\boldsymbol{G}$ by $(\boldsymbol{G} - s\boldsymbol{C})$ when calculating the starting vector $\boldsymbol{R}$. The eigenvalues of this new 12-order system are listed in the right bottom of Table 3-4. Now the system contains high frequency eigenvalues (highlighted in yellow).

Table 3-4. Eigenvalues of the 24-order and 12-order around s=0 rad/s and s= 6000rad/s.

Eigenvalues of 24-order system Phase-A (1x10^3)	Eigenvalues of 12-order system Phase-A (1x10^3)
	-0.1134 + 1.8824i
-0.0994 + 5.8856i	-0.1134 - 1.8824i
-0.0994 - 5.8856i	-0.6754 + 0.0000i
-0.1120 + 1.8770i	-0.4901 + 0.0000i
-0.1120 - 1.8770i	-0.1082 + 1.8705i
-0.7385 + 0.0000i	-0.1082 - 1.8705i
-0.6711 + 0.0000i	-0.7025 + 0.0000i
-0.5146 + 0.0000i	-0.5094 + 0.0000i
-0.4820 + 0.0000i	-0.1124 + 1.8927i
-0.1018 + 5.5698i	-0.1124 - 1.8927i
-0.1018 - 5.5698i	-0.7004 + 0.0000i
-0.1083 + 1.8611i	-0.5085 + 0.0000i
-0.1083 - 1.8611i	
-0.7636 + 0.0000i	Eigenvalues of 12-order system Phase-A (1x10^3) around s=6000 rad/s
-0.6942 + 0.0000i	
-0.5261 + 0.0000i	-0.0732 + 4.7106i
-0.5090 + 0.0000i	-0.0732 - 4.7106i
-0.1024 + 5.3151i	-0.3581 + 0.8420i
-0.1024 - 5.3151i	-0.3581 - 0.8420i
-0.1136 + 1.8836i	-0.0832 + 4.3820i
-0.1136 - 1.8836i	-0.0832 - 4.3820i
-0.7512 + 0.0000i	-0.3210 + 1.0484i
-0.6880 + 0.0000i	-0.3210 - 1.0484i
-0.5599 + 0.0000i	-0.0816 + 3.9471i
-0.5081 + 0.0000i	-0.0816 - 3.9471i
	-0.3529 + 0.9044i
	-0.3529 - 0.9044i

The magnitude response of the 12-order system derived around s=6000 rad/s, the 12-order system derived around s=0 rad/s, and the full-order system are presented in Fig. 3-24. The magnitude response of the 12-order system derived around 6000 rad/s begins to follow the behavior of the full-order system. Nevertheless, the deviation is still considerable around 6000 rad/s. It is noticeable that the deviation after 6000 rad/s is reduced compared with the 12-order system derived around s=0 rad/s.

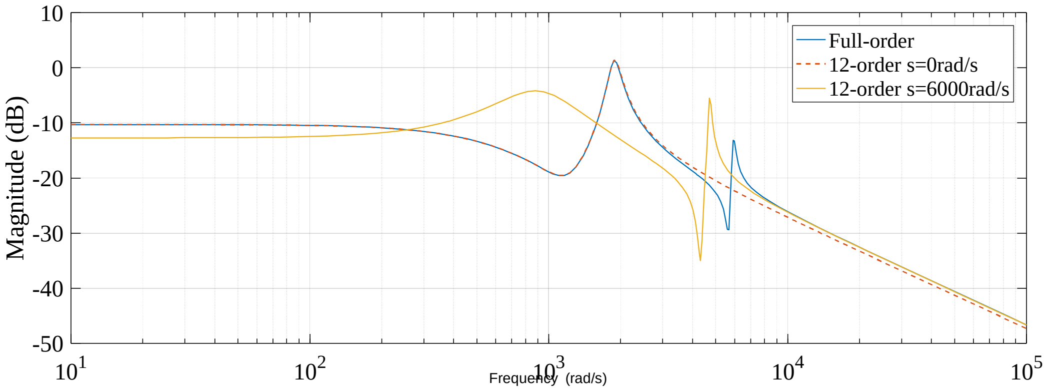


Fig. 3-24. Magnitude response of full-system and 12-order systems around s=0 rad/s and s=6000 rad/s.

However, moving the interpolation point to 6000 rad/s compromises the behavior of the reduced system at low frequencies as can be observed below 250 rad/s in Fig. 3-24. Thus, the point around which the moments of the system are derived will depend of the application and the frequency characteristic of the phenomena of interest. In the case of power system, the moments are usually derived around s=0 rad/s since low frequency modes of the original system are sought to be reproduced in the reduced system [20].

3.7 Model order reduction by using two-sided Arnoldi

In the previous section, the model order reduction of the test feeder was carried out using an one-sided Arnoldi algorithm similar to the one presented in [27] with the difference that the interpolation point is fixed to zero. This algorithm uses only the input Krylov subspace $K(\boldsymbol{A}^{-1}\boldsymbol{E}, \boldsymbol{A}^{-1}\boldsymbol{B})$ for the formulation of the basis.

However, further improvement in the accuracy of the reduced system, over a range of orders, can be obtained by using a two-sided Arnoldi including the output Krylov subspace $K(\boldsymbol{A^{-T}E^{T}}, \boldsymbol{A^{-T}C^{T}})$ [18]**.** The basis of the output Krylov subspace will be the left-side projection $\boldsymbol{W}$ for the reduced system, as was mention in chapter 2 and repeated here for convenience.

$$\boldsymbol{E_r} = \boldsymbol{W^T E V};\ \boldsymbol{A_r} = \boldsymbol{W^T A V}\ ;\ \boldsymbol{B_r} = \boldsymbol{W^T B}\ ;\ \boldsymbol{C_r} = \boldsymbol{CV}. \qquad (3\text{-}26)$$

The construction of the basis $\boldsymbol{W}$ is performed by using the algorithm described in section 3.5. Fig. 3-25 shows the error corresponding to the one-sided 30-order and the two-sided 30-order reduced systems. The error presents a considerable reduction compared with the error obtained using the approach presented in [7], [27].

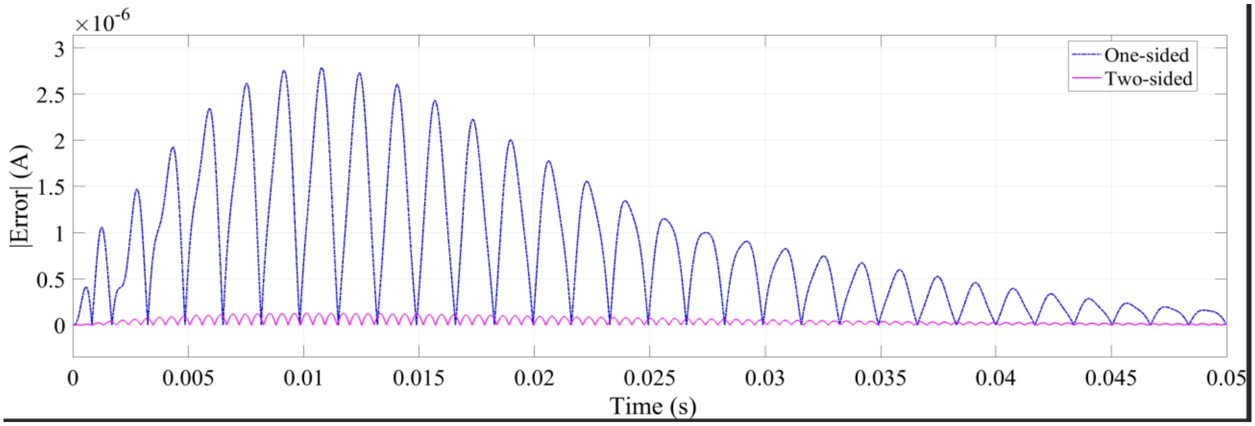


Fig. 3-25. Absolut error of the 30-order one sided and two-sided reduced systems

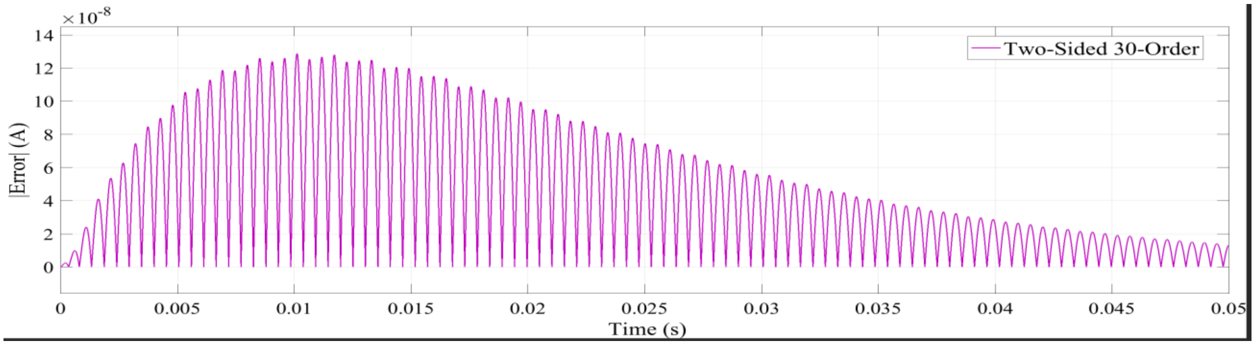


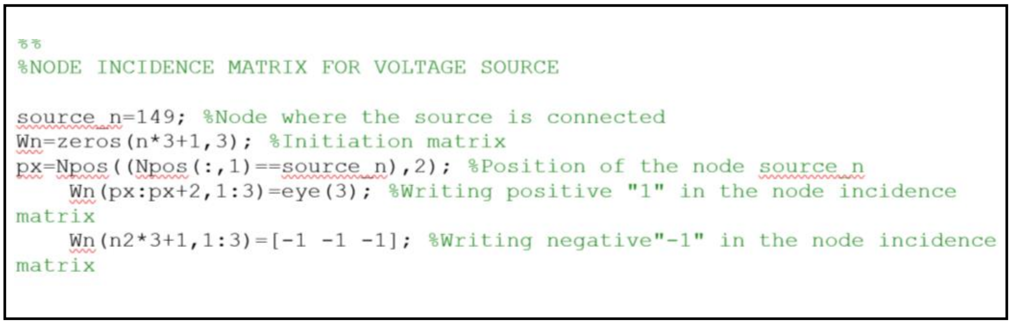
Fig. 3-26. Absolut error of 30-order two-sided reduced system.

3.8 Reduced order model for PV source transient simulation.

Similar to [7], a transient in a PV source connected to the system is simulated using the reduced order model. The PV source is adopted from the MATLAB/SIMULINK™ sample “250-kW Grid-Connected PV Array” in the samples library. The output voltage of the PV source was change from the original 25kV to 4.16kV to match the voltage of the test feeder. The source is connected to the node-450 on the test feeder. In order to include this new source to the reduced system, the number of the node is added to the entry “source_n” containing the source node in the algorithm for the Modified State Space in Appendix B.1. Fig. 3-27 shows the specific section of the algorithm.

The irradiance is varied to produce the transient in the system. The irradiance is dropped from 1000W/m^2 to 500W/m^2 at the time 0.5s and is increased again to 1000 W/m^2 at time 0.8s. The line segment between node-100 and 450 is used as tie line in order to simulate the reduced order system LTI block with the SimPower System™ PV source model.

The two systems required to be connected in a double feedback configuration as is shown in Fig. 3-28. The voltage at node-100 produced by the reduced order LTI block is fed into the PV model while the voltage at node-450 produced by the PV model in fed into the LTI block.



```
%%
%NODE INCIDENCE MATRIX FOR VOLTAGE SOURCE

source_n=149; %Node where the source is connected
Wn=zeros(n*3+1,3); %Initiation matrix
px=Npos((Npos(:,1)==source_n),2); %Position of the node source_n
    Wn(px:px+2,1:3)=eye(3); %Writing positive "1" in the node incidence
matrix
    Wn(n2*3+1,1:3)=[-1 -1 -1]; %Writing negative"-1" in the node incidence
matrix
```

Fig. 3-27. Scrip for the generation of the node incidence matrix corresponding to sources.

Since the output of the LTI block are the currents of the sources (ideal source at node-149 and the PV source at node-450), the voltage at the node-100 should be calculated as a combination of the branches voltage. Another option is to connect a small capacitor to the node-100.

The voltage of the node-100 is computed by applying the first option combining equations (3-27), (3-28) and (3-29), and selecting and adding the voltage of the branches between node-100 and ground with the use of constant matrix $\boldsymbol{C}200$ as follows

$$\boldsymbol{B_b v_{br}} = \boldsymbol{B_b R\, B_b^T i_x} + \boldsymbol{B_b L B_b^T} \frac{d\boldsymbol{i_x}}{dt} + \boldsymbol{B_b v_{Cbr}} + \boldsymbol{B_b e}, \tag{3-27}$$

$$\boldsymbol{B_b L B_b^T} \frac{d\boldsymbol{i_x}}{dt} + \boldsymbol{B_b M v_c} + \boldsymbol{B_b R\, B_b^T\, i_x} = -\boldsymbol{B_b W^T\, v_p}, \tag{3-28}$$

$$\boldsymbol{y} = \boldsymbol{C}200 * \boldsymbol{v_{br}} = \boldsymbol{C}200 * \left(\boldsymbol{C_y} \begin{bmatrix} \boldsymbol{v_c} \\ \boldsymbol{i_x} \end{bmatrix} + \boldsymbol{D_y v_p}\right), \tag{3-29}$$

where $\boldsymbol{C}200$ is of the form

$$[0_{\boldsymbol{br1}} \quad 0_{\boldsymbol{br2}} \quad \cdots \quad \boldsymbol{I_{bri}} \quad \cdots \quad \boldsymbol{I_{brj}} \quad \cdots]. \tag{3-30}$$

The resulting setup is presented in Fig. 3-28 and 3-29. In this configuration, the output of the reduced system and the full-order system is compared, and the resulting waveforms,

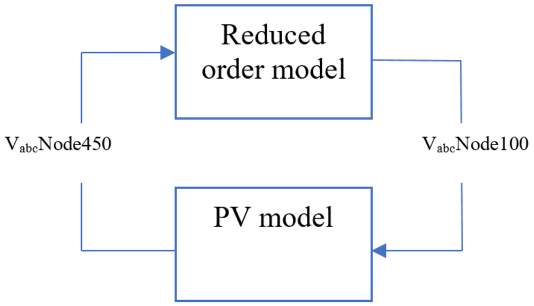


Fig. 3-28. Double feedback configuration for the simulation of the PV source.

corresponding to the current of phase-A, is shown in Fig. 3-30. The deviation of the reduced system is better appreciated in Fig. 3-31.

The current of the PV source decreases once the irradiance drops. This behavior can be considered as the effect shading may have over the power out of the PV source. At the time 0.8, the irradiance is increased to its original value. During this transient, the reduced model approximately matches waveform of the original system. However, the small deviation of the reduced system is better appreciated in Fig. 3-29.

However, with this configuration, the error between the reduced model and the SimPower System™ model is drastically increased. This error is shown in Fig. 3-32. The error approximately oscillates between 0 and 1.5 Amp during steady-state operation. This represents 1.67% of the magnitude of the current. Several tests were performed with this setup; however, the root cause of this deviation was not determined. Further work is required for this matter.

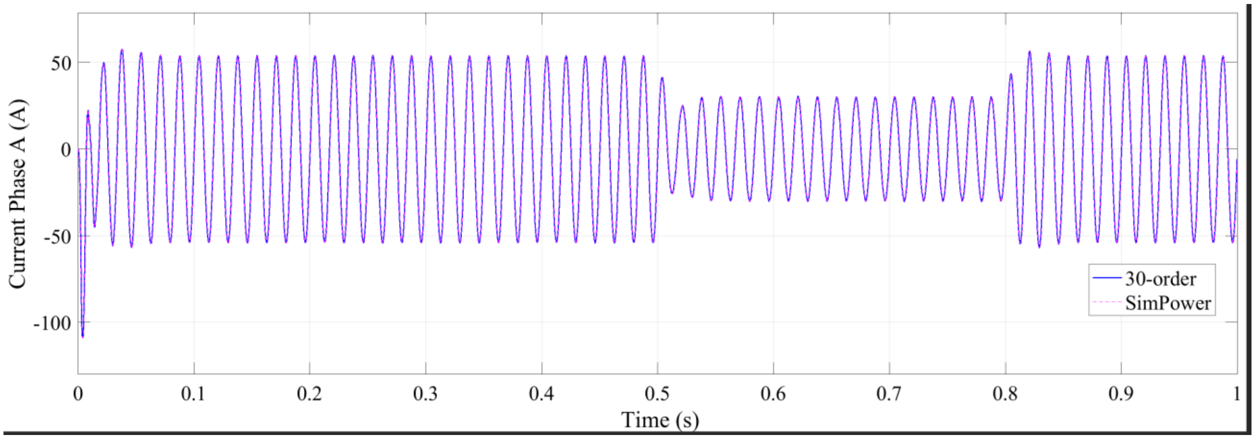


Fig. 3-29. PV source output current of the full- and 30-order system.

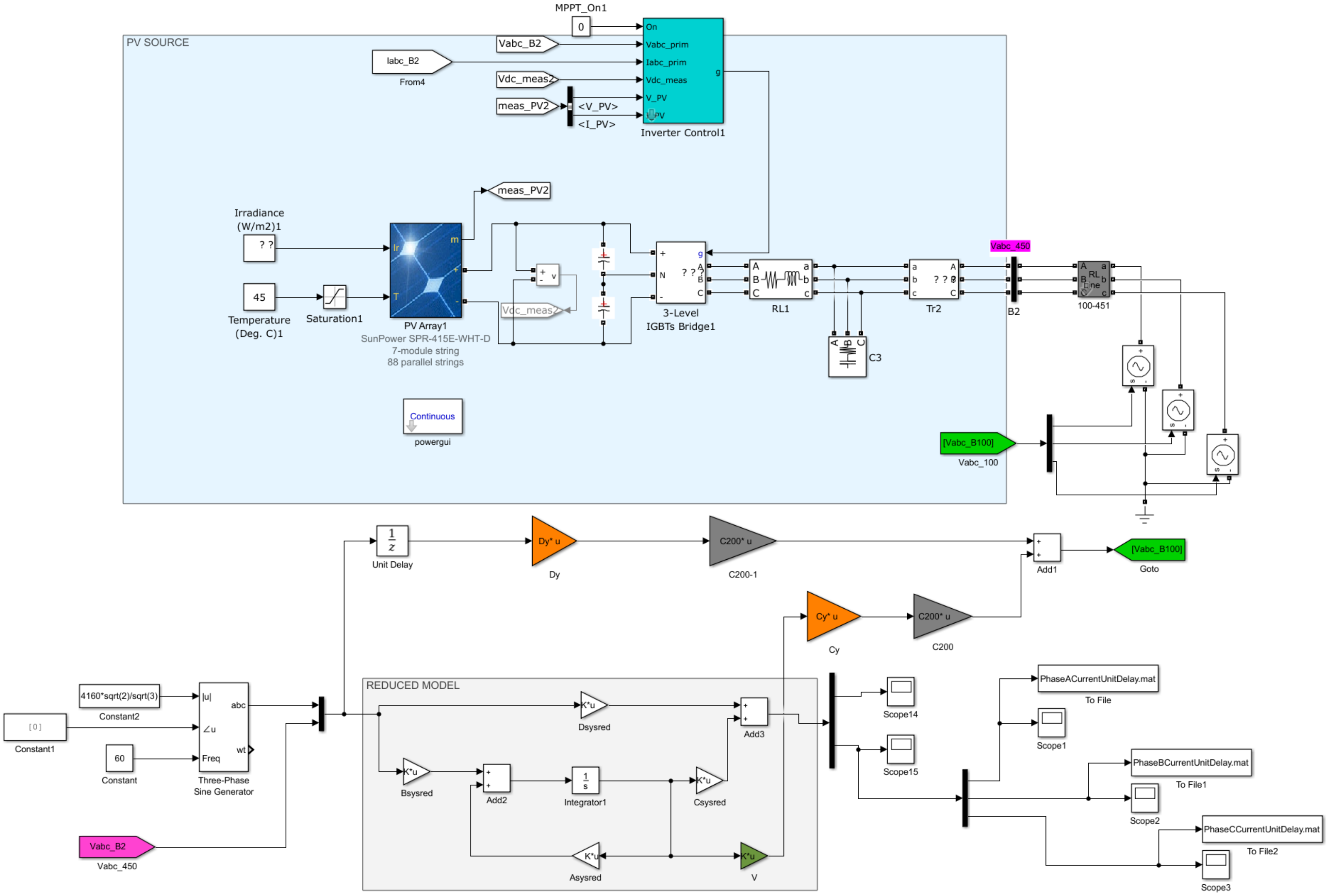


Fig. 3-30. Double feedback configuration for the simulation of the PV source in MATLAB/SIMULINK™.

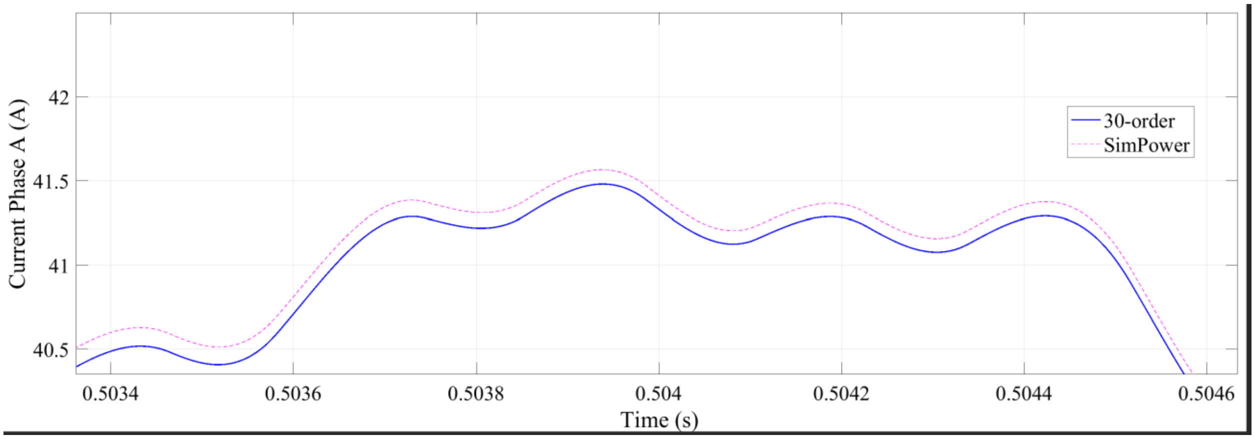


Fig. 3-31. Zoom-in of the PV source output current of the full- and 30-order system.

An interesting result is obtained when the order of the system was increased to 60. Fig. 3-33 shows the absolute value of the error between the full order and the 60-order system simulated with the setup presented in Fig. 3-29. The deviation of the systems starts with a low value, and at approximately 0.1413s, the deviation suddenly increases. A zoom-in of this section is presented in Fig. 3-34. It can be observed that the deviation before the increment is of the order of 10^{-4}A, and after oscillates between zero and 1.5A approximately. Some modifications to the simulation setup were implemented:

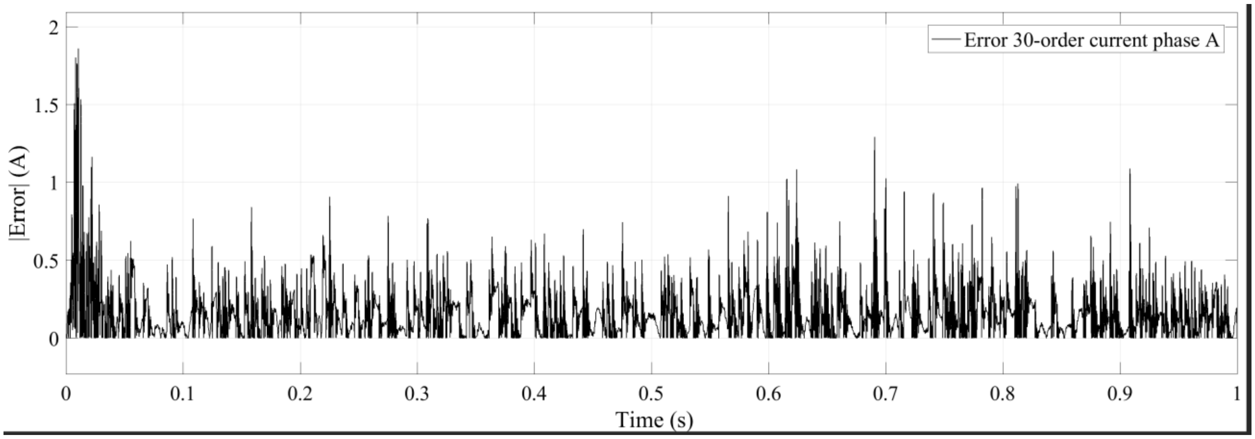


Fig. 3-32. Error between 30-order reduced model and SimPower System™ model.

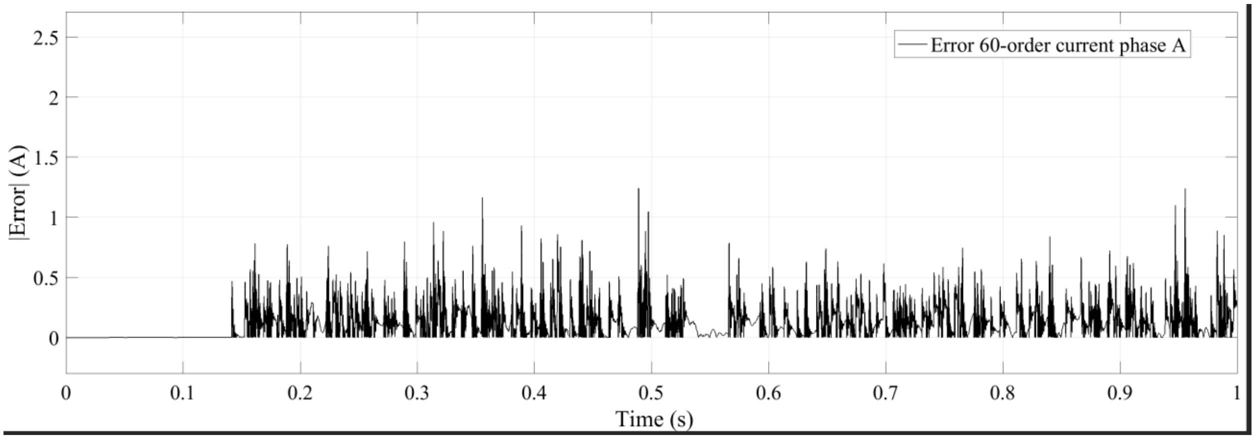


Fig. 3-33. Error between the 60-order reduced model and SimPower System™ model.

- Decreasing of the unit delay in the algebraic loop containing $\boldsymbol{D_y}$.
- Fix step solver.
- Changing the tolerance of the solver.
- Increasing the order of the system.

However, the root cause of this behavior is not clear since none of these modifications correct the increment in the deviation.

Table 3-5 shows the reduction in computation time when using the reduced order models. A 58.16% reduction is obtained with the 30-order model and 46.87% with the 60-order model. The reduction in the computation time also means a reduction in the computation power required for the simulation.

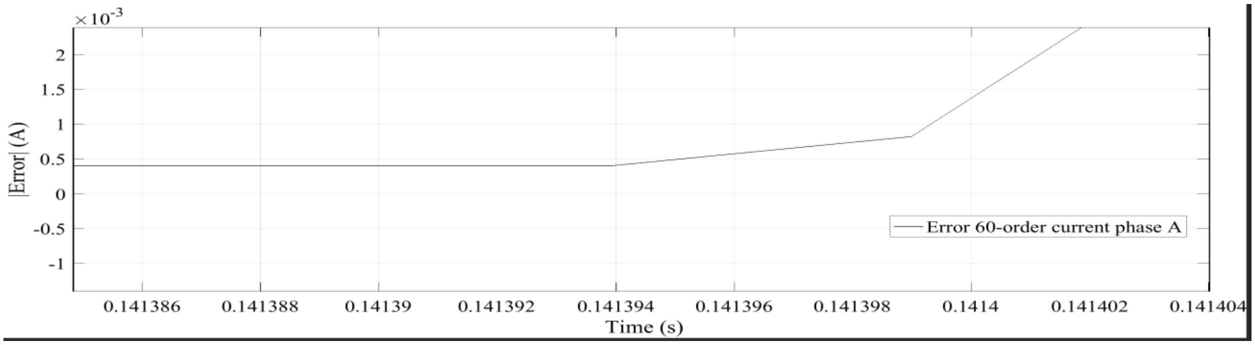


Fig. 3-34. Zoom-in of inflection point in Fig. 3-33.

Table 3-5. Reduction in computation time with PV source connected

Order	**SimPower**	**30-order**	**60-order**
Computation time (s)	992	415	527
Reduction (%)	-	58.16	46.87

3.9 Conclusion

The Modified State Space network formulation presented in [3], [7] was applied to the IEEE-123 Nodes Test Feeder using a MATLAB™ code editor. The code corresponding to the Modified State Space formulation can be found in Appendix B.1 and B.2. A Krylov subspace-based model order reduction algorithm was presented in Appendix B.3. The original system was reduced to 30- 24- and 12-order. It was found that once the system is reduced to 12-order, it losses some of the original high frequency behavior. The high frequency behavior was improved by shifting the interpolation point from s=0 to s=6000 rad/s; however, the low frequency behavior is affected.

A 11.23% reduction in the computation time was obtained by using the 30-order system while keeping a maximum deviation of approximately 3e-6 Amps between the original model and the reduced model.

In addition, a two-sided Arnoldi was implemented to improve the accuracy of the reduced model. Fig. 3-25 shows the errors of the one-sided and two-sided 30-order reduced systems, and this result validate the improvement in the performance of the reduced system using the propose method.

In Fig. 3-28 and 3-29 a set up in MATLAB/SIMULINK™ for simulating a PV source is presented. A transient in the power output of the PV source was simulated using the arrangement in Fig. 3-29. This set up can be extended to other nonlinear sources or devices as STACOMs to study their impact in a distribution level network. However, an increment in the error between this setup and the SimPower System™ model was found when the order of the system was increase from 30 to 60.

CHAPTER 4

POWER SYSTEMS CONTROLLER DESIGN USING BALANCED TRUNCATION MODEL ORDER REDUCTION

This chapter presents the application of a Balanced Truncation-based model order reduction in the context of generation- and transmission-level power systems. The system selected for this application is the 3-Machine 9-Bus presented in [9]. A stable dynamic representation of the system is obtained by computing the matrices that constitute the differential algebraic equations of the system. The linearized small-signal model is obtained considering the power flow results presented in [28]. A reduced state feedback is designed based on the reduced system. The performance of the closed- and open-loop systems are compared.

4.1 3-Machine 9-Bus system parameters

The one-line diagram of the 3-Machine 9-Bus [9] is shown in Fig. 4-1. The system is constituted of three synchronous generators with three step-up transformers, six transmission lines, and three balanced three-phase loads, all arranged in a ring configuration. In this study, the loads are considered constant-power loads.

The following information, required to model the system, is available in Appendix D.1:

- Voltage of each bus obtained from the power flow study results presented in [28].
- Generators' parameters.
- Admittance of the network. All impedance in Fig. 4-1 are converted to admittance and added at each bus.

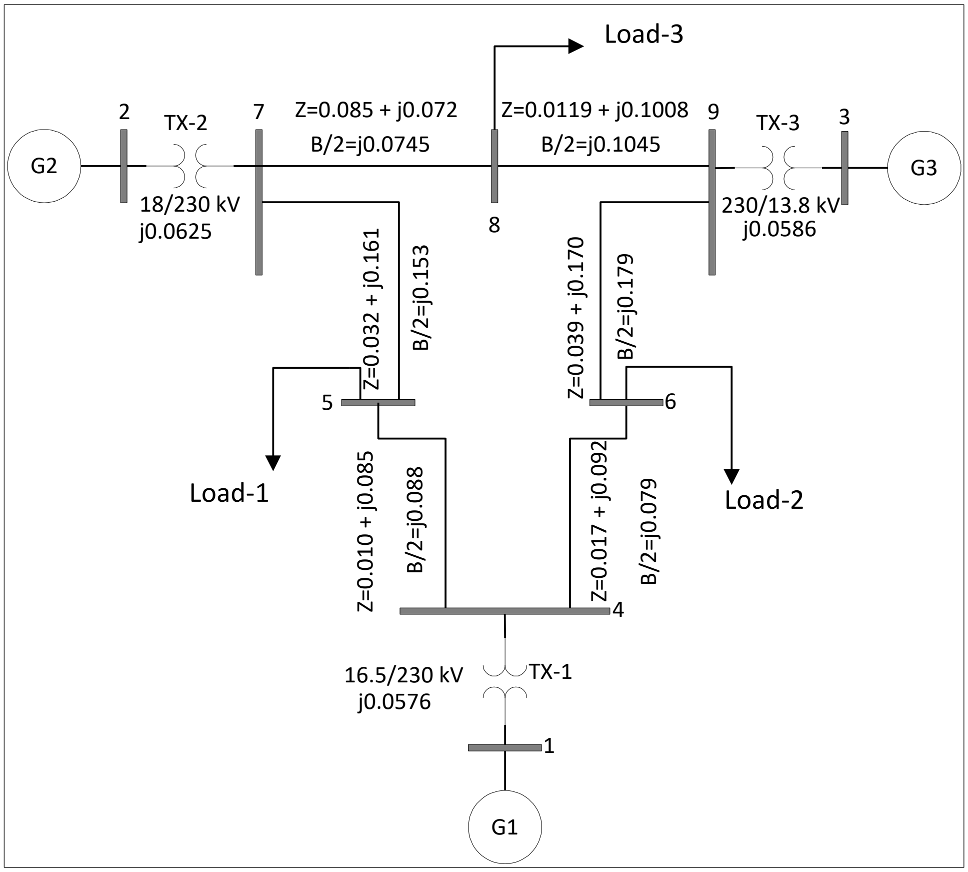


Fig. 4-1. 3-Machine 9-Bus system one line diagram.

A linear time-invariant dynamic representation of this system is required to applied model order reduction. The following sections describe the methodology implemented for this end.

4.2 Dynamic representation of the system

The representation of power system networks at generation- and transmission-level involves a set of differential algebraic equations of the form (4-1) [9], [14], [29], [30]. In this representation, the dynamic of the generator is represented by the set of state-space variable while the set algebraic equations introduce the interaction of the network, borrowing notation from [14]

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{t}) \\ \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{t}) \\ \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{t}) \end{cases} \tag{4-1}$$

in which the vector $\boldsymbol{x}$ comprises the state variables, the vector $\boldsymbol{z}$ comprises dependent voltages and currents that were not adopted as state variables, and $\boldsymbol{u}$ includes the inputs to the system. Since the variables in $\boldsymbol{z}$ can be expressed in terms of $\boldsymbol{x}$ and $\boldsymbol{u}$, the matrix of the system can be obtained as

$$\boldsymbol{A} = \boldsymbol{F_x} - \boldsymbol{F_z G_z^{-1} G_x}, \boldsymbol{B} = \boldsymbol{F_u} - \boldsymbol{F_z G_z^{-1} G_u}, \tag{4-2a}$$

$$\boldsymbol{C} = \boldsymbol{H_x} - \boldsymbol{H_z G_z^{-1} G_x}, \boldsymbol{D} = \boldsymbol{H_u} - \boldsymbol{H_z G_z^{-1} G_u} \tag{4-2b}$$

where the matrices $\boldsymbol{F_{x,z,u}}$, $\boldsymbol{G_{x,z,u}}$, and $\boldsymbol{H_{x,z,u}}$ are the coefficient matrices obtained after linearization.

The formulation of the system in (4-1) and the following linearization is carried out based on the procedure presented in [28], and it is described in the following sections.

4.2.1 Synchronous machine two-axis model

The two-axis model of synchronous generator was adopted for this study. Four state space variables are introduced by this model $\boldsymbol{x_i} = [\delta_i \ \omega_i \ E_{qi} \ E_{di}]$ for i=1, 2 … m in which m is the number of generators in the system. The derivation of this model involves the following assumptions [9], [28], [31]:

- The generator consists of six windings: three-phase windings in the stator, one field winding, and two damping windings over the direct and quadrature axis respectively.
- The change in the stator fluxes are neglected ($\dot{\lambda}_d = 0, \dot{\lambda}_q = 0$).

- The rotor speed during transient state is approximately synchronous speed ($\omega \cong \omega_s$). As a result, the power through the airgap is equal to the electromagnetic torque [28].
- The effect of the damping windings during transient state is neglected.

The resulting differential equations describing the generator are

$$\frac{d\delta}{dt} = \omega - \omega_s, \tag{4-3}$$

$$\frac{d\omega}{dt} = \frac{T_M}{M} - \frac{(E'_q - X'_d I_d)I_q}{M} - \frac{(E'_d + X'_q I_q)I_d}{M}, \tag{4-4}$$

$$\frac{dE'_q}{dt} = -\frac{E'_q}{\tau'_{do}} - \frac{(X_d - X'_d)I_d}{\tau'_{do}} + \frac{E_{fd}}{\tau'_{do}}, \tag{4-5}$$

$$\frac{dE'_d}{dt} = -\frac{E'_d}{\tau'_{qo}} - \frac{(X_q - X'_q)I_q}{\tau'_{qo}} \tag{4-6}$$

where:

δ is the torque angle,
ω and ω_s are the rotor speed and the synchronous speed,
T_M is the mechanic torque input to the rotor shaft,
E'_q and E'_d are the emfs over the q-axis and d-axis respectively,
I_q and I_d are the q-axis and d-axis currents,
X_d and X_q are the d-axis and q-axis reactance, while $X'_d$and X'_q are the transient reactance corresponding to q- and d-axis,
E_{fd} is the emf across the field winding,
τ'_{do} and τ'_{qo} are the time constants for the d- and q-axis,
M is the inertia coefficient.

The above quantities are given in per unit system, except for δ ω τ'_{do} and τ'_{qo}, which are given in rad, rad/s, respectively, and the two time constants are given in seconds.

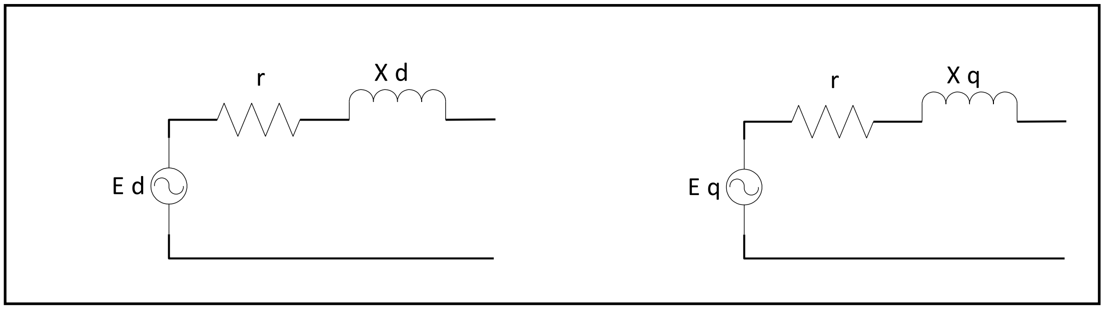


Fig. 4-2. D- and q-axis equivalent two-axis generator model.

The two-axis generator model can be considered as two emf, E'_q and E'_d, behind the transient reactance $X'_q$and X'_d, respectively [31], as is shown in Fig. 4-2.

The differential equations (4-3) – (4-6) are then augmented by the inclusion of auxiliary equipment and other elements that constitute the plant such as: turbine, actuators, automatic voltage regulators, and exciter control loops. For this study, just the latter two are additionally included in the model, and its behavior is briefly described next.

4.2.2 Synchronous generator exciter

The function of the exciter is to provide the dc current to the field winding of the rotor. The exciter controls the terminal voltage magnitude by varying the current injected to this winding.

The input to the exciter is modulated by a voltage regulator consisting on a transducer and amplifying stage. The transducer processes the terminal voltage and transforms it into a dc component to be compared with a reference signal and amplified it [30]. Exciter systems can be classified as: dc systems, ac systems, and static excitation systems. The excitation system selected in for this study is the IEEE-Type I exciter consisting on the IEEE-DC1A excitation system presented in [32]. The block diagram of this exciter is shown in Fig. 4-3 [28]. The input of the system is the reference voltage V_{ref} and an additional signal often designated for the inclusion of a power system stabilizer.

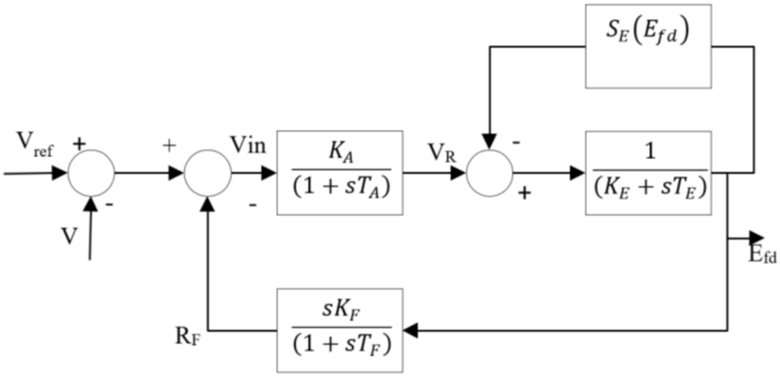


Fig. 4-3. IEEE-Type I exciter block diagram.

The exciter is, as can be observed in Fig. 4-3, a combination of three first-order sub-systems. Thus, the addition of the exciter to the previous model described by (4-3)-(4-6) increases the order of the system from four to seven state variables. This means that each generator will contribute with seven variables to make a total, in the cases of the system under study here, of a 21-order system realization. The differential equation corresponding to the three states of the IEEE-Type I exciter are defined as follows [28]

$$\frac{dE_{fd}}{dt} = -\frac{\left(K_E+S_E(E_{fd})\right)E_{fd}}{T_E} + \frac{V_R}{T_E}, \tag{4-7}$$

$$\frac{dV_R}{dt} = \frac{K_A R_F}{T_A} + \frac{K_A K_F E_{fd}}{T_F T_A} - \frac{V_R}{T_A} + \frac{K_A\left(V_{ref}-V+V_s\right)}{T_A}, \tag{4-8}$$

$$\frac{dR_F}{dt} = -\frac{R_F}{T_F} + \frac{K_F}{{T_F}^2}E_{fd}. \tag{4-9}$$

In the system without the exciter, the emf applied to the field winding E_{fd} was an input to the system; however, with the inclusion of the exciter, E_{fd} is now a state variable and the voltage reference voltage V_{ref}, then, is the new input to the system.

4.2.3 Algebraic equations

In the system of equations (4-3) to (4-9), there are remaining algebraic variables that should be expressed in terms of the state variables. These algebraic variables are stated based on three set of algebraic equations: stator algebraic equations, network equation regarding the generator buses, and the network equations regarding the load buses. The linearized matrix representation of these are expressed as follows [28], [29]

- Stator algebraic equation:

$$\mathbf{0} = \boldsymbol{C_1} \Delta \boldsymbol{X} + \boldsymbol{D_1} \Delta \boldsymbol{I_g} + \boldsymbol{D_2} \Delta \boldsymbol{V_g}. \tag{4-10}$$

- Network equation regarding generator buses:

$$\mathbf{0} = \boldsymbol{C_2} \Delta \boldsymbol{X} + \boldsymbol{D_3} \Delta \boldsymbol{I_g} + \boldsymbol{D_4} \Delta \boldsymbol{V_g} + \boldsymbol{D_5} \Delta \boldsymbol{V_1}. \tag{4-11}$$

- Network equation regarding load buses:

$$\mathbf{0} = \boldsymbol{D_6} \Delta \boldsymbol{V_g} + \boldsymbol{D_7} \Delta \boldsymbol{V_1} \tag{4-12}$$

in which $\Delta \boldsymbol{X}$, $\Delta \boldsymbol{I_g}$, $\Delta \boldsymbol{V_g}$, and $\Delta \boldsymbol{V_1}$ are the linearized vectors of state variables, generator bus currents, generator bus voltages, and non-generator bus voltages respectively. For a detail description of the calculation of these matrices refers to [28], [29].

The state-space representation of the system can then be computed from the combination of the above coefficient matrices as follows [28], [29]

$$\boldsymbol{A} = \boldsymbol{A}' - \boldsymbol{B}' \begin{bmatrix} \boldsymbol{K_1} & \boldsymbol{D_5} \\ \boldsymbol{D_6} & \boldsymbol{D_7} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{K_2} \\ \mathbf{0} \end{bmatrix}, \tag{4-13}$$

$$\boldsymbol{E_{ii}} = \begin{bmatrix} 0 & \frac{1}{M_i} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{K_{Ai}}{T_{Ai}} & 0 \end{bmatrix}^T, \tag{4-14}$$

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{E_{11}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E_{22}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E_{33}} \end{bmatrix}, \tag{4-15}$$

$$\Delta\dot{X} = \boldsymbol{A}\Delta X + \boldsymbol{E}\Delta U \tag{4-16}$$

where

$$\boldsymbol{K_1} = \left[\boldsymbol{D_4} - \boldsymbol{D_3}\boldsymbol{D_1}^{-1}\boldsymbol{D_2}\right], \tag{4-17}$$

$$\boldsymbol{K_2} = \left[\boldsymbol{C_2} - \boldsymbol{D_3}\boldsymbol{D_1}^{-1}\boldsymbol{C_1}\right], \tag{4-18}$$

$$\boldsymbol{A'} = \left[\boldsymbol{A_1} - \boldsymbol{B_1}\boldsymbol{D_1}^{-1}\boldsymbol{C_1}\right], \tag{4-19}$$

$$\boldsymbol{B'} = \left[\boldsymbol{B_2} - \boldsymbol{B_1}\boldsymbol{D_1}^{-1}\boldsymbol{D_2} \quad \boldsymbol{0}\right]. \tag{4-20}$$

The process described above results in the linearized small-signal representation (4-16). These steps are implemented by using a MATLAB™ script. The code of this script can be found in Appendix D.1.

4.3 3-Machine 9-Bus small signal model

The small-signal representation of the 3-Machine 9-Bus system developed in the previous section is tested by using MATLAB/SIMULINK™. The setup for the simulation is shown in Fig. 4-4.

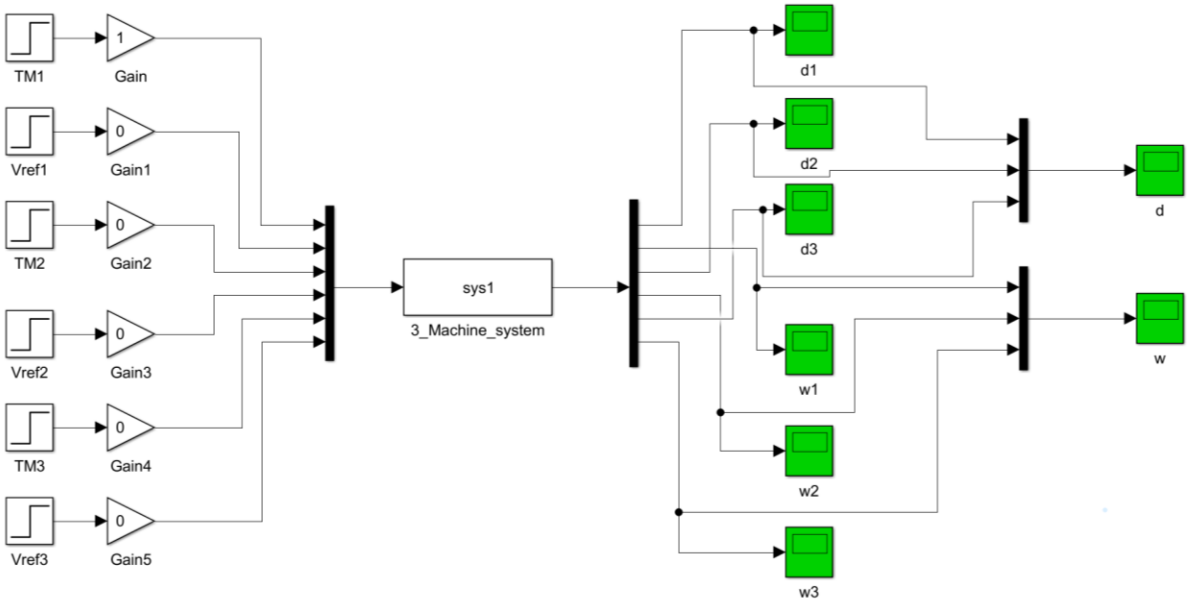


Fig. 4-4. MATLAB/SIMULINK™ simulation set-up for 3-Machine 9-Bus system.

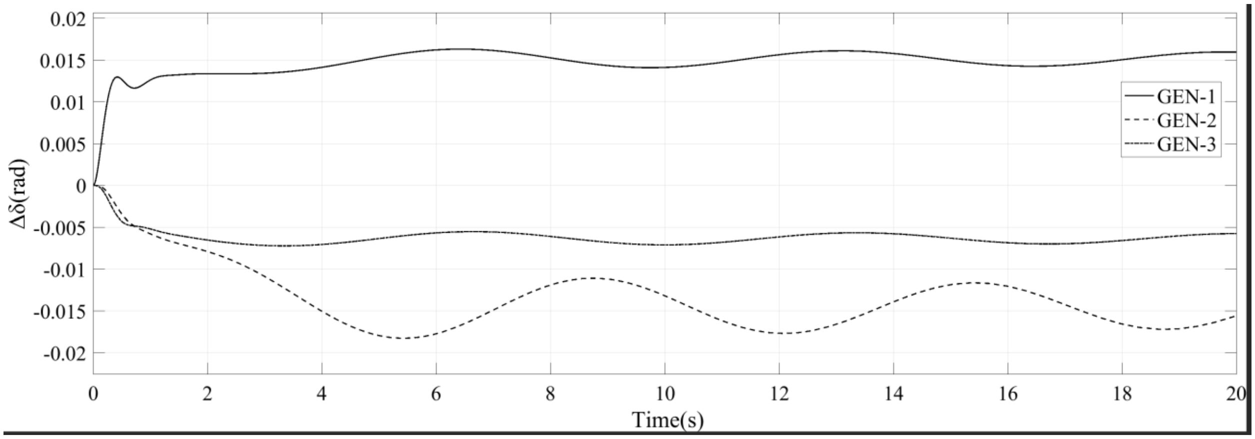


Fig. 4-5. Torque angle (δ) for each generator.

Fig. 4-5 shows the change in the torque angle (δ) of each generator after a 10% step input in the mechanic torque is applied to generator-1 T_{M1}. The torque angle of the generator-1 increases as the positive step in the mechanic torque means more input power. On the contrary, the torque angle of the two other generators decrease also decreasing their output power since there is

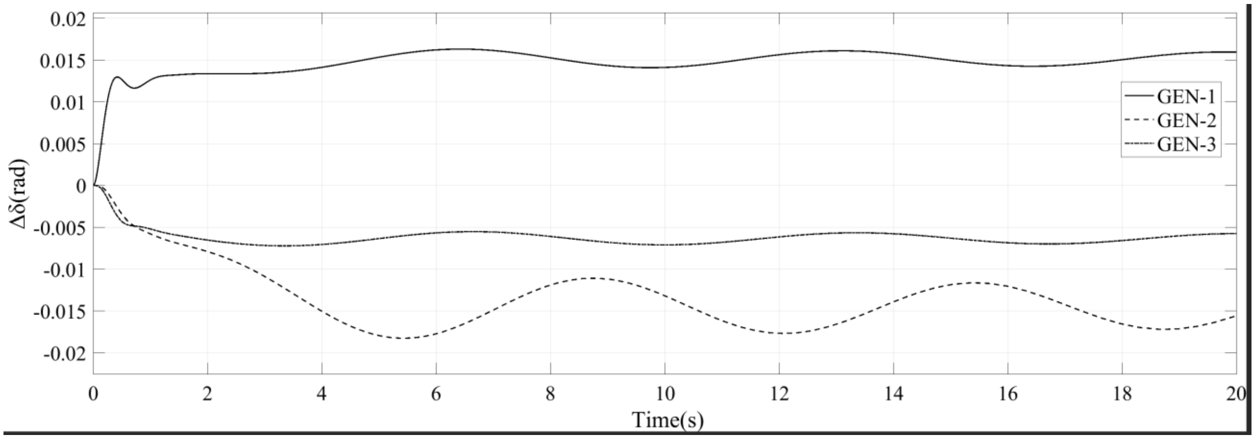


Fig. 4-6. Speed (ω) for each generator.

additional power coming from generator-1. The change in the speed of the generator is shown in Fig. 4-6. Similar behavior is observed for this case. The speed in the generator-1 rises to increase the torque angle, and the speed of generators- 2 and -3 decreases to reduce their torque angle. It is worth to notice that generator-2 presents higher oscillation than the two others. The linear model is, thus, properly simulating the behavior of the 3-Machines 9-Bus system.

4.4 3-Machine 9-Bus model order reduction

The model reduction of the system was performed applying a Balanced Truncation-based method. The Krylov subspace-based reduction presented in the previous chapter was not implemented for this model since the stability is not preserved for all conditions of the system matrices [20]. The stability can be preserved if the original system is passive meeting the following conditions [7]:

Given the original system of the form:

$$\boldsymbol{C\dot{x}} + \boldsymbol{Gx} = \boldsymbol{Bu}\,; \boldsymbol{y} = \boldsymbol{L^T x} \tag{4-21}$$

- $\boldsymbol{C}$ should be non-positive definite

- $\boldsymbol{G} + \boldsymbol{G}^T$ should be non-positive definite
- $\boldsymbol{B} = \boldsymbol{L}$

The reduction of the system was applied by using the MATLAB™ function "balred". This function implements the Balanced Truncation method described in chapter 2. The resulting output signals of the reduced and full-order system corresponding to the speed of the generators is shown in Fig. 4-7 to 4-9.

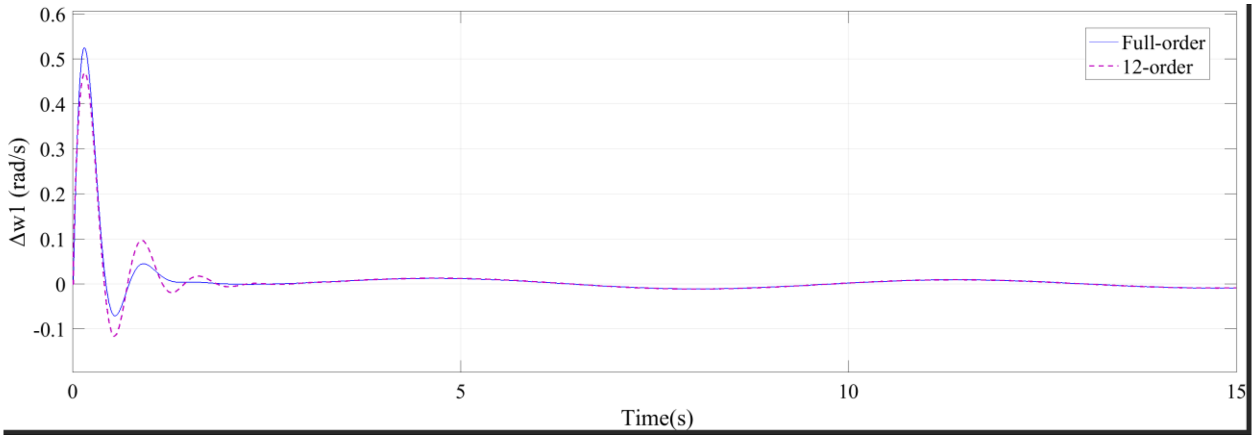


Fig. 4-7. Full- and 12-order Generator-1 speed.

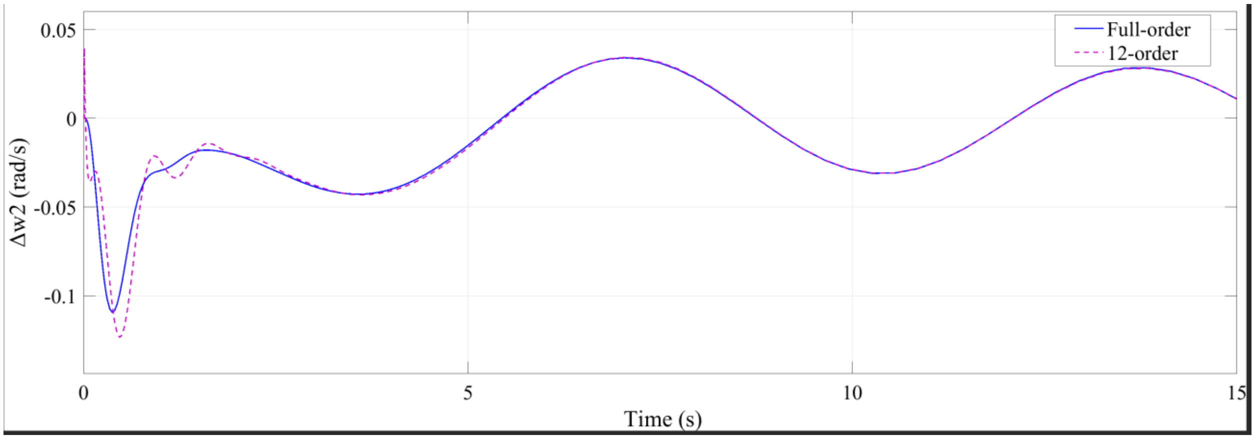


Fig. 4-8. Full- and 12-order Generator-2 speed.

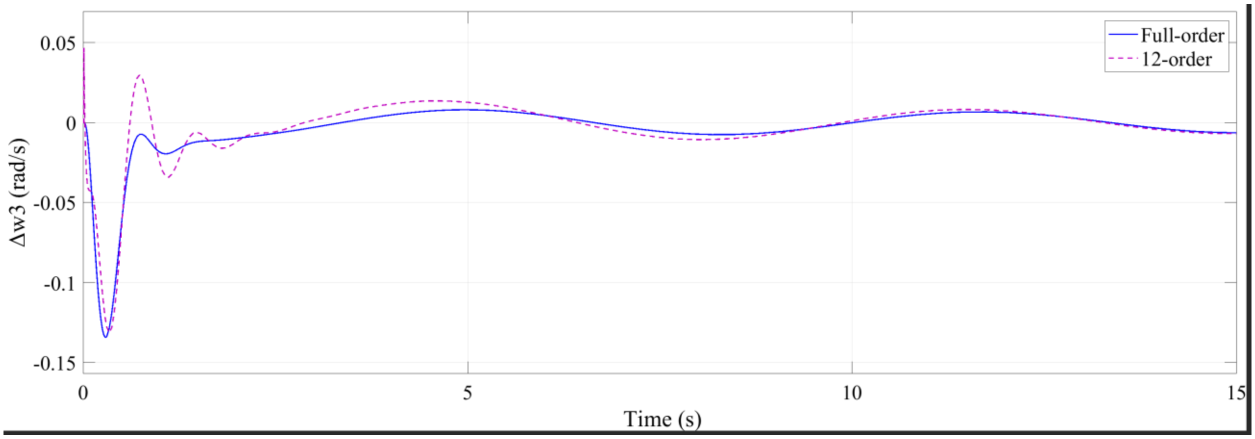


Fig. 4-9. Full- and 12-order Generator-3 speed.

The system reduced to 12-order follows the behavior of the full-order system after the first peak in the signal stars to settles. This performance reflects a loss of information in the reduced system. This immediate loss of information is expected because the original system is 21-order and a considerable amount of information is lost when reducing to 12-order. However, in the following section will be demonstrated that the performance of the 12-order reduced is system is enough to design a state feedback-based controller for the full-order system.

4.5 Linear Quadratic Regulator (LQR) design based on reduced order model

The reduced order model dynamic can be used to design a LQR controller to improve the performance of the system over a small-signal disturbance. The linear quadratic regulator method finds an optimal gain $\boldsymbol{K}$ based on the energy constraints imposed by the weighting matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$, for the state variables and the inputs signals, respectively. This method requires the solution of the cost function

$$J = \int_{t_0}^{t_1} (\boldsymbol{x^T Q x} + \boldsymbol{u^T R u}) \tag{4-21}$$

which also implies the solution of the Riccati equation

$$\boldsymbol{A^T P + PA - PBR^{-1}B^T P + Q = 0}. \tag{4-22}$$

Thus, the design of the controller is summarized in the selection of the elements the diagonal weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. Different approaches are presented in the literature for this end. One approach is to select these elements based on trial and error, and use the step response of the system to check the performance during each trail. Another approach is to calculate the weighting matrices as follows

$$\boldsymbol{Q = C^T C} \qquad \boldsymbol{R = I_{nxn}}. \tag{4-23}$$

The approach presented in [1] for the selection of $\boldsymbol{Q}$ and $\boldsymbol{R}$ is adopted in this study, and a description of the method is presented next.

4.5.1 Construction of the $\boldsymbol{Q}$ matrix

The elements of the diagonal $\boldsymbol{Q}$ matrix [q_1 q_2 ...q_n] are an indication of the "amount of energy" allowed to be spent by the state variables during the control of the system. The higher values of q_n indicates that state variable is more constrained than the others. The main idea of the approach presented in [1] is to constraint the state variables based on how observable and controllable they are. The first step in this method is to rearrange the system in its balanced form such that its controllability and observability Gramians are equal. Then the Hankel values σ_n are computed. Finally, the elements of the $\boldsymbol{Q}$ matrix are calculated as

$$\boldsymbol{Q} = diag(1 \ \frac{\sigma_1}{\sigma_2} \ \frac{\sigma_1}{\sigma_3} \ ... \frac{\sigma_1}{\sigma_m} \ 0 \ 0 \ 0). \tag{4-24}$$

Since the Hankel values are defined such that $\sigma_1 > \sigma_2 > \sigma_3 \ ... > \sigma_n$, the *n*th element in the $\boldsymbol{Q}$ matrix will be higher than the *n*-1th element, the *n*th state variable will be more constraint that the *n*-1th element.

The balanced transformation required for the method described above was already performed when reducing the system as mentioned in the previous section. In fact, the Hankel singular values are calculated during the process of reducing the model when using Balanced Truncation methods, therefore, the convenience of using this approach for calculating the elements of the $\boldsymbol{Q}$ matrix.

In the approach presented in [1], the last elements of the $\boldsymbol{Q}$ were set to zero because they do not influence the behavior of the system as the first elements do. However, in this study all the elements of the matrix were computed considering the fact that the LQR controller is designed based on a reduced system, and the information contained in it was previously truncated.

A similar method is used for the construction of the $\boldsymbol{R}$ matrix. The Hankel singular value of the system are calculated considering one input signal at a time. The system is described in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b_i}\boldsymbol{u}\,, \qquad\qquad \boldsymbol{y} = \boldsymbol{I_{nxn}}\boldsymbol{x}. \tag{4-25}$$

The summation of the Hankel values corresponding to *i*th system is then calculated, and the elements of the of the $\boldsymbol{R}$ matrix are obtained as follows

$$w_i = \left(\textstyle\sum_{j=1}^{n} \sigma_j\right)_i, \tag{4-26}$$

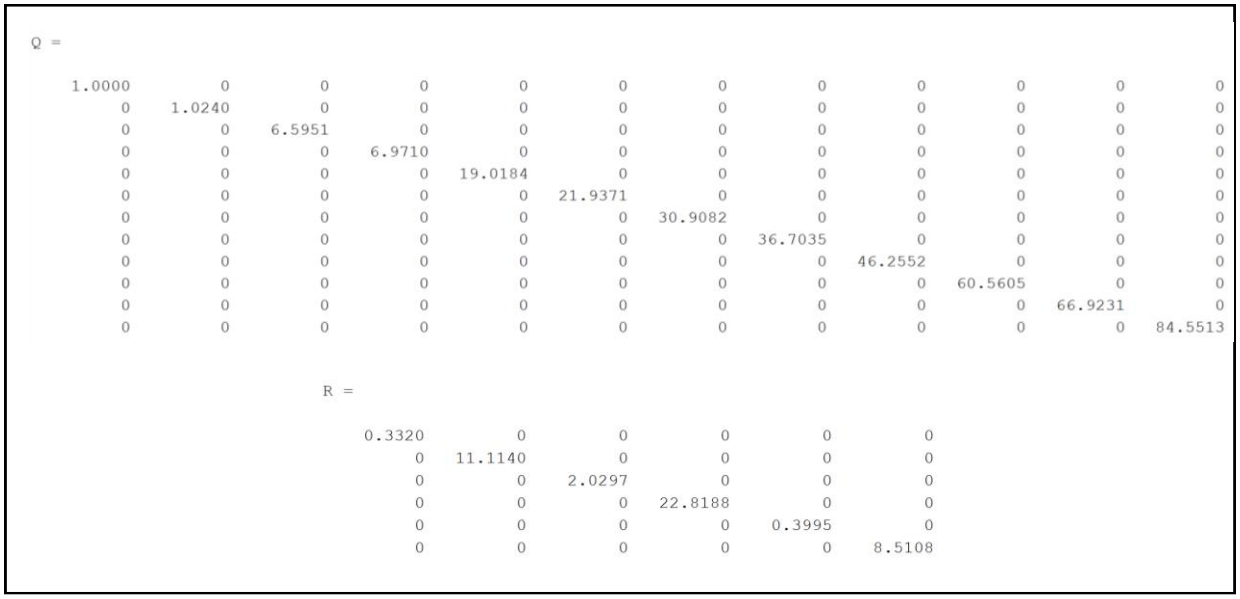
$$\boldsymbol{R} = \gamma\left(diag(1\ \frac{w_1}{w_2}\ \frac{w_1}{w_3} \ldots \frac{w_1}{w_i})\right) \tag{4-27}$$

where γ is calculate as

$$\gamma = (\ \frac{w_1}{\sigma_2}\ ) \tag{4-28}$$

this in order to create a ratio between the constraints in the state variables and the input signals. The above procedure was implemented using a MATLAB™ code editor, the resulting $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices are shown in Fig. 4-10.

The design of the LQR controller is carried out using the MATLAB™ command “lqr” based on the reduced model. Is important to note that a state feedback based on the full-order model would result in a gain of the same order. Hence, the used of the reduced model provide a way to avoid complex, and in some cases, non-realizable controller designs. The implementation of the reduced LQR controller and its results are reported next.



```
Q =

    1.0000         0         0         0         0         0         0         0         0         0         0         0
         0    1.0240         0         0         0         0         0         0         0         0         0         0
         0         0    6.5951         0         0         0         0         0         0         0         0         0
         0         0         0    6.9710         0         0         0         0         0         0         0         0
         0         0         0         0   19.0184         0         0         0         0         0         0         0
         0         0         0         0         0   21.9371         0         0         0         0         0         0
         0         0         0         0         0         0   30.9082         0         0         0         0         0
         0         0         0         0         0         0         0   36.7035         0         0         0         0
         0         0         0         0         0         0         0         0   46.2552         0         0         0
         0         0         0         0         0         0         0         0         0   60.5605         0         0
         0         0         0         0         0         0         0         0         0         0   66.9231         0
         0         0         0         0         0         0         0         0         0         0         0   84.5513


                              R =

                                  0.3320         0         0         0         0         0
                                       0   11.1140         0         0         0         0
                                       0         0    2.0297         0         0         0
                                       0         0         0   22.8188         0         0
                                       0         0         0         0    0.3995         0
                                       0         0         0         0         0    8.5108
```

Fig. 4-10. Resulting $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices.

.

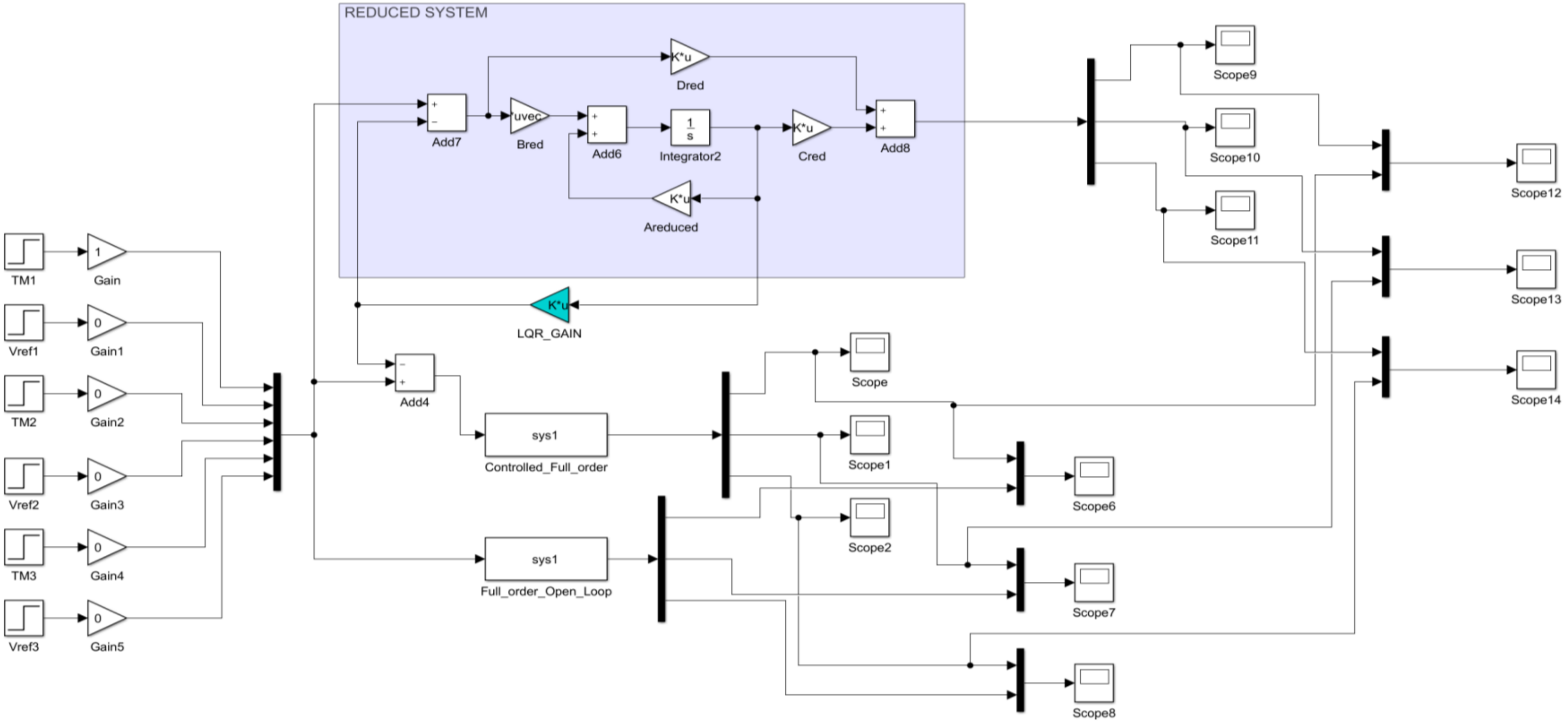


Fig. 4-11. MATLAB/SIMULINK™ set-up for the implementation of the reduced LQR controller

4.6 Performance of the 3-Machine 9-Bus system with reduced order LQR controller

In the previous section, the procedure for the design of a state feedback LQR controller was presented. Since in a state feedback the gain requires the signal of all the states, the use of a state observer may be one option. However, in this study the use of the explicit representation of the reduced system is proposed. This implementation can be accomplished using digital devices

It is important to note that because the controller was designed based on the reduced system, the state required by it are the states of the reduced model, which are no longer physical quantities. The proposed setup for the implementation of the controller is shown in Fig. 4-10. The states generated by the reduced system feed the controller gain, and the control signal is introduced to the full order system.

Fig. 4-12 to 4-14 show the behavior of the speed of generator 1, 2, and 3 to a step of 10% in the mechanical torque of generator 1.

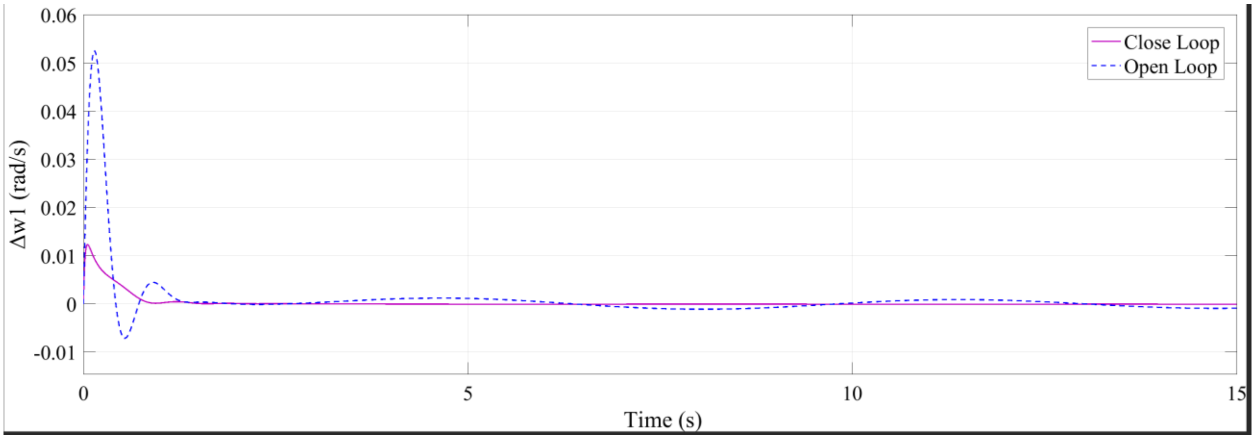


Fig. 4-12. Generator 1 speed for the close and open loop system.

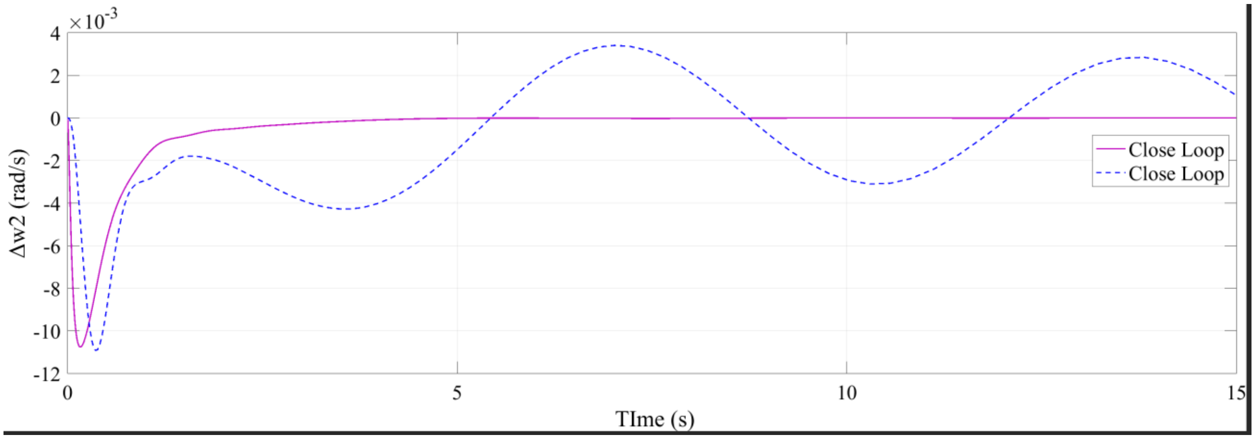


Fig. 4-13. Generator 2 speed for the close and open loop system.

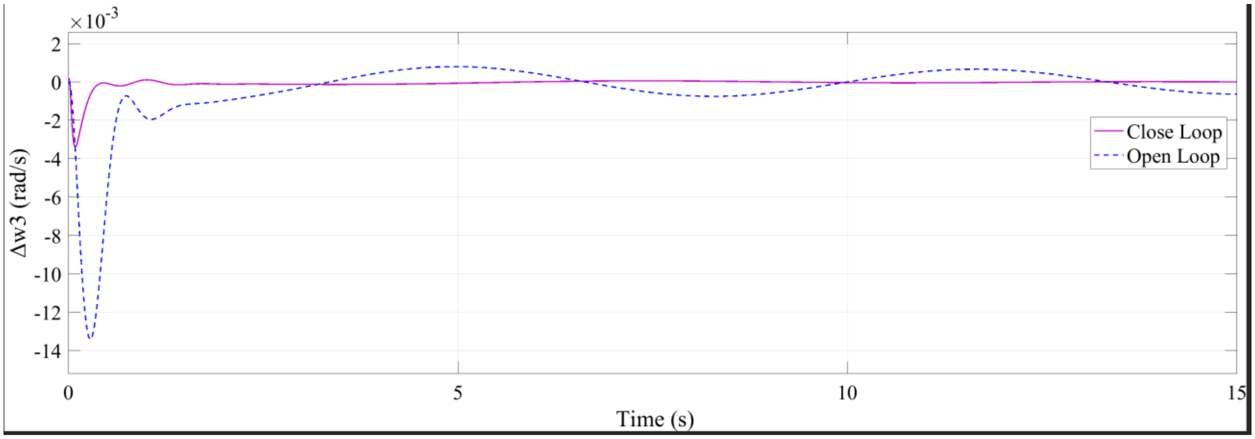


Fig. 4-14. Generator 3 speed for the close and open loop system.

The speed oscillations are damped for the three generators, as observed in Fig. 4-11 to 4-13.

The main disadvantage of this approach is that is a centralized controller. In the classic power system stabilizer (CPSS) approach, the designed stabilizer receives local signals, and the physical implementation of the device is distributed on each machine. Therefore, the CPSS approach is reliable since the malfunctioning of one PSS does not imply the unavailability of the others. In contrast, a centralized controller relies in a robust communication system since requires of several input signals to generate the control signals.

In the literature, an output feedback approach is often reported. For instance, in [33] a dynamic output feedback is designed based on the concept that the input signal can be represented by a reduced vector $\boldsymbol{x_r}$, where $\boldsymbol{x_r}$ is associated just with the state variables with higher participation factor. Similar to the case proposed in this study, the resulting controller consists of a reduced closed-loop dynamic system providing control signals to a bigger full-order system. Another approach to power system controller design is presented in [34]. A power system stabilizer is design based on H_∞ method. However, the authors apply the method to a reduced-order model since the H_∞ method produces a dynamic feedback system of the same order of the plant where is applied. The previous approaches, as the one presented in this study, are based on centralized controllers; however, [6] presents a coordinated procedure to design decentralized power system controllers. The procedure is called coordinated because seeks the optimal parameters of all controllers at the same time. The parameters are obtained by minimizing a cost function based on the desired eigenvalues of the closed-loop system. This optimization requires the calculation of the eigenvalues, and the authors implements a Cross-Gramian-based model reduction to decrease the burden of calculating the eigenvalues of the full-order system.

4.7 Conclusion

A linearized small signal model of the 3-Machine 9-Bus was obtained. The Balanced Truncation model reduction was implemented to generate a reduced model for the 3-Machine 9-Bus. A LQR controller was designed using the reduced order model. The selection of the elements of the $\boldsymbol{Q}$ and $\boldsymbol{R}$ was based on the procedure presented in [1]. The used of this procedure was proved to be convenient when a Balanced Truncation method is selected for reducing the model. The explicit representation of the reduced system is proposed to be implemented to produce the reduced states for the feedback gain instead of using a reduced observer. The proposed controller improves the damping of the system as can be observed in Fig. 4-11 to 4-13.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Two applications of model order reduction in power systems have been discussed in this thesis. Specifically, Krylov subspace-based model order reduction was applied in the context of distribution level power systems simulation, and Balanced Truncation-based model reduction was implemented for designing a power system controller in the context of generation- and transmission-level power systems.

The theory background regarding these two model order reduction methods was presented in chapter 2. Krylov subspace-based model reduction creates the projection bases based on the moments of the original system, so the reduced system emulates these moments. On the other hand, Balanced Truncation-based model reduction is based on re-arranging the system in its balanced form. In this form, the firsts states of the system are the most controllable and observable. Thus, the system can be truncated, so the less controllable and observable states are discarded. Although the two model reduction methods are based on different mathematical concepts, they produce the reduced model by projecting the original system onto bases of smaller size.

In chapter 3, the Krylov subspace-based reduction was implemented for the reduction of the IEEE-123 Node Test Feeder. Before the reduction was applied, a Modified State Space model of the test feeder was developed, adopting the procedure presented in [7]. The derivation of this model was implemented using MATLAB™ code editor, and the corresponding code is presented in the algorithms 1 and 2 in Appendix B.1 and B.2, respectively. The resulting model was demonstrated to match the behavior of a SimPower System™-based model of the same test feeder. The reduction of the resulting state-space model was performed by using an approach adopted from [7], which is based on the one-sided Arnoldi algorithm. In addition, the use of two-sided Arnoldi algorithm was proved to improve the performance of the reduced model. The deviation between the reduced and original model was further decreased with the two-sided Arnoldi algorithm, as can be observed in Fig. 3-25. The transient produced by a PV source connected to the system was simulated using a reduced model and the setup shown in Fig. 3-29. However, the deviation between the results obtained with this setup and the results from a SimPower System™-based model was higher than expected. The deviation with no PV source is approximately 1e-6 Amps during steady-state. The error with PV source approximately oscillates between 0 and 1.5 Amps during steady-state operation. This represents 1.67% of the magnitude of the current. Further trails are required for the improvement of this simulation.

In chapter 4, the Balanced Truncation method was implemented for designing a power system controller for the 3-Machine 9-Bus power system. The controller design was based on the reduced model. A linearized small signal model of the system was developed following the procedure presented in [28], [29]. The procedure was implemented using MATLAB™ code editor, and the code is presented in Appendix D.1. The "lqr" command available in MATLAB™ was used to design the controller. The choice of the matrix $\boldsymbol{Q}$ and $\boldsymbol{R}$ was based on the procedure presented in [1]. The convenience of using this procedure when the reduced model is obtained through Balanced Truncation methods was demonstrated. The resulting controller was implemented using the explicit representation of the reduced model in order to retrieve the reduced state variables, as is shown in Fig. 4-10. The designed controller properly damps the oscillation present in the open-loop system, as can be observed in Fig. 4-11 to 4-13.

5.2 Future work

The following are suggestions for future work and improvement opportunities regarding the application of Krylov subspace-based model order reduction in the context of distribution level power systems presented in chapter 3:

- Scale up the model of the distribution system: the algorithms for the development of a modified state space model and the reduction of the model can be extended to bigger distribution test feeders.
- Inclusion of D-FACTS: in this study, the reduced model was used to simulate the transient of a PV source connected to the test feeder. The reduced model can also be used for the inclusion of distribution level FACTS equipment.
- Real-time simulation: the reduced model can be applied for real-time simulation of transients in distribution systems. Since the reduced introduces a deviation with respect to full model, the study of the decrement of the order and its effect over the accuracy of the simulation is of interest. If the order of the system is smaller, a smaller time step can be used.
- Improvement of the current deviation between the reduced model and the SimPower System™ model: in chapter 3, the set up in Fig. 3-29 was proposed for simulating the transient in a PV source using the reduced model. However, a deviation was found between the resulting current using the setup in Fig. 3-29 and using a SimPower System™-based model. After several tests, no evident root cause was found. Thus, further work is required to address this issue.

The following are suggestions for future work and improvement opportunities regarding the application of Balanced Truncation model order reduction in the context of power system controller design discussed in chapter 4:

- Comparison with other schemes: the application of optimization algorithms for tuning the parameter of the controller is gaining attention in the literature. Comparing the proposed controller with controllers designed using these algorithms may be pertinent.
- Implementation of the controller in real-time simulation: because the proposed controller scheme is centralized, it requires a stable communication infrastructure to carry the signals produced by Phasor Measurement Units. A real-time simulation setup can be implemented to study the vulnerability of the control to communication outage and bad data issues.

REFERENCE

[1] M. Aldeen, “M. Aldeen and F. Crusca, "Multimachine power system stabiliser design based on new LQR approach," in *IEE Proceedings - Generation, Transmission and Distribution*, vol. 142, no. 5, pp. 494-502, Sep 1995.

[2] IEEE PES Distribution Systems Analysis Subcommittee Radial Test Feeders. [Online]. Available: http://ewh.ieee.org/soc/pes/dsacom/testfeeders.html.

[3] O. Wasynczuk and S. D. Sudhoff, "Automated state model generation algorithm for power circuits and systems," in *IEEE Transactions on Power Systems*, vol. 11, no. 4, pp. 1951-1956, Nov 1996.

[4] Z. H. Rather, Z. Chen and P. Thøgersen, "Challenge of primary voltage control in large scale wind integrated Power System: A Danish power system case study," *IEEE PES ISGT Europe 2013*, Lyngby, 2013, pp. 1-5.

[5] R. H. Bhesdadiya and R. M. Patel, "Available transfer capability calculation using deterministic methods: A case study of Indian power system," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, 2016, pp. 2261-2264.

[6] C. Li, Z. Du, Y. Ni, and G. Zhang, “Reduced Model-Based Coordinated Design of Decentralized Power System Controllers,” *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2172–2181, May 2016.

[7] C. Ding, P. Li, J. Wu, C. Wang, and H. Yu, “Model order reduction for transient simulation of active distribution networks,” *IET Gener. Transm. Distrib.*, vol. 9, no. 5, pp. 457–467, Apr. 2015.

[8] M. D. O. Faruque *et al.*, “Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis,” *IEEE Power Energy Technol. Syst. J.*, vol. 2, no. 2, pp. 63–73, Jun. 2015.

[9] P. M. Anderson and A. A. Fouad, "The Synchronous Machine," in *Power System Control and Stability*. Wiley-IEEE Press, 2003, pp.368-397.

[10] B. Nouri, M. S. Nakhla, and X. Deng, “Stable Model-Order Reduction of Active Circuits,” *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. PP, no. 99, pp. 1–10, 2017.

[11] M. Simeoni, G. A. E. Vandenbosch, and I. E. Lager, “Model-order reduction techniques for linear electromagnetic problems - an overview,” in *2005 European Microwave Conference*, 2005, vol. 2, p. 4 pp.1-4.

[12] V. N. Tulsky, A. S. Vanin, M. A. Tolba, A. Y. Sharova and A. A. Z. Diab, "Study and analysis of power quality for an electric power distribution system — Case study: Moscow region," *2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW)*, St. Petersburg, 2016, pp. 710-716.

[13] L. Fortuna, *et al*, "Model Reduction," in *Model Order Reduction Techniques with Applications in Electrical Engineering*, London, UK: Springer London, 1992, pp. 11-62.

[14] 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, Nov. 2016.

[15] A. Castagnotto, H. K. F. Panzer and B. Lohmann, "Fast H2-optimal model order reduction exploiting the local nature of Krylov-subspace methods," *2016 European Control Conference (ECC)*, Aalborg, 2016, pp. 1958-1969.

[16] Y. Saad, *Iterative methods for sparse linear systems*. Boston: PWS Pub. Co, 1996.

[17] X. Sheldon, *et al.*, *Advanced model order reduction techniques in VLSI design*. Cambridge: Cambridge University Press, 2007.

[18] B. Lohmann and B. Salimbahrami, "Introduction to Krylov subspace methods in model order reduction," *Methods Appl. Autom.*, pp. 1–13, 2000.

[19] R. W. Freund, "Krylov-subspace methods for reduced-order modeling in circuit simulation," *J. Comput. Appl. Math.*, vol. 123, no. 1–2, pp. 395–421, Nov. 2000.

[20] D. Chaniotis and M. A. Pai, "Model reduction in power systems using Krylov subspace methods," in *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 888-894, May 2005.

[21] S. Ghosh and N. Senroy, "A comparative study of two model order reduction approaches for application in power systems," *2012 IEEE Power and Energy Society General Meeting*, San Diego, CA, 2012, pp. 1-8.

[22] B. N. Datta, *Numerical methods for linear control systems: design and analysis*, vol. 1. Academic Press, 2004.

[23] W. Ford, *Numerical linear algebra with applications: using MATLAB*, 1th ed. London ; San Diego, CA: Academic Press, 2015.

[24] W. H. Kersting, "Radial distribution test feeders," in *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 975-985, Aug 1991.

[25] T. Gönen, *Electric power distribution system engineering*, vol. 2^{nd} ed. Boca Raton: CRC Press, 2008.

[26] Matlabworks. “Reduced row echelon form of matrix (Gauss-Jordan elimination) - MATLAB rref.” [Online]. Available: https://www.mathworks.com/help/symbolic/rref.html?requestedDomain=www.mathworks.com.

[27] I. M. Elfadel and D. D. Ling, "A block rational Arnoldi algorithm for multipoint passive model-order reduction of multiport RLC networks," *1997 Proceedings of IEEE International Conference on Computer Aided Design (ICCAD)*, San Jose, CA, USA, 1997, pp. 66-71.

[28] D. Mondal, et al. *Power System Small Signal Stability Analysis and Control.* Academic Press, US, 2014.

[29] P. W. Sauer and M. A. Pai, *Power System Dynamics and Stability*. New Jersey: Prentice-Hall, 1998.

[30] P. Kundur, *et al*, *Power system stability and control*. New York: McGraw-Hill, 1994.

[31] J. Machowski, *et al*, *Power System Dynamics and Stability*. New York: Wiley, 1997.

[32] “IEEE Recommended Practice for Excitation System Models for Power System Stability Studies,” *IEEE Std 4215-2016 Revis. IEEE Std 4215-2005*, pp. 1–207, Aug. 2016.

[33] M. Aldeen and L. Lin, “A new reduced order multimachine power system stabilizer design,” *Electr. Power Syst. Res.*, vol. 52, no. 2, pp. 97–114, Nov. 1999.

[34] Hardiansyah, S. Furuya, and J. Irisawa, “A robust H∞ power system stabilizer design using reduced-order models,” *Int. J. Electr. Power Energy Syst.*, vol. 28, no. 1, pp. 21–28, Jan. 2006.

APPENDIX A.1

LINE SEGMENTS' RESISTANCE AND INDUCTANCE MATRICES

The values of resistance are in ohms/ft and inductance in H/ft

Table A-1. Resistance and inductance matrices corresponding to configuration #1.

Configuration 1			
R1	0.2841		
		0.2897	
			0.2865
L1	0.0018		
		0.0017	
			0.0018

Table A-2. Resistance and inductance matrices corresponding to configuration #2.

Configuration 2			
R2	0.2897		
		0.2865	
			0.2841
L2	0.0017		
		0.0018	
			0.0018

Table A-3. Resistance and inductance matrices corresponding to configuration #3.

Configuration 3			
R3	0.2865		
		0.2841	
			0.2897
L3	0.0018		
		0.0018	
			0.0017

Table A-4. Resistance and inductance matrices corresponding to configuration #4.

Configuration 4			
R4	0.2865		
		0.2897	
			0.2841
L4	0.0018		
		0.0017	
			0.0018

Table A-5. Resistance and inductance matrices corresponding to configuration #5.

Configuration 5			
R5	0.2897		
		0.2841	
			0.2865
L5	0.0017		
		0.0018	
			0.0018

Table A-6. Resistance and inductance matrices corresponding to configuration #6.

Configuration 6			
R6	0.2841		
		0.2865	
			0.2897
L6	0.0018		
		0.0018	
			0.0017

Table A-7. Resistance and inductance matrices corresponding to configuration #7.

Configuration 7			
R7	0.2841		
			0.2865
L7	0.0018		
			0.0018

Table A-8. Resistance and inductance matrices corresponding to configuration #8.

Configuration 8			
R8	0.2841		
		0.2865	
L8	0.0018		
		0.0018	

Table A-9. Resistance and inductance matrices corresponding to configuration #9, #10, and #11.

Configuration 9			
R9	0.8257		
L9	0.0022		

APPENDIX B.1

ALGORITHM 1: NODE INCIDENCE MATRIX FORMULATION

```
            File name:

      %NODE INCIDENCE MATRIX LINE SEGMENTS

      Conf7=[1 0 0;0 0 0;0 0 1];  %Configuration 7 Phase_ACN
      Conf8=[1 0 0;0 1 0;0 0 0];  %Configuration 8 Phase_ABN
      Conf9=[1 0 0;0 0 0;0 0 0];  %Configuration 9 Phase_AN
      Conf10=[0 0 0;0 1 0;0 0 0]; %Configuration 10 Phase_BN
      Conf11=[0 0 0;0 0 0;0 0 1]; %Configuration 11 Phase_CN
      Nname=unique([SegLine(:,1);SegLine(:,2)]);  %Extracting the names of the
nodes
      n=length(SegLine);  %Rows of SegLine=Number of line segments
      n2=length(Nname);   %Number of nodes in the system
      a=1:3:n2*3;         %Position of nodes spaced by 3
      b=1;                %To indicate the number of column
      A1=zeros(n2*3);     %Initiating matrix to contain Node incidence matrix
      Rline=zeros(n*3);   %Initiating matrix to contain resistance values
      Lline=zeros(n*3);   %Initiating matrix to contain inductance values
      Npos=[Nname a'];     %Matrix to relate node and its position in the Node
incidence matrix

      %Loop to read the values in Line Segment matrix
      for r=1:n
          x=SegLine(r,1); %Node connected to terminal #1
          y=SegLine(r,2); %Node connected to terminal #2
          z=SegLine(r,4); %Configuration number of the line segment
          l=SegLine(r,3); %length in ft
          px=Npos((Npos(:,1)==x),2);  %position of Node x
          p2=Npos((Npos(:,1)==y),2);  %position of Node y
          %Choosing the matrix depending on the line segment configuration
          if z<=6||z==12  %Three-phase configuration
              c=eye(3);   %Identity matrix to add positive "1"
              eval(['Rline1=l*R',int2str(z),';']); %Resistance matrix of line
segment
              eval(['Lline1=l*L',int2str(z),';']); %Inductance matrix of line
segment
          else            %Non-three-phase configuration
              eval(['c=Conf',int2str(z),';']);     %Assign configuration number
Conf_z
              eval(['Rline1=l*R',int2str(z),';']); %Resistance matrix of line
segment
              eval(['Lline1=l*L',int2str(z),';']); %Inductance matrix of line
segment
          end
          A1(px:px+2,b:b+2)=c;     %Writing positive "1" in the node incidence
matrix
          A1(p2:p2+2,b:b+2)=-c;   %Writing negative "-1" in the node incidence
matrix
          Rline(b:b+2,b:b+2)=Rline1; %Writing in the resistance matrix
          Lline(b:b+2,b:b+2)=Lline1; %Writing in the inductance matrix
          b=b+3;  %augmenting to move to the next columns
      end
      %%
      %NODE INCIDENCE MATRIX FOR LOAD

      n3=length(Load);        %Number of loas in the system
      A2=zeros(n2*3+1,n3*3);   %Initiating matrix to contatin node incidence
matrix
      Rload=zeros(n3*3);      %Load resistance matrix
      Lload=zeros(n3*3);      %Load inductance matrix
      b2=1;                   %Indicates the number of column
      %Loop for reading the values in the matrix Load
      for r=1:n3
          x2=Load(r,1);   %Name of the node where is connected
          px2=Npos((Npos(:,1)==x2),2);    %Position of the Node x2
          if Load(r,2)==0 && Load(r,4)==0
              %"0" in columns 2 and 4 indicate load connected at phase C
              c2=Conf11;
              cn=[0 0 -1];
              Rload1=Load_Imp(r,6)*Conf11; %Resistance matrix of the load
              Lload1=Load_Imp(r,7)*Conf11; %Inductance matrix of the load
          elseif Load(r,2)==0 && Load(r,6)==0
              %"0" in columns 2 and 6 indicates load conneceted at phase B
              c2=Conf10;
              cn=[0 -1 0];
              Rload1=Load_Imp(r,4)*Conf10;
              Lload1=Load_Imp(r,5)*Conf10;
          elseif Load(r,4)==0 && Load(r,6)==0
              %"0" in columns 4 and 6 indicates load connected at phase A
              c2=Conf9;
              cn=[-1 0 0];
              Rload1=Load_Imp(r,2)*Conf9;
              Lload1=Load_Imp(r,3)*Conf9;
          else
              %The remaining cases are three-phase loads
              c2=eye(3);
              cn=[-1 -1 -1];
              Rload1=[Load_Imp(r,2) 0 0;0 Load_Imp(r,4) 0; 0 0 Load_Imp(r,6)];
              Lload1=[Load_Imp(r,3) 0 0;0 Load_Imp(r,5) 0; 0 0 Load_Imp(r,7)];
          end
          A2(px2:px2+2,b2:b2+2)=c2; %Writing positive "1" in the node incidence
matrix
          A2(n2*3+1,b2:b2+2)=cn;    %Writing negative "-1" in the node incidence
matrix
          Rload(b2:b2+2,b2:b2+2)=Rload1; %Writing in Load resisance matrix
          Lload(b2:b2+2,b2:b2+2)=Lload1; %Writing in Load inductance matrix
          b2=b2+3; %Augmenting to move to the next columns
      end
      %%
      %NODE INCIDENCE MATRIX FOR CAPACITOR

      n4=length(Cload); %Number of capacitors connected in the system
      A3=zeros(n2*3+1,n4*3); %Initiating node incidence matrix for capacitors
      Craw=zeros(n4*3,n4*3); %Initiating capacitance matrix
      b3=1;                  %Indicates the number of columns
      %Loop for reading the values in Cload
      for r=1:n4
```

```
          x3=Cload(r,1); %Name of node where is connected
          px3=Npos((Npos(:,1)==x3),2); %Position of the node x3
          %Identifying the configuration
          if Cload(r,2)==0 && Cload(r,3)==0
              %"0" in columns 2 and 3 indicates capacitor connected to phase C
              c3=Conf11;
              cn=[0 0 -1];
              Cload1=Cload(r,4)*Conf11; %Capacitance matrix of the capacitor
          elseif Cload(r,2)==0 && Cload(r,4)==0
              %"0" in columns 2 and 4 indicates capacitor connected to phase B
              c3=Conf10;
              cn=[0 -1 0];
              Cload1=Cload(r,3)*Conf10;
          elseif Cload(r,3)==0 && Cload(r,4)==0
              %"0" in columns 3 and 4 indicates capacitor connected to phase A
              c3=Conf9;
              cn=[-1 0 0];
              Cload1=Cload(r,2)*Conf9;
          else
              %The remaning are three-phase configurations
              c3=eye(3);
              cn=[-1 -1 -1];
              Cload1=[Cload(r,2) 0 0;0 Cload(r,3) 0; 0 0 Cload(r,4)];
          end
          A3(px3:px3+2,b3:b3+2)=c3;  %Writing  positive  "1"  in  node  incidence
matrix
          A3(n2*3+1,b3:b3+2)=cn;     %Writing negative "-1" in node incidence
matrix
          Craw(b3:b3+2,b3:b3+2)=Cload1; %Writing the capacitance matrix
          b3=b3+3; %Augmented to move to the next columns
      end
      Craw  =  Craw(any(Craw,2),:);  %Removing  all-zero  rows  from  capacitance
matrix
      Craw= Craw(:,any(Craw,1));  %Removing all-zero columns from capacitance
matrix
      %%
      %NODE INCIDENCE MATRIX FOR VOLTAGE SOURCE

      source_n=149; %Node where the source is connected
      Wn=zeros(n*3+1,3); %Initiation matrix
      px=Npos((Npos(:,1)==source_n),2); %Position of the node source_n
          Wn(px:px+2,1:3)=eye(3); %Writing positive "1" in the node incidence
matrix
          Wn(n2*3+1,1:3)=[-1  -1  -1];  %Writing  negative"-1"  in  the  node
incidence matrix

      %%
      % ELIMINATION OF ALL-ZERO ROWS AND COLUMNS

      %Merging all node incidence matrix
      Araw=[[A1;zeros(1,size(A1,2))] A2 A3 Wn];

      %Merging loads and line segments resistance and inductance matrices
      Rraw=[Rline                          zeros(length(Rline),length(Rload));
zeros(length(Rload),length(Rline)) Rload];
      Lraw=[Lline                          zeros(length(Lline),length(Lload));
zeros(length(Lload),length(Lline)) Lload];
```

```
      %Eliminating all-zero rows and columns from matrices
      Araw2 = Araw(any(Araw,2),:);
      A= Araw2(:,any(Araw2,1)); %FINAL NODE INCIDENCE MATRIX
      Rraw2 = Rraw(any(Rraw,2),:);
      Rbr= Rraw2(:,any(Rraw2,1));
      Lraw2 = Lraw(any(Lraw,2),:);
      Lbr= Lraw2(:,any(Lraw2,1));

      %Equating the dimension of the matrices
      Rbr=[Rbr                   [zeros(size(Rbr,1),(size(A,2)-size(Rbr,2)))];
[zeros((size(A,2)-size(Rbr,1)),size(A,2))]];
      Lbr=[Lbr                   [zeros(size(Lbr,1),(size(A,2)-size(Lbr,2)))];
[zeros((size(A,2)-size(Lbr,1)),size(A,2))]];
      Cbr=zeros(size(A,2));
      Cbr(size(Rraw2,1)+1:size(Rraw2,1)+size(Craw,1),size(Rraw2,1)+1:size(Rra
w2,1)+size(Craw,1))=Craw;

      %Creating matrix for capacitor connection
      M=zeros(size(A,2),size(Craw,1));
      M(size(M,1)-size(Craw,1)-nx*3+1:size(M,1)-
nx*3,1:size(Craw,1))=eye(size(Craw,1));

      %Creating Matrix for input connection
      W=zeros(size(A,2),3*nx);
      W(size(W,1)-nx*3+1:size(W,1),1:nx*3)=eye(nx*3);
      W=W';
```

APPENDIX B.2

ALGORITHM 2: MODIFIED STATE SPACE MODEL FORMULATION

```
      %MODIFIED STATE SPACE MODEL CONSTRUCTION

      Ahat=rref(A);           %Node incidence matrix in echelon form
      [row,col]=size(Ahat);   %Extracting dimension of the matrix
      BbT=[-Ahat(1:row-1,row:col);eye(col-row+1)];  %creatin  Bt  from  non-
identity part
      Bb=[BbT]';
      a=size(Bb,2);

      %Creating matrix C
      C22=[Bb*Lbr*BbT];
      C11=[M'*Cbr*M];
      C=[[C11]                               [zeros(size(C11,1),size(C22,2))];
[zeros(size(C22,1),size(C11,2))], C22];

      %Creating matrix G
      G12=[-M'*BbT];
      G21=[Bb*M];
      G22=[Bb*Rbr*BbT];
      G=[[zeros(size(Craw,1))] G12;G21 G22];

      %Creating input and output matrix
      B12=[-Bb*W'];
      B=[zeros(size(Craw,1),3);B12];
      L_12=[-Bb*W'];
      L=[zeros(size(Craw,1),3);L_12];

      %Modified state space model
      sys_d=dss(-G,B,L',[0],C);
```

APPENDIX B.3

ALGORITHM 3: ARNOLDI ALGORIHTM

```
R=G^-1*B;
V0=orth(R);
q=30;
[row,col]=size(B);
n=q/col;
V=zeros(row,q);
V(:,1:col)=V0;
b=col+1;
for k=1:n
    eval(['X=C*V',int2str(k-1),';']);
    eval(['V0',int2str(k),'=G^-1*X;']);
    for j=1:k
      eval(['H=transpose(V',int2str(k-j),')*V',int2str(j-
1),int2str(k),';']);
      eval(['V',int2str(j),int2str(k),'=V',int2str(j-1),int2str(k),'-
V',int2str(k-j),'*H;']);
    end
    eval(['V',int2str(k),'=orth(V',int2str(k),int2str(k),');']);

    eval(['V(:,b:b+col-1)=V',int2str(k),';']);
    b=b+col;
end

V=V(:,1:q);
Cr30=V'*C*V;
Gr30=V'*G*V;
Br30=V'*B;
Lr30=V'*L;
sys_dred2=dss(-Gr30,Br30,Lr30',[0],Cr30);
```

APPENDIX C.1

REDUCED MODEL CURRENT WAVEFORMS

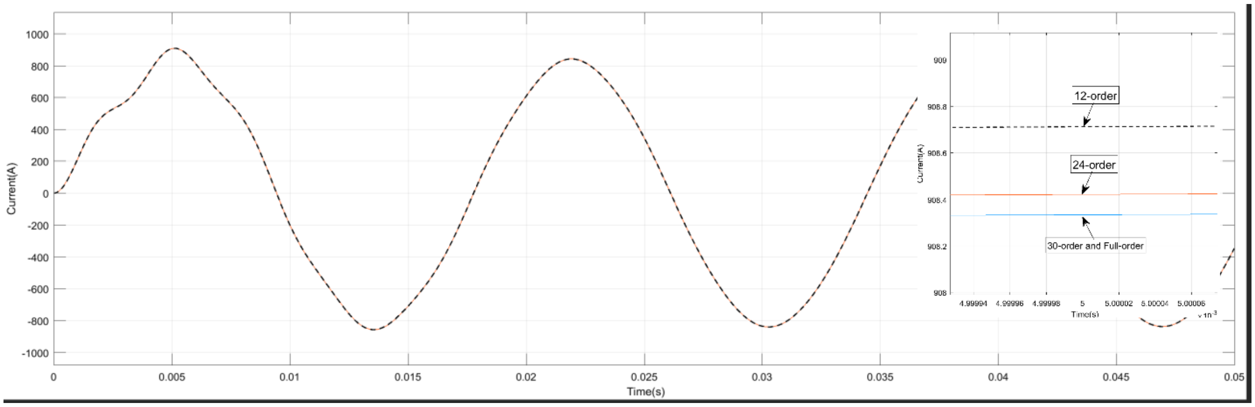


Fig C.1-1. Waveform of output Current of Phase A of full and reduced systems.

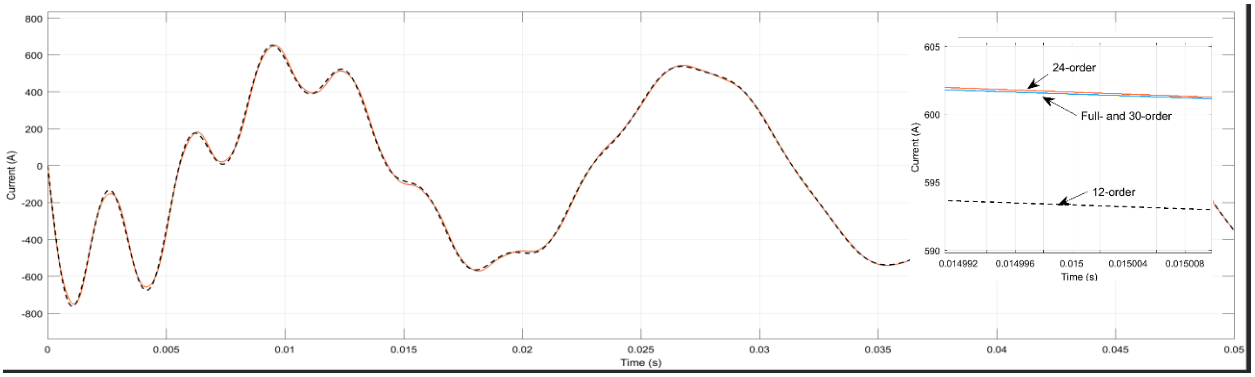


Fig. C.1-2. Waveform of output Current of Phase B of full and reduced systems.

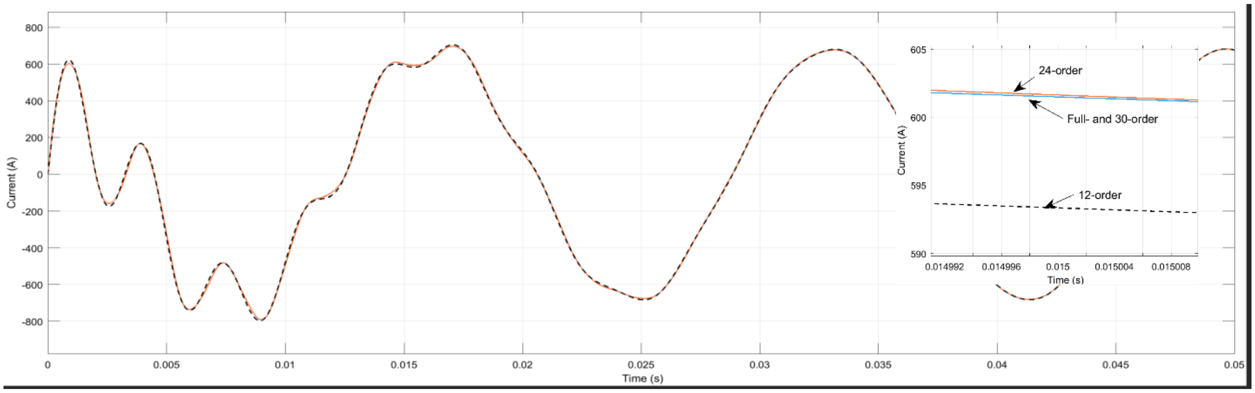


Fig. C.1-3. Waveform of output Current of Phase C of full and reduced systems.

APPENDIX D.1

3-MACHINE 9-BUS LINEARIZED SYSTEM

```
%System Parameters
%Generators Parameters
f=60;
%Generator#1
Rs1=0.0;
H1=23.64;
M1=2*H1/(2*pi*f);
D1=1;
xd1=0.269;
xdp1=0.0608;
xq1=0.0969;
xqp1=0.0969;
Tdo1=8.96;
Tqo1=0.31;

%Generator#2
Rs2=0.0;
H2=6.4;
M2=2*H2/(2*pi*f);
D2=1;
xd2=0.8958;
xdp2=0.1198;
xq2=0.8645;
xqp2=0.8645;
Tdo2=6;
Tqo2=0.535;

%Generator#3
Rs3=0.0;
H3=3.01;
M3=2*H3/(2*pi*f);
D3=1;
xd3=1.998;
xdp3=0.1813;
xq3=1.2578;
xqp3=0.2578;
Tdo3=5.89;
Tqo3=0.6;

%Exciter Data
%Type I
Ka=40;
Ta=0.2;
Ke=1;
Te=0.314;
Kf=0.063;
Tf=0.35;
Ae=0.0039;
Be=1.555;


%Load Flow Data

Sbase=100; %MVA

%Generators
```

```
V1=1.04;
P1=71.6;
Q1=27;

V2=1.025*exp((9.48*pi/180)*1i);
P2=163;
Q2=6.7;

V3=1.025*exp((4.77*pi/180)*1i);
P3=85;
Q3=-10.9;

V4=1.10*exp((-2.26*pi/180)*1i);
V4abs=abs(V4);
V4ang=angle(V4);

V5=0.972*exp((-4.06*pi/180)*1i);
V5abs=abs(V5);
V5ang=angle(V5);

V6=0.989*exp((-3.7*pi/180)*1i);
V6abs=abs(V6);
V6ang=angle(V6);

V7=1.011*exp((3.84*pi/180)*1i);
V7abs=abs(V7);
V7ang=angle(V7);

V8=0.997*exp((0.78*pi/180)*1i);
V8abs=abs(V8);
V8ang=angle(V8);

V9=1.018*exp((2.03*pi/180)*1i);
V9abs=abs(V9);
V9ang=angle(V9);

%NETWORK ADMINATANCE
Y11=-8.4459*1i;
Y11abs=abs(Y11);
Y11ang=angle(Y11);

Y22=-5.4855*1i;
Y22abs=abs(Y22);
Y22ang=angle(Y22);

Y33=-4.1684*1i;
Y33abs=abs(Y33);
Y33ang=angle(Y33);

Y44=3.3074-30.3937*1i;
Y44abs=abs(Y44);
Y44ang=angle(Y44);

Y55=3.8138-17.8426*1i;
Y55abs=abs(Y55);
Y55ang=angle(Y55);

Y66=4.1019-16.1335*1i;
Y66abs=abs(Y66);
Y66ang=angle(Y66);

Y77=2.8047-24.9311*1i;
Y77abs=abs(Y77);
```

```
Y77ang=angle(Y77);

Y88=3.7412-23.6424*1i;
Y88abs=abs(Y88);
Y88ang=angle(Y88);

Y99=2.4371-19.2574*1i;
Y99abs=abs(Y99);
Y99ang=angle(Y99);

Y14=8.4459*1i;
Y14abs=abs(Y14);
Y14ang=angle(Y14);

Y41=8.4459*1i;
Y41abs=abs(Y41);
Y41ang=angle(Y41);

Y72=5.4855*1i;
Y72abs=abs(Y72);
Y72ang=angle(Y72);

Y27=5.4855*1i;
Y27abs=abs(Y27);
Y27ang=angle(Y27);

Y93=4.1684*1i;
Y93abs=abs(Y93);
Y93ang=angle(Y93);

Y39=4.1684*1i;
Y39abs=abs(Y39);
Y39ang=angle(Y39);

Y54=-1.3652+11.6041*1i;
Y54abs=abs(Y54);
Y54ang=angle(Y54);

Y45=-1.3652+11.6041*1i;
Y45abs=abs(Y45);
Y45ang=angle(Y45);

Y64=-1.9422+10.5107*1i;
Y64abs=abs(Y64);
Y64ang=angle(Y64);

Y46=-1.9422+10.5107*1i;
Y46abs=abs(Y46);
Y46ang=angle(Y46);

Y75=-1.1876+5.9751*1i;
Y75abs=abs(Y75);
Y75ang=angle(Y75);

Y57=-1.1876+5.9751*1i;
Y57abs=abs(Y57);
Y57ang=angle(Y57);

Y96=-1.2820+5.5882*1i;
Y96abs=abs(Y96);
Y96ang=angle(Y96);

Y69=-1.2820+5.5882*1i;
```

```
Y69abs=abs(Y69);
Y69ang=angle(Y69);

Y87=-1.6171+13.6980*1i;
Y87abs=abs(Y87);
Y87ang=angle(Y87);

Y78=-1.6171+13.6980*1i;
Y78abs=abs(Y78);
Y78ang=angle(Y78);

Y98=-1.1551+9.7843*1i;
Y98abs=abs(Y98);
Y98ang=angle(Y98);

Y89=-1.1551+9.7843*1i;
Y89abs=abs(Y98);
Y89ang=angle(Y98);

%INITIAL STATES CALCULATION

Ig10=((P1-Q1*1i)/Sbase)/conj(V1);
d10=angle(V1+Ig10*(Rs1+xqp1*1i));
Ig10ang=angle(Ig10);
Ig10ab=abs(Ig10);
Id10=Ig10ab*cos(-d10+Ig10ang+pi/2);
Iq10=Ig10ab*sin(-d10+Ig10ang+pi/2);

V1abs=abs(V1);
V1ang=angle(V1);
Vd10=V1abs*cos(V1ang-d10+pi/2);
Vq10=V1abs*sin(V1ang-d10+pi/2);

Ed10=(xq1-xqp1)*Iq10;

Eq10=Vq10+Rs1*Iq10+xdp1*Id10;

Efd10=Eq10+(xd1-xdp1)*Id10;

Se1=Ae*exp(Be*Efd10);
fe1=(Ke+Efd10*Be*Se1+Se1)/Te;

Vr10=(Ke+Se1)*Efd10;

Rf10=(Kf/Tf)*Efd10;

Vref1=V1abs+(Vr10/Ka);

Tm10=Ed10*Id10+Eq10*Iq10+(xqp1-xdp1)*Iq10*Id10;

%

Ig20=((P2-Q2*1i)/Sbase)/conj(V2);
d20=angle(V2+Ig20*(Rs2+xqp2*1i));
Ig20ang=angle(Ig20);
Ig20ab=abs(Ig20);
Id20=Ig20ab*cos(-d20+Ig20ang+pi/2);
Iq20=Ig20ab*sin(-d20+Ig20ang+pi/2);

V2abs=abs(V2);
V2ang=angle(V2);
Vd20=V2abs*cos(V2ang-d20+pi/2);
Vq20=V2abs*sin(V2ang-d20+pi/2);
```

```
Ed20=(xq2-xqp2)*Iq20;

Eq20=Vq20+Rs2*Iq20+xdp2*Id20;

Efd20=Eq20+(xd2-xdp2)*Id20;

Se2=Ae*exp(Be*Efd20);
fe2=(Ke+Efd20*Be*Se2+Se2)/Te;

Vr20=(Ke+Se2)*Efd20;

Rf20=(Kf/Tf)*Efd20;

Vref2=V2abs+(Vr20/Ka);

Tm20=Ed20*Id20+Eq20*Iq20+(xqp2-xdp2)*Iq20*Id20;

%

Ig30=((P3-Q3*1i)/Sbase)/conj(V3);
d30=angle(V3+Ig30*(Rs3+xqp3*1i));
Ig30ang=angle(Ig30);
Ig30ab=abs(Ig30);
Id30=Ig30ab*cos(-d30+Ig30ang+pi/2);
Iq30=Ig30ab*sin(-d30+Ig30ang+pi/2);

V3abs=abs(V3);
V3ang=angle(V3);
Vd30=V3abs*cos(V3ang-d30+pi/2);
Vq30=V3abs*sin(V3ang-d30+pi/2);

Ed30=(xq3-xqp3)*Iq30;

Eq30=Vq30+Rs3*Iq30+xdp3*Id30;

Efd30=Eq30+(xd3-xdp3)*Id30;

Se3=Ae*exp(Be*Efd30);
fe3=(Ke+Efd30*Be*Se3+Se3)/Te;

Vr30=(Ke+Se3)*Efd30;

Rf30=(Kf/Tf)*Efd30;

Vref3=V3abs+(Vr30/Ka);

Tm30=Ed30*Id30+Eq30*Iq30+(xqp3-xdp3)*Iq30*Id30;

%A matrix formation

A11=[0 1 0 0 0 0 0 ;0 -D1/M1 -Iq10/M1 -Id10/M1 0 0 0; 0 0 -1/Tdo1 0 1/Tdo1 0 0; 0 0 0 -
1/Tqo1 0 0 0; 0 0 0 0 fe1 1/Te 0; 0 0 0 0 -(Ka*Kf)/(Ta*Tf) -1/Ta Ka/Ta; 0 0 0 0 Kf/(Tf^2)
0 -1/Tf];
A12=[0 1 0 0 0 0 0 ;0 -D2/M2 -Iq20/M2 -Id20/M2 0 0 0; 0 0 -1/Tdo2 0 1/Tdo2 0 0; 0 0 0 -
1/Tqo2 0 0 0; 0 0 0 0 fe2 1/Te 0; 0 0 0 0 -(Ka*Kf)/(Ta*Tf) -1/Ta Ka/Ta; 0 0 0 0 Kf/(Tf^2)
0 -1/Tf];
A13=[0 1 0 0 0 0 0 ;0 -D3/M3 -Iq30/M3 -Id30/M3 0 0 0; 0 0 -1/Tdo3 0 1/Tdo3 0 0; 0 0 0 -
1/Tqo3 0 0 0; 0 0 0 0 fe3 1/Te 0; 0 0 0 0 -(Ka*Kf)/(Ta*Tf) -1/Ta Ka/Ta; 0 0 0 0 Kf/(Tf^2)
0 -1/Tf];
A1=blkdiag(A11,A12,A13);
```

```
B11=[0  0;(Iq10*(xdp1-xqp1)-Ed10)/M1  Id10*(xdp1-xqp1)-Eq10/M1;  -(xd1-xdp1)/Tdo1  0;  0
(xq1-xqp1)/Tqo1; 0 0; 0 0; 0 0];
B12=[0  0;(Iq20*(xdp2-xqp2)-Ed20)/M1  Id20*(xdp2-xqp2)-Eq20/M2;  -(xd2-xdp2)/Tdo2  0;  0
(xq2-xqp2)/Tqo2; 0 0; 0 0; 0 0];
B13=[0  0;(Iq30*(xdp3-xqp3)-Ed30)/M3  Id30*(xdp3-xqp3)-Eq30/M3;  -(xd3-xdp3)/Tdo3  0;  0
(xq3-xqp3)/Tqo3; 0 0; 0 0; 0 0];
B1=blkdiag(B11,B12,B13);

B21=[zeros(5,2);0 -Ka/Ta; 0 0];
B22=[zeros(5,2);0 -Ka/Ta; 0 0];
B23=[zeros(5,2);0 -Ka/Ta; 0 0];
B2=blkdiag(B21,B22,B23);

E11=[0 0; 1/M1 0; 0 0; 0 0;0 0; 0 Ka/Ta; 0 0];
E12=[0 0; 1/M2 0; 0 0; 0 0;0 0; 0 Ka/Ta; 0 0];
E13=[0 0; 1/M3 0; 0 0; 0 0;0 0; 0 Ka/Ta; 0 0];
E1=blkdiag(E11,E12,E13);

C11=[-V1abs*cos(d10-V1ang) 0 0 1 0 0 0; V1abs*sin(d10-V1ang) 0 1 0 0 0 0];
C12=[-V2abs*cos(d20-V2ang) 0 0 1 0 0 0; V2abs*sin(d20-V2ang) 0 1 0 0 0 0];
C13=[-V3abs*cos(d30-V3ang) 0 0 1 0 0 0; V3abs*sin(d30-V3ang) 0 1 0 0 0 0];
C1=blkdiag(C11,C12,C13);

D11=[-Rs1 xqp1;-xdp1 -Rs1];
D12=[-Rs2 xqp2;-xdp2 -Rs2];
D13=[-Rs3 xqp3;-xdp3 -Rs3];
D1=blkdiag(D11,D12,D13);

D21=[V1abs*cos(d10-V1ang) -sin(d10-V1ang); -V1abs*sin(d10-V1ang) -cos(d10-V1ang)];
D22=[V2abs*cos(d20-V2ang) -sin(d20-V2ang); -V2abs*sin(d20-V2ang) -cos(d20-V2ang)];
D23=[V3abs*cos(d30-V3ang) -sin(d30-V3ang); -V3abs*sin(d30-V3ang) -cos(d30-V3ang)];
D2=blkdiag(D21,D22,D23);

C21=[((Id10*V1abs*cos(d10-V1ang))-(Iq10*V1abs*sin(d10-V1ang)))   0   0   0   0   0   0;((-
Id10*V1abs*sin(d10-V1ang))-(Iq10*V1abs*cos(d10-V1ang))) 0 0 0 0 0 0];
C22=[((Id20*V2abs*cos(d20-V2ang))-(Iq20*V2abs*sin(d20-V2ang)))   0   0   0   0   0   0;((-
Id20*V2abs*sin(d20-V2ang))-(Iq20*V2abs*cos(d20-V2ang))) 0 0 0 0 0 0];
C23=[((Id30*V3abs*cos(d30-V1ang))-(Iq30*V3abs*sin(d30-V3ang)))   0   0   0   0   0   0;((-
Id30*V3abs*sin(d30-V3ang))-(Iq30*V1abs*cos(d30-V3ang))) 0 0 0 0 0 0];
C2=blkdiag(C21,C22,C23);

D31=[V1abs*sin(d10-V1ang)  V1abs*cos(d10-V1ang);  V1abs*cos(d10-V1ang)  -V1abs*sin(d10-
V1ang)];
D32=[V2abs*sin(d20-V2ang)  V2abs*cos(d20-V2ang);  V2abs*cos(d20-V2ang)  -V2abs*sin(d20-
V2ang)];
D33=[V3abs*sin(d30-V3ang)  V3abs*cos(d30-V3ang);  V3abs*cos(d30-V3ang)  -V3abs*sin(d30-
V3ang)];
D3=blkdiag(D31,D32,D33);



%D4 MATRIX


D4111=-Id10*V1abs*cos(d10-V1ang)+Iq10*V1abs*sin(d10-
V1ang)+V1abs*V4abs*Y14abs*sin(V1ang-V4ang-Y14ang);
D4112=Id10*sin(d10-V1ang)+Iq10*cos(d10-V1ang)-V1abs*Y11abs*cos(V1ang-V1ang-Y11ang)-
V4abs*Y14abs*cos(V1ang-V4ang-Y14ang)-V1abs*Y11abs*cos(V1ang-V1ang-Y11ang);
D4121=Id10*V1abs*sin(d10-V1ang)+Iq10*V1abs*cos(d10-V1ang)-
V1abs*V4abs*Y14abs*cos(V1ang-V4ang-Y14ang);
D4122=Id10*cos(d10-V1ang)-Iq10*sin(d10-V1ang)-V1abs*Y11abs*sin(V1ang-V1ang-Y11ang)-
V4abs*Y14abs*sin(V1ang-V4ang-Y14ang)-V1abs*Y11abs*sin(V1ang-V1ang-Y11ang);
```

```
D4213=-Id20*V2abs*cos(d20-V2ang)+Iq20*V2abs*sin(d20-
V2ang)+V2abs*V7abs*Y27abs*sin(V2ang-V7ang-Y27ang);
D4214=Id20*sin(d20-V2ang)+Iq20*cos(d20-V2ang)-V2abs*Y22abs*cos(V2ang-V2ang-Y22ang)-
V7abs*Y27abs*cos(V2ang-V7ang-Y27ang)-V2abs*Y22abs*cos(V2ang-V2ang-Y22ang);
D4223=Id20*V2abs*sin(d20-V2ang)+Iq20*V2abs*cos(d20-V2ang)-
V2abs*V7abs*Y27abs*cos(V2ang-V7ang-Y27ang);
D4224=Id20*cos(d20-V2ang)-Iq20*sin(d20-V2ang)-V2abs*Y22abs*sin(V2ang-V2ang-Y22ang)-
V7abs*Y27abs*sin(V2ang-V7ang-Y27ang)-V2abs*Y22abs*sin(V2ang-V2ang-Y22ang);


D4315=-Id30*V3abs*cos(d30-V3ang)+Iq30*V3abs*sin(d30-
V3ang)+V3abs*V9abs*Y39abs*sin(V3ang-V9ang-Y39ang);
D4316=Id30*sin(d30-V3ang)+Iq30*cos(d30-V3ang)-V3abs*Y33abs*cos(V3ang-V3ang-Y33ang)-
V9abs*Y39abs*cos(V3ang-V9ang-Y39ang)-V3abs*Y33abs*cos(V3ang-V3ang-Y33ang);
D4325=Id30*V3abs*sin(d30-V3ang)+Iq30*V3abs*cos(d30-V3ang)-
V3abs*V9abs*Y39abs*cos(V3ang-V9ang-Y39ang);
D4326=Id30*cos(d30-V3ang)-Iq30*sin(d30-V3ang)-V3abs*Y33abs*sin(V3ang-V3ang-Y33ang)-
V9abs*Y39abs*sin(V3ang-V9ang-Y39ang)-V3abs*Y33abs*sin(V3ang-V3ang-Y33ang);

D41=[D4111 D4112;D4121 D4122];
D42=[D4213 D4214;D4223 D4224];
D43=[D4315 D4316;D4325 D4326];
D4=blkdiag(D41,D42,D43);

%D5 MATRIX
D5111=-V1abs*V4abs*Y14abs*sin(V1ang-V4ang-Y14ang);
D5112=-V1abs*Y14abs*cos(V1ang-V4ang-Y14ang);
D5121=V1abs*V4abs*Y14abs*cos(V1ang-V4ang-Y14ang);
D5122=-V1abs*Y14abs*sin(V1ang-V4ang-Y14ang);
D51=[D5111 D5112;D5121 D5122];

D5211=-V2abs*V7abs*Y27abs*sin(V2ang-V7ang-Y27ang);
D5212=-V2abs*Y27abs*cos(V2ang-V7ang-Y27ang);
D5221=V2abs*V7abs*Y27abs*cos(V2ang-V7ang-Y27ang);
D5222=-V2abs*Y27abs*sin(V2ang-V7ang-Y27ang);
D52=[D5211 D5212;D5221 D5222];

D5311=-V3abs*V9abs*Y39abs*sin(V3ang-V9ang-Y39ang);
D5312=-V3abs*Y39abs*cos(V3ang-V9ang-Y39ang);
D5321=V3abs*V9abs*Y39abs*cos(V3ang-V9ang-Y39ang);
D5322=-V3abs*Y39abs*sin(V3ang-V9ang-Y39ang);
D53=[D5311 D5312;D5321 D5322];

D5=[D51 zeros(2,10);zeros(2,6) D52 zeros(2,4); zeros(2,10) D53];

%D6 MATRIX
D6111=-V4abs*V1abs*Y41abs*sin(V4ang-V1ang-Y41ang);
D6112=-V4abs*Y41abs*cos(V4ang-V1ang-Y41ang);
D6121=V4abs*V1abs*Y41abs*cos(V4ang-V1ang-Y41ang);
D6122=-V4abs*Y41abs*sin(V4ang-V1ang-Y41ang);
D61=[D6111 D6112;D6121 D6122];

D6211=-V7abs*V2abs*Y72abs*sin(V7ang-V2ang-Y72ang);
D6212=-V7abs*Y72abs*cos(V7ang-V2ang-Y72ang);
D6221=V7abs*V2abs*Y72abs*cos(V7ang-V2ang-Y72ang);
D6222=-V7abs*Y72abs*sin(V7ang-V2ang-Y72ang);
D62=[D6211 D6212; D6221 D6222];

D6311=-V3abs*V3abs*Y33abs*sin(V3ang-V3ang-Y33ang);
D6312=-V3abs*Y33abs*cos(V3ang-V3ang-Y33ang);
D6321=V3abs*V3abs*Y33abs*cos(V3ang-V3ang-Y33ang);
D6322=-V3abs*Y33abs*sin(V3ang-V3ang-Y33ang);
```

```
D63=[D6311 D6312; D6321 D6322];

D6=[D61 zeros(2,4); zeros(4,6);zeros(2,2) D62 zeros(2,2);zeros(2,6);zeros(2,4) D63];

%D7 MATRIX
D7411=V4abs*V5abs*Y45abs*sin(V4ang-V5ang-Y45ang)+V4abs*V6abs*Y46abs*sin(V4ang-V6ang-
Y46ang);
D7412=-V4abs*V4abs*Y44abs*cos(V4ang-V4ang-Y44ang)-V4abs*V5abs*Y45abs*cos(V4ang-V5ang-
Y45ang)-V4abs*V6abs*Y46abs*cos(V4ang-V6ang-Y46ang)-V4abs*V4abs*Y44abs*cos(V4ang-V4ang-
Y44ang);
D7421=-V4abs*V5abs*Y45abs*cos(V4ang-V5ang-Y45ang)-V4abs*V6abs*Y46abs*cos(V4ang-V6ang-
Y46ang);
D7422=-V4abs*V4abs*Y44abs*sin(V4ang-V4ang-Y44ang)-V4abs*V5abs*Y45abs*sin(V4ang-V5ang-
Y45ang)-V4abs*V6abs*Y46abs*sin(V4ang-V6ang-Y46ang)-V4abs*V4abs*Y44abs*sin(V4ang-V4ang-
Y44ang);
D711=[D7411 D7412;D7421 D7422];

D7413=-V4abs*V5abs*Y45abs*sin(V4ang-V5ang-Y45ang);
D7414=-V4abs*Y45abs*cos(V4ang-V5ang-Y45ang);
D7423=V4abs*V5abs*Y45abs*cos(V4ang-V5ang-Y45ang);
D7424=-V4abs*Y45abs*sin(V4ang-V5ang-Y45ang);
D712=[D7413 D7414;D7423 D7424];

D7415=-V4abs*V6abs*Y46abs*sin(V4ang-V6ang-Y46ang);
D7416=-V4abs*Y46abs*cos(V4ang-V6ang-Y46ang);
D7425=V4abs*V6abs*Y46abs*cos(V4ang-V6ang-Y46ang);
D7426=-V4abs*Y46abs*sin(V4ang-V6ang-Y46ang);
D713=[D7415 D7416;D7425 D7426];

D7511=-V5abs*V4abs*Y54abs*sin(V5ang-V4ang-Y54ang);
D7512=-V5abs*Y54abs*cos(V5ang-V4ang-Y54ang);
D7521=V5abs*V4abs*Y54abs*cos(V5ang-V4ang-Y54ang);
D7522=-V5abs*Y54abs*sin(V5ang-V4ang-Y54ang);
D721=[D7511 D7512;D7521 D7522];

D7513=V4abs*V5abs*Y54abs*sin(V5ang-V4ang-Y54ang)+V5abs*V7abs*Y57abs*sin(V5ang-V7ang-
Y57ang);
D7514=-V5abs*V5abs*Y55abs*cos(V5ang-V5ang-Y55ang)-V4abs*V5abs*Y54abs*cos(V5ang-V4ang-
Y54ang)-V5abs*V7abs*Y57abs*cos(V5ang-V7ang-Y57ang)-V5abs*V5abs*Y55abs*cos(V5ang-V5ang-
Y55ang);
D7523=-V4abs*V5abs*Y54abs*cos(V5ang-V4ang-Y54ang)-V5abs*V7abs*Y57abs*cos(V5ang-V7ang-
Y57ang);
D7524=-V5abs*V5abs*Y55abs*sin(V5ang-V5ang-Y55ang)-V4abs*V5abs*Y54abs*sin(V5ang-V4ang-
Y54ang)-V5abs*V7abs*Y57abs*sin(V5ang-V7ang-Y57ang)-V5abs*V5abs*Y55abs*sin(V5ang-V5ang-
Y55ang);
D722=[D7513 D7514;D7523 D7524];

D7517=-V5abs*V7abs*Y57abs*sin(V5ang-V7ang-Y57ang);
D7518=-V5abs*Y57abs*cos(V5ang-V7ang-Y57ang);
D7527=V5abs*V7abs*Y57abs*cos(V5ang-V7ang-Y57ang);
D7528=-V5abs*Y57abs*sin(V5ang-V7ang-Y57ang);
D724=[D7517 D7518;D7527 D7528];

D7611=-V6abs*V4abs*Y64abs*sin(V6ang-V4ang-Y64ang);
D7612=-V6abs*Y64abs*cos(V6ang-V4ang-Y64ang);
D7621=V6abs*V4abs*Y64abs*cos(V6ang-V4ang-Y64ang);
D7622=-V6abs*Y64abs*sin(V6ang-V4ang-Y64ang);
D731=[D7611 D7612;D7621 D7622];

D7615=V6abs*V4abs*Y64abs*sin(V6ang-V4ang-Y64ang)+V6abs*V9abs*Y69abs*sin(V6ang-V9ang-
Y69ang);
```

```
D7616=-V6abs*V6abs*Y66abs*cos(V6ang-V6ang-Y66ang)-V6abs*V4abs*Y64abs*cos(V6ang-V4ang-
Y64ang)-V6abs*V9abs*Y69abs*cos(V6ang-V9ang-Y69ang)-V6abs*V6abs*Y66abs*cos(V6ang-V6ang-
Y66ang);
D7625=-V4abs*V5abs*Y54abs*cos(V6ang-V4ang-Y64ang)-V6abs*V9abs*Y69abs*cos(V6ang-V9ang-
Y69ang);
D7626=-V5abs*V5abs*Y55abs*sin(V6ang-V6ang-Y66ang)-V4abs*V6abs*Y64abs*sin(V6ang-V4ang-
Y64ang)-V6abs*V9abs*Y69abs*sin(V6ang-V9ang-Y69ang)-V6abs*V6abs*Y66abs*sin(V6ang-V6ang-
Y66ang);
D733=[D7615 D7616;D7625 D7626];

D76111=-V6abs*V9abs*Y69abs*sin(V6ang-V9ang-Y69ang);
D76112=-V6abs*Y69abs*cos(V6ang-V9ang-Y69ang);
D76211=V6abs*V9abs*Y69abs*cos(V6ang-V9ang-Y69ang);
D76212=-V6abs*Y69abs*sin(V6ang-V9ang-Y69ang);
D736=[D76111 D76112;D76211 D76212];

D7713=-V7abs*V5abs*Y75abs*sin(V7ang-V5ang-Y75ang);
D7714=-V7abs*Y75abs*cos(V7ang-V5ang-Y75ang);
D7723=V7abs*V5abs*Y75abs*cos(V7ang-V5ang-Y75ang);
D7724=-V7abs*Y75abs*sin(V7ang-V5ang-Y75ang);
D742=[D7713 D7714;D7723 D7724];

D7717=V7abs*V2abs*Y72abs*sin(V7ang-V2ang-Y72ang)+V7abs*V8abs*Y78abs*sin(V7ang-V8ang-
Y78ang);
D7718=-V7abs*V7abs*Y77abs*cos(V7ang-V7ang-Y77ang)-V7abs*V2abs*Y72abs*cos(V7ang-V2ang-
Y72ang)-V7abs*V8abs*Y78abs*cos(V7ang-V8ang-Y78ang)-V7abs*V7abs*Y77abs*cos(V7ang-V7ang-
Y77ang);
D7727=-V7abs*V2abs*Y72abs*cos(V7ang-V2ang-Y72ang)-V7abs*V8abs*Y78abs*cos(V7ang-V8ang-
Y78ang);
D7728=-V7abs*V7abs*Y77abs*sin(V7ang-V7ang-Y77ang)-V7abs*V2abs*Y72abs*sin(V7ang-V8ang-
Y72ang)-V7abs*V8abs*Y78abs*sin(V7ang-V8ang-Y78ang)-V7abs*V7abs*Y77abs*sin(V7ang-V7ang-
Y77ang);
D744=[D7717 D7718;D7727 D7728];

D7719=-V7abs*V8abs*Y78abs*sin(V7ang-V8ang-Y78ang);
D77110=-V7abs*Y78abs*cos(V7ang-V8ang-Y78ang);
D7729=V7abs*V8abs*Y78abs*cos(V7ang-V8ang-Y78ang);
D77210=-V7abs*Y78abs*sin(V7ang-V8ang-Y78ang);
D745=[D7719 D77110;D7729 D77210];

D7817=-V7abs*V8abs*Y87abs*sin(V8ang-V7ang-Y87ang);
D7818=-V8abs*Y87abs*cos(V8ang-V7ang-Y87ang);
D7827=V8abs*V7abs*Y87abs*cos(V8ang-V7ang-Y87ang);
D7828=-V8abs*Y87abs*sin(V8ang-V7ang-Y87ang);
D754=[D7817 D7818;D7827 D7828];

D7819=V8abs*V7abs*Y87abs*sin(V8ang-V7ang-Y87ang)+V8abs*V9abs*Y89abs*sin(V8ang-V9ang-
Y89ang);
D78110=-V8abs*V8abs*Y88abs*cos(V8ang-V8ang-Y88ang)-V8abs*V7abs*Y87abs*cos(V8ang-V7ang-
Y87ang)-V8abs*V9abs*Y89abs*cos(V8ang-V9ang-Y89ang)-V8abs*V8abs*Y88abs*cos(V8ang-V8ang-
Y88ang);
D7829=-V8abs*V7abs*Y87abs*cos(V8ang-V7ang-Y87ang)-V8abs*V9abs*Y89abs*cos(V8ang-V9ang-
Y89ang);
D78210=-V8abs*V8abs*Y88abs*sin(V8ang-V8ang-Y88ang)-V8abs*V7abs*Y87abs*sin(V8ang-V7ang-
Y87ang)-V8abs*V9abs*Y89abs*sin(V8ang-V9ang-Y89ang)-V8abs*V8abs*Y88abs*sin(V8ang-V8ang-
Y88ang);
D755=[D7819 D78110;D7829 D78210];

D78111=-V8abs*V9abs*Y89abs*sin(V8ang-V9ang-Y89ang);
D78112=-V8abs*Y89abs*cos(V8ang-V9ang-Y89ang);
D78211=V8abs*V9abs*Y89abs*cos(V8ang-V9ang-Y89ang);
D78212=-V8abs*Y89abs*sin(V8ang-V9ang-Y89ang);
D756=[D78111 D78112;D78211 D78212];
```

```
D7915=-V9abs*V6abs*Y96abs*sin(V9ang-V6ang-Y96ang);
D7916=-V9abs*Y96abs*cos(V9ang-V6ang-Y96ang);
D7925=V9abs*V6abs*Y96abs*cos(V9ang-V6ang-Y96ang);
D7926=-V9abs*Y96abs*sin(V9ang-V6ang-Y96ang);
D763=[D7915 D7916;D7925 D7926];

D7919=-V8abs*V9abs*Y98abs*sin(V9ang-V8ang-Y98ang);
D79110=-V8abs*Y98abs*cos(V9ang-V8ang-Y98ang);
D7929=V9abs*V8abs*Y98abs*cos(V9ang-V8ang-Y98ang);
D79210=-V8abs*Y98abs*sin(V9ang-V8ang-Y98ang);
D765=[D7919 D79110;D7929 D79210];

D79111=V9abs*V6abs*Y96abs*sin(V9ang-V6ang-Y96ang)+V8abs*V9abs*Y98abs*sin(V9ang-V8ang-
Y98ang);
D79112=-V9abs*V9abs*Y99abs*cos(V9ang-V9ang-Y99ang)-V9abs*V6abs*Y96abs*cos(V9ang-V6ang-
Y96ang)-V9abs*V8abs*Y98abs*cos(V9ang-V8ang-Y98ang)-V9abs*V9abs*Y99abs*cos(V9ang-V9ang-
Y99ang);
D79211=-V9abs*V6abs*Y96abs*cos(V9ang-V6ang-Y96ang)-V8abs*V9abs*Y98abs*cos(V9ang-V8ang-
Y98ang);
D79212=-V9abs*V9abs*Y99abs*sin(V9ang-V9ang-Y99ang)-V9abs*V6abs*Y96abs*sin(V9ang-V6ang-
Y96ang)-V8abs*V9abs*Y98abs*sin(V9ang-V8ang-Y98ang)-V9abs*V9abs*Y99abs*sin(V9ang-V9ang-
Y99ang);
D766=[D79111 D79112;D79211 D79212];

D7=[D711 D712 D713 zeros(2,6);D721 D722 zeros(2,2) D724 zeros(2,4);D731 zeros(2,2) D733
zeros(2,4) D736;zeros(2,2) D742 zeros(2,2) D744 D745 zeros(2,2);zeros(2,6) D754 D755
D756; zeros(2,4) D763 zeros(2,2) D765 D766];




Ap=A1-B1*D1^-1*C1;
K1=D4-D3*D1^-1*D2;
K2=C2-D3*D1^-1*C1;
B1p=B2-B1*D1^-1*D2;
Jae=[[K1; D6] [D5 ;D7]];
J1=Jae^-1*[K2;zeros(12,21)];
A=Ap-[B1p zeros(21,12)]*Jae^-1*[K2;zeros(12,21)];
C=[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0];

sys1=ss(A,E1,C,[0]);
```

APPENDIX D.2

3-MACHINE 9-BUS LQR CONTROLLER DESIGN AND MODEL REDUCTION

```
sys1_red=ss(Ar12,Br12,Cr12,[0]);
[Ared,Bred,Cred,Dred]=ssdata(sys1_red);
hsv=hsvd(sys1);
sigma=[1 hsv(1)/hsv(2) hsv(1)/hsv(3) hsv(1)/hsv(4) hsv(1)/hsv(5) hsv(1)/hsv(6)
hsv(1)/hsv(7)   hsv(1)/hsv(8)   hsv(1)/hsv(9)   hsv(1)/hsv(10)   hsv(1)/hsv(11)
hsv(1)/hsv(12)];
%sigma=[1 hsv(2)/hsv(1) hsv(3)/hsv(1) hsv(4)/hsv(1) hsv(5)/hsv(1) hsv(6)/hsv(1)
hsv(7)/hsv(1)   hsv(8)/hsv(1)   hsv(9)/hsv(1)   hsv(10)/hsv(1)   hsv(11)/hsv(1)
hsv(12)/hsv(1)];
Q=diag(sigma);
opt = balredOptions('StateElimMethod','Truncate');
sys2_redBal=balred(sys1,12);
[AredBal,BredBal,CredBal,DredBal]=ssdata(sys2_redBal);
Qfull=diag([1 10 0 0 0 0 0 1 10 0 0 0 0 0 1 10 0 0 0 0 0] );
Rfull=eye(6);
Kfull=lqr(sys1,Qfull,Rfull);
%%
sys1_1=ss(A,E1(:,1),eye(21),[0]);
hsv1_1=hsvd(sys1_1);
w1=trace(diag(hsv1_1));

sys1_2=ss(A,E1(:,2),eye(21),[0]);
hsv1_2=hsvd(sys1_2);
w2=trace(diag(hsv1_2));

sys1_3=ss(A,E1(:,3),eye(21),[0]);
hsv1_3=hsvd(sys1_3);
w3=trace(diag(hsv1_3));

sys1_4=ss(A,E1(:,4),eye(21),[0]);
hsv1_4=hsvd(sys1_4);
w4=trace(diag(hsv1_4));

sys1_5=ss(A,E1(:,5),eye(21),[0]);
hsv1_5=hsvd(sys1_5);
w5=trace(diag(hsv1_5));

sys1_6=ss(A,E1(:,6),eye(21),[0]);
hsv1_6=hsvd(sys1_6);
w6=trace(diag(hsv1_6));

gamma=w1/hsv(1);

r=gamma.*[1 w2/w1 w3/w1 w4/w1 w5/w1 w6/w1];
R=diag(r);

Kred=lqr(sys2_redBal,Q,R);
```