# Power System Simulation Using a Differential Transformation Method

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Yang Liu

May 2022

Copyright © 2022 by Yang Liu

All rights reserved.

## ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor Prof. Kai Sun for the funding support, continuous guidance, and persistent help for this dissertation and all other research during my Ph.D. study at the University of Tennessee, Knoxville (UTK). Through our meetings and discussions, his insights have shaped my understanding of many topics in power systems and nonlinear dynamic systems. I want to thank him for providing the wonderful opportunity of visiting Harvard University and many other opportunities for me to open my mind, meet new people and learn new research areas. I also wish to thank him for sharing his priceless industry and academic experiences, and for all the support and suggestions on my career, my life, and how to be a better person. I was very fortunate to have Prof. Kai Sun as my Ph.D. advisor and I could not have imagined having better advisors for my Ph.D. study.

I would like to extend my sincere thanks to Prof. Kevin Tomsovic, Prof. Fangxing (Fran) Li, and Prof. Xueping Li for serving on my Ph.D. committee. I also want to thank Prof. Na Li at Harvard University for hosting me as a visiting student.

I appreciate the helpful discussions with the students in Dr. Kai Sun's group, Dr. Bin Wang, Dr. Rui Yao, Dr. Chengxi Liu, Dr. Nan Duan, Dr. Wenyun Ju, Dr. Yongli Zhu, Dr. Denis Osipov, Dr. Weihong Huang, Dr. Xin Xu, and Tianwei Xia, Min Xiong, Dr. Changsheng Chen, Dr. Li Sun, Dr. Yichen Guo, Dr. Guoqiang Zu.

I also had the great pleasure of working with Dr. Srdjan Simunovic, Dr. Park Byungkwon, Dr. Srikanth Allu, and Dr. Aleksandar Dimitrovski on a project with the

Oak Ridge National Laboratory, and special thanks to them for the interesting and productive collaboration.

I very much appreciate my master's advisor Prof. Zheng Li in Tsinghua University, who has not only taught me how to do research but also taught me how to shape better character. I also want to thank Prof. Pei Liu at Tsinghua University for the encouragement and help that make my Ph.D. study possible.

I would like to thank Dr. Weihong Huang and Dr. Zheyu Zhang for their invaluable help in my life. I would not come to UTK without their help. I would like to thank Prof. Jie Chen and her family, for teaching me many life skills and for helping me get through the first several months in the U.S. I want to thank my parents and parents-in-law for their priceless patience and support during my graduate studies. Finally, I want to thank my wife, Dr. Jiaojiao Dong, who has been extremely supportive of me throughout this entire process and has made countless sacrifices to help me get to this point.

This work was supported in part by UT-Battelle, LLC under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy, in part by the National Science Foundation under Grant No. ECCS-1610025, and in part by the Engineering Research Center Program of the National Science Foundation and the Department of Energy under Grant No. EEC-1041877.

## ABSTRACT

This work proposes a novel approach for efficient and robust power system simulation based on differential transformation (DT).

First, this work introduces the DT to study power systems as high-dimensional nonlinear dynamical systems for the first time. This work designs a DT-based high-order semi-analytical simulation scheme that allows significantly prolonged time steps to reduce simulation time compared to a traditional numerical approach. The numerical stability, accuracy, and time performance of the proposed approach are compared with widely used numerical methods on the IEEE 39-bus system and Polish 2383-bus system.

Second, this work proposes a novel non-iterative method to solve power system differential-algebraic equations (DAEs) using the DT. The non-state variables (e.g. current injections and bus voltages), nonlinearly coupled in network equations, are conventionally solved by numerical methods with time-consuming iterations, but their DTs are proved to satisfy formally linear equations in this work. Thus, a non-iterative algorithm is designed to analytically solve all variables of a power system DAE model. From test results on a Polish 2383-bus system, the proposed method demonstrates fast and reliable time performance compared to traditional numerical approaches.

Third, this work proposes a novel dynamized power flow (DPF) method for solving and tracing power flow solutions. Different from the conventional continuation power flow (CPF) method, the proposed method extends the power flow model into a fictitious dynamic system by adding a differential equation on the loading parameter. A non-iterative algorithm based on DT is proposed to analytically solve the dynamized

model in form of power series of time. Case studies on several test systems including a 2383-bus system show the merits of the proposed method.

Fourth, this work integrates the DT into Parareal, a Parallel in time framework, as the first step towards industrial applications. The case studies on large systems demonstrate that the combination of DT and Parareal provide a promising direction towards faster-than-real-time simulation.

Besides, this work also employs the DT for adaptive frequency control of wind turbines and further generalizes the DT method for general nonlinear DAEs.

**Keywords:** Differential transformation; power system simulation; transient stability; differential-algebraic equations; power flow; voltage stability.

# TABLE OF CONTENTS

CHAPTER 1 Introduction ..... 1

1.1 Background and Motivation ..... 1

1.1.1 Review of Power System Models ..... 4

1.1.2 Review of Parallel Computing Methods ..... 8

1.1.3 Review of Semi-analytical Methods ..... 9

1.2 Contributions of This Work ..... 11

CHAPTER 2 Solving Ordinary Differential Equation Model using Differential Transformation ..... 16

2.1 Introduction of the Differential Transformation ..... 16

2.2 DTs of Power System Models ..... 18

2.2.1 New Transform Rules for Power System Models ..... 19

2.2.2 Governor Model ..... 21

2.2.3 Turbine Model ..... 22

2.2.4 PSS Type I Model ..... 22

2.2.5 Synchronous Machine Model ..... 23

2.2.6 IEEE Type I Exciter Model ..... 24

2.3 Proposed Scheme for Dynamic Simulation ..... 26

2.3.1 DT-based Solution Scheme ..... 26

2.3.2 Remarks ..... 29

2.4 Case Study ..... 30

2.4.1 Scanning Various Contingencies ..... 31

2.4.2 Comparison of Numerical Stability ..... 32

2.4.3 Comparison of Accuracy and Time Performance ..... 34

2.5 Conclusion ..... 35

CHAPTER 3 Solving Differential Algebraic Equation Model using Differential Transformation ..... 36

3.1 Proposed Method for Solving Power System DAE Model Using DT ..... 36

3.1.1 Conceptual Description of the Proposed Method ..... 36

3.1.2 Step 1: Deriving DTs of Power System DAE Model ..... 37

3.1.3 Step 2: Solving Power Series Coefficients of State Variables and Bus Voltages ..... 38

3.2 DTs of Power System DAE Model ..... 39

3.2.1 Vectorized Transformation Rules ..... 40

3.2.2 DTs of the Current Injection Equation of Generators ..... 41

3.2.3 DTs of the Current Injection Equation of Loads ..... 42

3.2.4 DTs of the Network Equation ..... 44

3.3 Solving Power Series Coefficients of State Variables and Bus Voltages ..... 45

3.3.1 Linear Relationship Between Current Injection and Bus Voltage in terms of Power Series Coefficients ..... 45

3.3.2 Non-iterative Algorithm to Solve Power Series Coefficients ..... 48

3.3.3 Extension ..... 49

3.4 Case Study ..... 51

3.4.1 Illustration on a 3-machine 9-bus Power System ..... 52

3.4.2 Accuracy and Time Performance ..... 54

3.4.3 Robustness ..... 56

3.5 Conclusion ..... 58

CHAPTER 4 A Dynamized Power Flow Method based on Differential Transformation ..... 59

4.1 Problem statement ..... 59

4.2 Proposed Dynamized Power Flow Method ..... 60

4.2.1 Idea of the proposed method ..... 60

4.2.2 Step 1: Dynamized Power Flow Equation ..... 63

4.2.3. Step 2: Deriving Different Transformation ..... 64

4.2.4. Step 3: Recursively Solving Variables As Power Series Of Time ..... 70

4.3 Case Studies ..... 72

4.3.1. Demonstration on the 9-bus Power System ..... 73

4.3.2. Accuracy, Computation Complexity, And Time Performance On Large Systems ..... 74

4.3.3. Application To N-1 Contingency Analysis ..... 75

4.4 Conclusion ..... 75

CHAPTER 5 A DT based Adaptive Parareal Method for Power System Simulation ..... 77

5.1 Proposed Method ..... 77

5.1.1. Conventional Numerical-based Parareal Method ..... 78

5.1.2. DT-based Parareal Method ..... 80

5.1.3. VOVS-DT Method ..... 82

5.1.4. VW-Parareal ..... 83

5.1.5. Improved Coarse Update Strategy ..... 85

5.1.6. Integrated Approach ..... 86

5.2 Case Study on a 39-bus system ..... 86

5.2.1. Conventional Numerical-based Parareal Method ..... 87

5.2.2. DT-based Parareal Method ..... 87

5.2.3. VW-Parareal ..... 88

5.2.4. Improved Coarse Update Strategy ..... 89

5.2.5. VOVS-DT Method ..... 89

5.3 Case Study on a 2383-bus system ..... 90

5.4 Conclusion ..... 91

CHAPTER 6 DT Based Adaptive Switching Control of Wind Turbines ..... 93

6.1 Problem Description ..... 93

6.2 Proposed Adaptive Switching Control Strategy ..... 95

6.2.1. Proposed Switching Control Strategy ..... 95

6.2.2. System Model ..... 96

6.2.3. Frequency Response Predictor ..... 97

6.3 Case Study ..... 99

6.3.1. Test on a 4-bus System ..... 99

6.3.2. Test on the 39-bus System ..... 101

6.4 Conclusion ..... 102

CHAPTER 7 An Affine Recursion Form of Nonlinear DAEs ..... 103

7.1 ARF of Compositional Functions ..... 103

7.1.1. Definition ..... 103

7.1.2. Existence and Uniqueness ..... 106

7.1.3. Derivation ..... 107

7.2 ARF of DAEs ..... 113

7.3 Application to Power System DAEs ..... 116

7.3.1. ARF of Load Model ..... 118

7.3.2. ARF of Generator Model ..... 120

7.3.3. ARF of Power System DAEs ..... 121

7.4 Conclusion ..... 122

CHAPTER 8 CONCLUSION ..... 123

LIST OF REFERENCES ..... 127

APPENDICES ..... 134

APPENDIX A – Figures ..... 135

APPENDIX B – Tables ..... 169

VITA ..... 178

# LIST OF TABLES

Table 1. Comparison of Numerical Stability under Three Scenarios ..... 170

Table 2. Comparison of Maximum Time Step Length to Maintain the Numerical Stability (Unit: s) ..... 170

Table 3. Comparison of Time Step Length Under Different Error Tolerances (Unit: s) 170

Table 4. Comparison of Computation Time Under Different Error Tolerances (Unit: s) ..... 171

Table 5. Comparison of Accuracy and Time Performance..... 171

Table 6. Accuracy and Time Performance for Different Values of $K$ ..... 171

Table 7. Comparison of Number of LU Factorization..... 171

Table 8. Numbers of Times of Solving Linear Equations ..... 172

Table 9. Comparison of Time Performance (Unit: second)..... 172

Table 10. Average Number of Iterations Using the Conventional Parareal Algorithm.. 172

Table 11. Average Number of Iterations Using the DT-based Parareal Method..... 173

Table 12. Average Number of Iterations using the VW-Parareal Method ..... 173

Table 13. Average Window Length Using the VW-Parareal Method..... 174

Table 14. Average Number of Iterations Using the VW-Parareal Method with Improved Coarse Update Strategy..... 175

Table 15. Average Window Length Using the VW-Parareal Method with Improved Coarse Update Strategy..... 176

Table 16. Average Time Step Length of 39-bus System (Seconds) ..... 176

Table 17. Average Order of 39-bus System..... 177

Table 18. Average Order of 2383-bus System..... 177

Table 19. Operation Status of Frequency Support Mode ..... 177

Table 20. Settings of the Three Scenarios ..... 177

# LIST OF FIGURES

Figure 1. 39-bus system under a stable contingency ..... 136

Figure 2. 39-bus system under a marginal stable contingency. ..... 137

Figure 3. 39-bus system under marginal unstable contingency. ..... 138

Figure 4. 2383-bus system under a stable contingency. ..... 139

Figure 5. 2383-bus system under a marginally stable contingency. ..... 140

Figure 6. 2383-bus system under a marginally unstable contingency. ..... 141

Figure 7. Accuracy of the DT method when faults occur at all buses. ..... 142

Figure 8. Accuracy of the DT method when faults occur at all branches. ..... 142

Figure 9. Accuracy of the DT method with and without PSS. ..... 143

Figure 10. Accuracy of the DT method with and without saliency. ..... 143

Figure 11. Error propagation for the three scenarios with different time steps ..... 144

Figure 12. Recursive process to solve power series coefficients ..... 145

Figure 13. Trajectories and errors of bus voltages for the 9-bus system (a) Voltage trajectories at bus 1 and bus 5 (b) Maximum voltage error of all 9 buses ..... 145

Figure 14. Trajectories of the stable scenario for the 2383-bus system ..... 146

Figure 15. Trajectories of the unstable scenario for the 2383-bus system ..... 147

Figure 16. Error propagation under different time step lengths ..... 148

Figure 17. Robustness against different disturbances ..... 148

Figure 18. Robustness against different percentages of constant power load ..... 148

Figure 19. Illustration of the two dynamized formulations for tracing curve segments of power flow equations ..... 149

Figure 20. Time domain trajectory of the dynamized power flow model ..... 149

Figure 21. Solution curve of load bus 9 on 9-bus system ..... 150

Figure 22. Solution curves under non-uniform load change direction, a) not consider reactive power limit, b) consider reactive power limit ..... 151

Figure 23. Solution curves on 39-bus system ..... 152

Figure 24. Solution curves on 57-bus system ..... 152

Figure 25. Solution curves on 2383-bus system ..... 152

Figure 26. Maximum loading conditions by CPF and DPF ..... 153

Figure 27. Illustration of the Parareal method ..... 153

Figure 28. Structure of the proposed method ..... 154

Figure 29. Illustration of the DT method ..... 154

Figure 30. Illustration of the VW-Parareal method ..... 155

Figure 31. Trajectories of rotor angles, rotor speeds, bus voltages and electrical power in scenario 2 ..... 156

Figure 32. Evolution of time-step length and order of DT over the simulation ..... 157

Figure 33. Evolution of time-step length and order of DT over the simulation ..... 158

Figure 34. Evolution of time-step length and order of DT over the simulation ..... 158

Figure 35. Evolution of time-step length and order of DT over the simulation ..... 159

Figure 36. System frequency response with and without wind power ..... 159

Figure 37. Frequency response under various disturbances ..... 160

Figure 38. Diagram of four-bus test system ..... 160

Figure 39. Comparisons of DTM and RK4 ..... 161

Figure 40. Frequency responses using DTM-based semi-analytical solutions with different orders ..... 162

Figure 41. Switching of the frequency support mode with and without safety check( 1 – switched; 0 – unswitched ) ..... 162

Figure 42. Frequency response when WTGs are switched to frequency support modes at $\mathrm{D}w = 0.2$ HZ ..... 163

Figure 43. Switching strategy for the safe and unsafe frequency response ..... 164

Figure 44. a) DAG representation of one-input one-output one-layer composition function b) DAG representation of the transformed function equipped with a linear structure ..... 165

Figure 45. a) DAG representation of one-input one-output multi-layer composition function b) DAG representation of the transformed function equipped with a linear structure ..... 165

Figure 46. a) DAG representation of multi-input multi-output multi-layer composition function b) DAG representation of the transformed function equipped with a linear structure ..... 166

Figure 47. a) DAG representation of the compositional function in load model b) DAG representation of the transformed function. ..... 167

Figure 48. a) DAG representation of the compositional function in generator model b) DAG representation of the transformed function. ..... 168

# CHAPTER 1

# INTRODUCTION

## 1.1 Background and Motivation

The power system is regarded as one of the largest manmade systems in the world. For example, the power system in the US involves about 3,500 utility organizations, 200,000 miles of transmission lines, and 300 million customers [1]. Its reliable and economic operation directly affects the economy of the whole country and our daily life. To guarantee reliable and economic operation, power engineers need to perform a lot of computations almost every day with a large range of time scales in magnitude from microseconds to years [2].

The applications that involve frequent computations include [2] but are not limited to 1) steady-state analysis such as the power flow, state estimation, electricity market, etc. 2) power system planning (in years) and scheduling (in hours). 3) dynamic simulation, e.g., the voltage stability (in minutes), the electro-mechanical simulation (in seconds or milliseconds), the electro-magnetic simulations (in microseconds). Essentially, all these applications are facing computational problems, either computational speed or computational quality, or both.

Among the various power system applications, the dynamic security assessment (DSA) is one of the most time-demanding ones. The task of DSA is to assess the various security criteria with a given contingency occurring under a specific operating condition. These criteria generally include transient stability assessment, small-signal stability

assessment, and voltage stability assessment [3]-[5]. Especially, transient stability assessment (TSA) is the most time-consuming task in DSA.

Dynamic simulation is the most powerful tool in power system DSA, especially for the TSA. The goal of the dynamic simulation is to judge if a power grid can remain stable suffering from a given contingency. It is of great importance for the grid control center to improve situational awareness and take proactive control actions [6]. The model for dynamic simulation is originally a set of highly stiff ordinary differential equations (DEs) with a large span of time constants. Due to the model complexity and numerical stability issue, the dynamic simulation is usually performed separately for the electromechanical simulation with time constants of seconds or milliseconds and electromagnetic simulation with time constants of microseconds. This means the electromagnetic dynamics are neglected when simulating the electromechanical dynamics, and it leads to a differential-algebraic equation (DAE) model [7].

Despite the time scale separation, it is still a big challenge to solve the DAE model for a large-scale power system. Traditionally, the differential equations are solved by a numerical integration method, and the algebraic network equations are solved by a numerical iteration method at each integration step [1]-[4]. These methods may suffer from huge computation burdens caused by the iterations at each integration step for the convergence of the network equations and the large number of integration steps to ensure the accuracy and numerical stability of solving the differential equations. Moreover, the computation speed can further deteriorate when system states change significantly, or the system model has strong nonlinearity since the network equation is more difficult or even

fails to converge by numerical iteration methods. For example, using commercial software, completing a 20 seconds dynamic simulation for Western Electricity Coordinating Council (WECC) system using a base case model takes around 100 seconds [8]. Considering the time of recording and saving bus voltage trajectories to a data file, the time cost would be more than 400 seconds [8]. The computation time could be further increased if a more detailed system model is considered. Therefore, the simulation is mainly carried out offline on a daily or hourly basis. However, the offline assessment result is credible only if the real-time operating point does not deviate much from the anticipated condition.

In recent years, the power systems have been pushed to be operated closer to their stability limits due to the fast growth in electricity demands but a relatively slow construction of new transmission infrastructure. Moreover, the future smart grid will have more uncertain and diversified conditions with high penetration of renewable energy resources and responsive loads. To meet these uncertainties, one of the future directions for dynamic simulation is to have the “look-ahead” capability meaning the simulation is performed faster than real time so that the control center can predict the insecure status by simulation before it really happens. Therefore, it is more urgent than before to improve the computational speed and conduct dynamic simulation in real time. The power industry and the research community are seeking next-generation simulation tools which are more powerful for power system dynamic security assessment in real time.

Existing methods to speed up power system simulation fall into three categories. i.e., 1) the model reduction, 2) the parallel computing based on traditional numerical

methods and 3) the semi-analytical solution. The model reduction is from the model perspective while the parallel computing and semi-analytical solution are from the algorithm perspective.

Note that the model reduction on its own is a broad topic [9] and is beyond the scope of this work. In this category, both the differential equations and algebraic equations of a DAE model can be simplified. For instance, a widely used coherency-based model reduction technique aggregates a group of coherent generators into an equivalent generator [9]-[10]. Also, to avoid solving nonlinear algebraic network equations separately from solving differential equations, many simulation tools assume all constant impedance loads so as to eliminate the network equations of a DAE model and obtain an ordinary differential equation model [11]. However, methods of this category can bring substantial errors in simulation.

The remaining part of this section will give a detailed review of the power system models, Parallel computing techniques, and semi-analytical methods.

### 1.1.1 Review of Power System Models

There are two types of DAE models in power systems, i.e, the current injection model and the power injection model [11]-[12]. Both the current injection model and power injection model are commonly used in commercial software. The current injection model has the advantage of less nonlinearity, higher sparsity, and the capability of dimension reduction. As a comparison, the power injection model has the advantage of less additional computation burden and easier programing.

Since the current injection model allows one to reduce the dimensions of AEs with constant impedance assumptions, it can be further classified into two categories: the full DAE model and the reduced **Y**-bus model [12], [13]. The reduced Y-bus model is subjected to the constant impedance load assumption. For transient stability study in the time frame of several seconds, this assumption is reasonable since the load controls usually take a long time and the load impedance can be regarded as constant for the several seconds following a disturbance. Besides, the reduced Y-bus model can greatly reduce the computation burden since the linear AEs can be explicitly solved and substituted into the DEs, and one only needs to handle a set of ODEs instead of DAEs.

In recent years, with the increasing penetration of renewables, the model of power electronics interfaced devices and the loads are becoming more and more important in dynamic simulation. Below we review these models and analyze their impact on power system dynamic simulation.

1) Renewable Energy Sources

In modern and future power systems, there will be more penetration of renewables that are interfaced with power electronic devices. One typical renewable energy source is wind turbine generators. Generally, a complete wind turbine model involves the wind speed model, the wind turbine model, voltage source converter devices as well as various controls. The wind speed is usually modeled as Weibull's distribution to consider the stochastic feature. The wind turbine has four types of model in the literature, referred to Type A, B, C, and D, respectively [14].

- Type A wind turbines can be characterized by fixed-speed induction generators. Because of the stochastic feature of wind speed, the power injected into the network usually fluctuates seriously, which poses great challenges to grid integration.
- Type B wind turbines can adjust the rotor circuit parameters and have limited capability to be operated with variable speeds. It is usually characterized as a variable speed inductor generator.
- Type C wind turbines or the doubly-fed induction generators (DFIG) have a better capability of variable speed operation due to better control of rotor circuits. They can efficiently reduce the variation of injected power and is the most commonly studied in the literature.
- Type D wind turbines are fully converter-based configurations without a gearbox. In this type of wind turbine, the system frequency on the generator sides and the network side are fully decoupled.

In the dynamic simulation, wind turbine generators (WTG) can be represented by a set of DAEs. The detailed formulation of DAEs depends on the control strategies since there are many control schemes for WTGs proposed in the literature. For example, wind turbines are normally operated in the maximum power point tracking (MPPT) mode to extract as much power as possible. However, due to the low inertia of WTG, the system frequency response is usually inadequate. Therefore, many control strategies are proposed in the literature for WTGs to participate in the frequency response [15]-[19], including the inertia response and primary frequency response. In these cases, the power reference needs to be changed from the MPPT to other modes, for example, by

temporarily injecting additional power or by normally operating WTGs at the de-loaded condition to have some margin [16].

2) Impact of Power Electronics

In modern power systems, the power electronic devices are playing more and more important roles. Except for the renewable energy resources integration, there are many other power electronics applications [2], for example, the HVDC and the FACTS techniques. It is becoming more and more important to consider their impact in dynamic simulation. The impacts of considering these power electronic devices are summarized as two aspects.

First, the time scale of the whole system becomes wider than before due to the very fast dynamics caused by power electronic devices. This can result in a highly stiff dynamic system, which cannot be efficiently handled by traditional numerical algorithms. It is very urgent to develop more efficient simulation algorithms to handle such systems with a wider range of time scales.

Second, the hybrid behavior would be more frequent and more important. The power electronics allow better control and provide devices more flexibility to switch among several modes. For example, the future WTGs may need to work at both MPPT mode and the frequency support mode. These hybrid behavior needs to be well studied by both analytical approaches and simulation approaches.

3) Load Models

The load models are becoming increasingly important in recent years. The basic task of load modeling is to find a mathematical relationship between the power

consumption and the bus voltage at a given load bus. Depending on the different forms of mathematical relationships, load models include static load models and dynamic load models. The model parameters can be identified by the measurement method and component method.

The static load model is formulated as algebraic equations among power consumption, bus voltage, and system frequency. The typical relationships include the ZIP load model, exponential load model, frequency-dependent load model, etc. The dynamic load model mainly includes the induction motor load.

In recent literature, the composite load models are also widely studied combing both static and dynamic load models. In the dynamic study, a typically used model is a combination of the ZIP model and induction motor model. It is easy to implement and integrate with commercial tools. There are also some other composite load models in the literature. For example, the CLOD model combines large and small motors, discharge motors, transformer saturation, constant MVA, etc. It has been adopted by PSS/E software. Also, a more complicated WECC CLM model is proposed in the literature and is tested in PSS/E. But the WECC CLM model needs to identify 131 parameters and hence it is more difficult to implement [20].

### 1.1.2 Review of Parallel Computing Methods

Methods in this category employ parallel computers to speed up the simulation, which decomposes the DAE model or computation tasks onto multiple processors such as the Parareal in time method [21]-[22], multi-decomposition approach [23], the domain decomposition method [24]-[25], the waveform relaxation method [26], the instantaneous

relaxation method [27]-[28], the multi-area Thevenin equivalent method [29]-[30], and the practical parallel implementation techniques in [31]. Specifically, paper [23] decomposes a system model into three linear subsystems and ensures simulation accuracy by adaptive updates on linear subsystems. paper [26] proposes a two-stage parallel waveform relaxation method for parallel simulation, which adopts epsilon decomposition to partition a large-scale power system model into several subsystems. A Schur-complement-based network decomposition method is proposed in [24]-[25]. The parareal in time method in [21] and [22] adopts temporal decomposition of the simulation period into many intervals, conducts parallel simulations on individual intervals using a fine solver, and connects their results by a high-level coarse solver after a few iterations. A multi-area Thevenin equivalents method is also studied in [29]. Recently, paper [8] develops a practical framework to parallelize computation tasks of a single dynamic simulation in commercial software, and paper [31] designs a massively parallel computational platform for efficient dynamic security assessment. However, these methods still rely on the traditional numerical algorithms to solve DAEs, thus still requiring small-enough integration steps and numerical iterations. Moreover, the potential of parallel computing has been extensively studied in recent years and the room for further speeding up the simulation using parallel computing has been very limited under the paradigm of numerical methods.

### 1.1.3 Review of Semi-analytical Methods

For the semi-analytical solution methods in the third category, the basic idea is to derive semi-analytical solutions of the power system model in the form of a summation

of many terms so as to provide additional room for parallelization. The existing methods to derive semi-analytical solutions in power system field [32]-[48] include but are not limited to Adomian decomposition method [32]-[34], power series method [35]-[36], Pade approximation method [37]-[38], continued fraction method [39], holomorphic embedding method [40]-[41], homotopy analysis method [42], and the differential transformation method [43]-[48] proposed in this work.

The semi-analytical solution methods contain two stages. The first stage offline derives a semi-analytical solution (SAS) that expresses each state variable as an explicit function of symbolic variables including time, the initial state, and parameters on system conditions. The SAS is an approximate but analytical solution being accurate for a certain time window whose length depends on the inherent nonlinearity and order of the SAS expression. The second stage online evaluates the SAS over consecutive time windows to make up the desired simulation period by substituting values for those symbolic variables according to a given contingency and the real-time system condition.

Three methods, i.e. Taylor Expansion (TE), Adomian Decomposition Method (ADM), and Pade Approximants (PA), have been applied to power system models to derive SAS's respectively in the forms of polynomials of time, sinusoidal or polynomial functions of time, and fractional functions of time. Ref. [40] further integrates ADM-based SAS's with numerical integration methods for a hybrid strategy and tests it on realistic large system models. The speed of simulation using a SAS relies on the time window to maintain its accuracy, which further relies on its order, i.e. the number of summated terms, of its expression. However, when applied to a multi-machine power

system, a SAS derived from the above three methods has considerably increased complexity especially after its order exceeds three.

The major limitations of existing semi-analytical solution methods are summarized as follows. First, existing methods to derive semi-analytical solutions are very complicated and have limited scalability to large systems. Second, the derived SASs by existing methods are expanded into very long expressions, which are not efficient in evaluation and the shifted computation burden to the offline stage does not translate to increased performance in the online stage. Third, the huge computation burdens in offline stage and online stage prevent the existing semi-analytical methods from using high order approximation, thus the time step length using existing semi-analytical methods are not significantly improved compared with the numerical integration methods. Fourth, the existing semi-analytical methods focused on solving the differential equation, which only takes minority of the computation time, while solving the nonlinear algebraic network equations, which takes majority of the computation time, still relies on the numerical iteration methods.

## 1.2 Contributions of This Work

This dissertation aims at developing efficient and robust power system simulation algorithms. Specifically, this work aims at addressing the fundamental computational challenges of solving high dimensional nonlinear differential algebraic equation model in transient stability analysis, and its two variations, i.e., ordinary differential equation model in transient stability simulation under the assumption of constant impedance load, and the algebraic AC power flow equation model in voltage stability analysis. Some other

applications in power systems and general nonlinear dynamic systems are also explored. The contributions of this work are as follows.

First, this dissertation proposes a novel dynamic simulation approach based on Differential Transformation (DT), which is a mathematical tool and can effectively find an approximate solution of a set of nonlinear DEs. This work, for the first time, introduces DT to power system studies. By assuming the solution of a set of power system DEs as a power series in time, the proposed approach utilizes DT to calculate series coefficients efficiently by means of a set of transform rules designed for nonlinear functions involved in power system models instead of directly computing the high-order derivatives in DEs. These transform rules are proved for representative power system models including a detailed synchronous machine model involving trigonometric functions and a practical exciter model with the exponential and square root functions. The dissertation also proposed a DT-based dynamic simulation scheme that allows significantly prolonged time steps to reduce the overall simulation time compared to a traditional numerical approach.

Second, this dissertation proposes a novel non-iterative method to solve the DAE model of a large-scale power system using the DT method. First, we derive the DTs of the algebraic network equations with current injections. Then, we prove that current injections and bus voltages which are coupled by the original nonlinear network equations, satisfy a formally linear equation in terms of their power series coefficients after DT. Further, a non-iterative algorithm is designed to analytically solve both state

variables and non-state variables by power series of time. Simulation results show the proposed method is fast and reliable compared to traditional methods.

Third, to trace solution curves of power flow equations more efficiently, this dissertation proposes a novel dynamized power flow (DPF) method that extends the power flow model into a fictitious dynamic system, called a “dynamized” power flow model, by adding a differential equation about a fictitious time, and then solve the complete time-domain trajectory of the dynamic system instead of repeatedly solving power flow equations for a series of conditions. The DT method is applied to solve the dynamized model, named as dynamized power flow method. This work proves that the nonlinear AC power flow equations are converted to formally linear equations after DT, and further designs an efficient algorithm to solve the time domain trajectory without numerical iterations. Case studies on several test systems including a 2383-bus system demonstrate the accuracy, computational complexity and time performance of the proposed approach compared with a CPF solver.

Fourth, this dissertation examined the feasibility of combining the DT method and the Parareal algorithm to further speed up the dynamic simulation. A DT- based variable-order variable-step variable-window adaptive parareal method is proposed for temporal parallelization of power system simulation with greatly enhanced convergence performance and efficiency. The proposed method integrates the temporal parallelization capability of the Parareal method and the highly adaptive feature of the DT method. Extensive simulations on a 39-bus system and a 2383-bus system demonstrate the effectiveness of the proposed approach.

Fifth, this work proposes a novel switching control strategy which predicts the safety of a frequency response right after a disturbance by evaluating the derived semi-analytical solutions of system frequency response model over a certain post-disturbance period of interest and activates frequency support mode only when the frequency response is predicted as unsafe. The rationale of the proposed strategy is real time evaluation of an offline obtained semi-analytical solution on frequency responses using real-time measurements. Such a semi-analytical solution is in form of ultra-high order Taylor series derived by the DT method on the differential equation model of the system. The ultra-high order nature of the solution enables its largely extended convergence region to cover the frequency response period of interest, so that the frequency response of a WTG can be accurately predicted when a disturbance is detected and the WTG provides frequency support only for an unsafe response, thus avoiding the unnecessary switches. The case studies on a 4-bus power system and a New England 10-machine 39-bus system show the effectiveness of the proposed strategy.

Finally, this work proposes a DT-based affine recursion form (ARF) of general nonlinear DAEs. The advantage of ARF is that nonlinear DAEs are converted to formally linear equations about Taylor series coefficients, which enables straightforward calculation of semi-analytical solutions of the nonlinear DAEs in the form of arbitrary high-order Taylor series. Since practical DAE models often contain compositional functions with complicated compositional structures, this work first studies the ARF of generic compositional functions, including its existence, uniqueness, and propagation of the ARF over a compositional function structure to derive ARFs of compositional

functions from simple functions. After that, this work derives the ARF of nonlinear DAEs and designs an ARF-based DAE solution algorithm.

# CHAPTER 2

# SOLVING ORDINARY DIFFERENTIAL EQUATION MODEL USING DIFFERENTIAL TRANSFORMATION

This chapter proposes a novel approach for power system dynamic simulation based on the DT method. The DT is introduced to study power systems as high-dimensional nonlinear dynamical systems for the first time and is able to avoid computations of high-order derivatives with nonlinear differential equations by its transform rules. This chapter first proposes and proves several new transform rules for generic non-linear functions that often appear in power system models, and then uses these rules to transform representative power system models such as the synchronous machine model with trigonometric functions and the exciter model with exponential and square root functions. This chapter also designs a DT-based simulation scheme that allows significantly prolonged time steps to reduce simulation time compared to a traditional numerical approach. The numerical stability, accuracy and time performance of the proposed new DT-based simulation approach are compared with widely used numerical methods on the IEEE 39-bus system and Polish 2383-bus system.

## 2.1 Introduction of the Differential Transformation

For a set of nonlinear differential equations (DEs) that models a dynamical system, the explicit, analytical solution does not exist in general. However, an approximate solution can be represented by a finite power series in time up to a designed order. Thus, calculating power series coefficients would be a key to obtain such an approximate solution.

The DT is an effective mathematical tool to serve this purpose. It has been studied by researchers in the fields of applied mathematics and physics for various nonlinear dynamic systems such as the Van der Pol oscillator, Duffing equations and fractional order systems [49]-[57]. In existing literature, the DT is mainly applied to small systems described by low-order DEs and its capability has not been examined for real-life complex network systems like power systems modeled by a large set of nonlinear DEs. This section introduces the definition and rules of the DT and the next section will apply the DT to power system models.

**Definition 1:** Consider a function $x(t)$ of a real continuous variable *t*. The DT of $x(t)$ is defined by (2-1), and the inverse DT of $X(k)$ is defined by (2-2), where $k$ Î ¥ is the order.

$$X(k) = \frac{1}{k!}\left[\frac{d^k x(t)}{dt^k}\right]_{t=0} \tag{2-1}$$

$$x(t) = \sum_{k=0}^{\infty} X(k)\, t^k \tag{2-2}$$

The DT method provides a set of transform rules such as those in the following Proposition 1 and Proposition 2. Their detailed proofs can be found in [49].

**Proposition 1:** Denote $x(t)$, $y(t)$ and $z(t)$ as the original functions and $X(k)$, $Y(k)$ and $Z(k)$ as their DTs, respectively. The following propositions (a) - (f) hold, where $c$ is a constant, $n$ is a nonnegative integer and ñ is the Kronecker delta function defined in discrete domain.

(a) $X(0) = x(0)$.

(b) $y(t) = cx(t) \rightarrow Y(k) = cX(k)$.

(c) $z(t) = x(t) \pm y(t) \rightarrow Z(k) = X(k) \pm Y(k)$.

(d) $z(t) = x(t)y(t) \rightarrow Z(k) = \sum_{m=0}^{k} X(m)Y(k-m)$.

(e) $y(t) = t^n \rightarrow Y(k) = \delta(k-n) = \begin{cases} 1, & k = n \\ 0, & k \neq n \end{cases}$.

(f) $y(t) = c \rightarrow Y(k) = c\delta(k) = \begin{cases} c, & k = 0 \\ 0, & k \neq 0 \end{cases}$

(g) $y(t) = \dfrac{dx(t)}{dt} \rightarrow Y(k) = (k+1)X(k+1)$.

**Proposition 2**: If $f(t) = \sin d(t)$, $y(t) = \cos d(t)$ and $\mathrm{F}(k)$, $\mathrm{Y}(k)$ and $\mathrm{D}(k)$ are the DTs of $f(t)$, $y(t)$ and $d(t)$, respectively, then $\mathrm{F}(k)$ and $\mathrm{Y}(k)$ are calculated by:

$$\begin{aligned} \mathrm{F}(k) &= \sum_{m=0}^{k-1} \frac{k-m}{k} \mathrm{Y}(m)\mathrm{D}(k-m) \\ \mathrm{Y}(k) &= -\sum_{m=0}^{k-1} \frac{k-m}{k} \mathrm{F}(m)\mathrm{D}(k-m) \end{aligned} \tag{2-3}$$

## 2.2 DTs of Power System Models

The Taylor series of a nonlinear function can more easily be obtained by using the DT with the above transform rules than directly using the traditional Taylor expansion formulas. This section presents the DTs of several representative power system models,

which include the detailed 6th order synchronous machine model, the first order governor and first order turbine model, the IEEE Type I exciter model and the first order PSS (power system stabilizer) model.

### 2.2.1 New Transform Rules for Power System Models

The main idea of deriving the DT of a power system model is to apply DT to both sides of each equation. However, besides the nonlinear functions involved in Propositions 1 and 2, there are other commonly used nonlinear functions in power system models whose transform rules are not provided by the original DT theory, such as composite exponential functions, square root functions and fractional functions. This section has further proved the transform rules of these functions according to the basic idea in [49] as follows. These additional transform rules together with the basic transform rules can be implemented by a software library in a symbolic computing environment, e.g. Mathematica and Maple. Thus, the whole procedure of DT for a power system model can become automated.

**Proposition 3:** Given function $y_1(t) = e^{x(t)}$, if $Y_1(k)$ and $X(k)$ are the DTs of $y_1(t)$ and $x(t)$, respectively, then $Y_1(k)$ is calculated by:

$$Y_1(k) = \frac{1}{k}\sum_{m=0}^{k-1}(k-m)Y_1(m)X(k-m) \tag{2-4}$$

**Proof of Proposition 3:** From $\dot{y}_1 = y_1\dot{x}$, there is

$$(k+1)Y_1(k+1) = \sum_{m=0}^{k} Y_1(m) \times (k+1-m) X(k+1-m)$$

For convenience, it can be written as:

$$Y_1(k+1) = \frac{1}{(k+1)} \sum_{m=0}^{k} (k+1-m) Y_1(m) X(k+1-m)$$

Replacing *k* by *k-1* will lead to (2-4). ■

**Proposition 4:** Given function $y_2(t) = \sqrt{x}$, $Y_2(k)$ and $X(k)$ are the DTs of $y_2(t)$ and $x(t)$, respectively, and

$$Y_2(k) = \frac{1}{2Y_2(0)} X(k) - \frac{1}{2Y_2(0)} \sum_{m=1}^{k-1} Y_2(m) Y_2(k-m) \tag{2-5}$$

**Proof of Proposition 4:** From $y_2^2 = x$, there is

$$\sum_{m=0}^{k} Y_2(m) Y_2(k-m) = X(k)$$

$$2Y_2(0) Y_2(k) + \sum_{m=1}^{k-1} Y_2(m) Y_2(k-m) = X(k)$$

Then, it is easy to obtain (2-5). ■

**Proposition 5:** Given function $z(t) = x(t)/y(t)$, if $X(k)$, $Y(k)$ and $Z(k)$ are the DTs of $x(t)$, $y(t)$ and $z(t)$, respectively, then $Z(k)$ is calculated by:

$$Z(k) = \frac{1}{Y(0)} X(k) - \frac{1}{Y(0)} \sum_{m=0}^{k-1} Z(m) Y(k-m) \tag{2-6}$$

**Proof of Proposition 5:** Observe that $x = zy$, we have

$$\sum_{m=0}^{k} Z(m)Y(k-m) = X(k)$$

$$Y(0)Z(k) + \sum_{m=0}^{k-1} Z(m)Y(k-m) = X(k)$$

Then, it is easy to obtain (2-6). ■

To be differentiated from original functions, e.g. $e_{fd}(t), v_r(t)$, their DTs are denoted using capital letters, e.g., $E_{fd}(k)$ and $V_r(k), k = 0, 1, 2 \ldots K$. Besides, time "$t$" of original functions are omitted for simplicity, i.e. $e_{fd}(t), v_r(t) \rightarrow e_{fd}, v_r$.

**2.2.2 Governor Model**

The equation of a 1st order governor model is given in (2-7), where $p_{sv}$ is the governor output power, $p_{ref}$ is the electrical power setting point and $s_m$ is rotor slip; $T_{sv}$ and $R_d$ are governor time constant and the droop coefficient.

$$T_{sv}\dot{p}_{sv} = -p_{sv} + p_{ref} - \frac{1}{R_d}s_m \tag{2-7}$$

The DTs of the four terms in (2-7) can be obtained using Proposition 1 as explained in detail in the following:

1) The LHS (left hand side) term $T_{sv}\dot{p}_{sv}$ is a product of a constant and the derivative of $p_{sv}$. The DT of $\dot{p}_{sv}$ can be obtained from Proposition 1-g:

$$\dot{p}_{sv} \rightarrow (k+1)P_{sv}(k+1)$$

Then the DT of $T_{sv}\dot{p}_{sv}$ can be obtained from Proposition 1-b:

$$T_{sv}\dot{p}_{sv} \rightarrow (k+1)T_{sv}P_{sv}(k+1)$$

2) Similarly, the DTs of three RHS (right hand side) terms can be obtained using Proposition 1-b:

$$-p_{sv} \;\text{®}\; -P_{sv}(k)$$
$$p_{ref} \;\text{®}\; p_{ref}\tilde{n}(k)$$
$$-\frac{1}{R_d}s_m \;\text{®}\; -\frac{1}{R_d}S_m(k)$$

Finally, the original equation becomes

$$(k+1)T_{sv}P_{sv}(k+1) = -P_{sv}(k) + p_{ref}\tilde{n}(k) - \frac{1}{R_d}S_m(k) \tag{2-8}$$

**2.2.3 Turbine Model**

Consider the 1st order turbine model given in (2-9) about mechanical power $p_m$ with time constant $T_{ch}$:

$$T_{ch}\dot{p}_m = -p_m + p_{sv} \tag{2-9}$$

Similar to the above procedure for the governor model, apply DT to its both sides:

$$(k+1)T_{ch}P_m(k+1) = -P_m(k) + P_{sv}(k) \tag{2-10}$$

**2.2.4 PSS Type I Model**

A PSS Type I model is given in (2-11), where $v_s$ is the PSS output voltage that is used to modify the exciter reference voltage; the input signal is the rotor speed $w$, electrical power $p_e$ and bus voltage magnitude $v_t$; $T_w$ is the time constant and $K_w$ is the stabilizer gain.

$$\begin{aligned} T_w\dot{v}_1 &= -(K_w w + K_p p_e + K_v v_t + v_1) \\ v_s &= K_w w + K_p p_e + K_v v_t + v_1 \end{aligned} \tag{2-11}$$

Similar with the procedure for governor and turbine model, the DT on the PSS model can be written in (2-12).

$$
\begin{aligned}
&(k+1)T_w V_1(k+1) = -\left(K_w W(k) + K_p P_e(k) + K_v V_t(k) + V_1(k)\right) \\
&V_s(k) = K_w W(k) + K_p P_e(k) + K_v V_t(k) + V_1(k)
\end{aligned}
\tag{2-12}
$$

### 2.2.5 Synchronous Machine Model

A detailed 6th order synchronous machine model is given in (2-13)-(2-16) including a coordinate transform at the terminal bus for the network interface under the constant impedance load assumption.

$$
\begin{aligned}
T'_{q0}\dot{e}'_d &= -\frac{x_q - x''_q}{x'_q - x''_q} e'_d + \frac{x_q - x'_q}{x'_q - x''_q} e''_d \\
T'_{d0}\dot{e}'_q &= -\frac{x_d - x''_d}{x'_d - x''_d} e'_q + \frac{x_d - x'_d}{x'_d - x''_d} e''_q + e_{fd} \\
T''_{q0}\dot{e}''_d &= e'_d - e''_d + \left(x'_q - x''_q\right) i_q \\
T''_{d0}\dot{e}''_q &= e'_q - e''_q - \left(x'_d - x''_d\right) i_d \\
\dot{d} &= w_s s_m \\
2H\dot{s}_m &= p_m - p_e - D s_m
\end{aligned}
\tag{2-13}
$$

$$
p_e = v_d i_d + v_q i_q
$$

$$
\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} r_a & -x''_q \\ x''_d & r_a \end{bmatrix}^{-1} \left( \begin{bmatrix} e''_d \\ e''_q \end{bmatrix} - \begin{bmatrix} v_d \\ v_q \end{bmatrix} \right)
\tag{2-14}
$$

$$
i = \boldsymbol{Y}_r v
\tag{2-15}
$$

$$
\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \quad \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} v_d \\ v_q \end{bmatrix} \quad \text{where } \boldsymbol{R} = \begin{bmatrix} \sin d & \cos d \\ -\cos d & \sin d \end{bmatrix}
\tag{2-16}
$$

State variables $\delta$, $\omega$, $e'_q$, $e'_d$, $e''_q$ and $e''_d$ are respectively the rotor angle, rotor speed, $q$-axis and $d$-axis transient voltages and sub-transient voltages; $p_e$, $i_d$ and $i_q$ are the electrical power and $d$-axis and $q$-axis stator currents; $v_d$ and $v_q$ are the $d$-axis and $q$-axis terminal voltages; $i_x$ and $i_y$ are the $x$-axis and $y$-axis terminal currents; $v_x$ and $v_y$ are the $x$-axis and $y$-axis terminal voltages respectively; $p_m$ is the mechanical power; $e_{fd}$ is field voltage; $H$ is the inertia and $D$ is the damping constant; $T'_{d0}$, $T'_{q0}$, $T''_{d0}$ and $T''_{q0}$ are the

open circuit transient time constants and sub-transient time constants in *d*-axis and *q*-axis; $x_d$, $x_q$, $x'_d$, $x'_q$, $x''_d$ and $x''_q$ are the *d*-axis and *q*-axis synchronous reactances, transient reactances and transient reactances; $w_s = 2p \times 60$ is the nominal frequency and $\boldsymbol{Y_r}$ is the reduced network admittance matrix.

The DTs of (2-13)-(2-16) are given in (2-17)-(2-20), where $\mathrm{F}(m)$ and $\mathrm{Y}(m)$ denote the DTs of $\sin\delta$ and $\cos\delta$ obtained by Propositions 2.

$$
\begin{aligned}
(k+1)T'_{q0}E'_d(k+1) &= -\frac{x_q - x''_q}{x'_q - x''_q}E'_d(k) + \frac{x_q - x'_q}{x'_q - x''_q}E''_d(k) \\
(k+1)T'_{d0}E'_q(k+1) &= -\frac{x_d - x''_d}{x'_d - x''_d}E'_d(k) + \frac{x_d - x'_d}{x'_d - x''_d}E''_q(k) + E_{fd}(k) \\
(k+1)T''_{q0}E''_d(k+1) &= E'_d(k) - E''_d(k) + (x'_q - x''_q)I_q(k) \\
(k+1)T''_{d0}E''_q(k+1) &= E'_q(k) - E''_q(k) - (x'_d - x''_d)I_d(k) \\
(k+1)\mathrm{D}(k+1) &= w_s S_m(k) \\
2(k+1)HS_m(k+1) &= P_m(k) - P_e(k) - DS_m(k)
\end{aligned}
\tag{2-17}
$$

$$
\begin{bmatrix} I_d(k) \\ I_q(k) \end{bmatrix} = \begin{bmatrix} r_a & -x''_q \\ x''_d & r_a \end{bmatrix}^{-1} \left( \begin{bmatrix} E''_d(k) \\ E''_q(k) \end{bmatrix} - \begin{bmatrix} V_d(k) \\ V_q(k) \end{bmatrix} \right)
$$

$$
P_e(k) = \sum_{m=0}^{k} \left[ V_d(m) I_d(k-m) + V_q(m) I_q(k-m) \right]
\tag{2-18}
$$

$$
I(k) = Y_r V(k)
\tag{2-19}
$$

$$
\begin{aligned}
\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} &= \sum_{m=0}^{k} \begin{bmatrix} \mathrm{F}(m) & \mathrm{Y}(m) \\ -\mathrm{Y}(m) & \mathrm{F}(m) \end{bmatrix} \begin{bmatrix} I_d(m-k) \\ I_q(m-k) \end{bmatrix} \\
\begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} &= \sum_{m=0}^{k} \begin{bmatrix} \mathrm{F}(m) & \mathrm{Y}(m) \\ -\mathrm{Y}(m) & \mathrm{F}(m) \end{bmatrix} \begin{bmatrix} V_d(m-k) \\ V_q(m-k) \end{bmatrix}
\end{aligned}
\tag{2-20}
$$

### 2.2.6 IEEE Type I Exciter Model

The IEEE Type I exciter model is given in (2-21)-(2-23), where $e_{fd}, v_f, v_{ts}, v_r$ are the field voltage, feedback voltage, sensed terminal voltage and regulator voltage;

$T_e, T_f, T_r, T_a$ are exciter time constant, feedback time constant, filter time constant and regulator time constant; $K_e, K_f, K_a$ are exciter constants related to self-excited field, feedback gain and regulator gain; $s_e$ is the exciter saturation function which is a nonlinear exponential function determined by the two constants $a_e$ and $b_e$; $v_t$ is the bus voltage which is a nonlinear square root function; $v_{rmax}, v_{rmin}$ are the maximum and minimum voltage regulator outputs.

$$
\begin{aligned}
T_e \dot{e}_{fd} &= v_r - s_e e_{fd} - K_e e_{fd} \\
T_f \dot{v}_f &= -v_f + K_f \dot{e}_{fd} \\
T_r \dot{v}_{ts} &= -v_{ts} + v_t \\
T_a \dot{v}_r &= \begin{cases} -v_r + K_a\left(v_{ref} + v_s - v_{ts} - v_f\right), \\ \qquad if\ v_{rmin} < v_r < v_{rmax} \\ 0, if\left(v_r = v_{rmax}, \dot{v}_r > 0\right), OR \\ \qquad \left(v_r = v_{rmin}, \dot{v}_r < 0\right) \end{cases}
\end{aligned}
\tag{2-21}
$$

$$s_e = a_e e^{b_e e_{fd}} \tag{2-22}$$

$$v_t = \sqrt{v_x^2 + v_y^2} \tag{2-23}$$

The DTs of (2-21) is given in (2-24). Since the DT can handle the discrete events in (2-21) by deriving the DT for both expressions as shown in the last two equations in (23), they can be switched in the simulation as the traditional method does.

$$
\begin{aligned}
(k+1)T_e E_{fd}(k+1) &= V_r(k) - K_e E_{fd}(k) - \sum_{m=0}^{k} S_e(m) E_{fd}(k-m) \\
(k+1)T_f V_f(k+1) &= -V_f(k) + (k+1)K_f E_{fd}(k+1) \\
(k+1)T_r V_{ts}(k+1) &= -V_{ts}(k) + V_t(k) \\
(k+1)T_a V_r(k+1) &= -V_r(k+1) \\
&\quad + K_a\left(V_{ref} d(k) + V_s(k) - V_{ts}(k) - V_f(k)\right) \\
(k+1)T_a V_r(k+1) &= 0
\end{aligned}
\tag{2-24}
$$

However, both (2-22) and (2-23) are composite nonlinear functions without existing transform rules. To obtain their DTs, Propositions 3 and 4 proved previously for the composite exponential function and composite square root function are applied. From proposition 3, the DT of the saturation function is given in (2-25).

$$S_e(k) = a_e \frac{1}{k} \sum_{m=0}^{k-1} (k-m) S_e(m) E_{fd}(k-m) \tag{2-25}$$

From proposition 4, the DT of terminal voltage is given in (2-26)-(2-27), where $u = v_t^2 = v_x^2 + v_y^2$, and $U(k)$ is its DT.

$$U(k+1) = \sum_{m=0}^{k} V_X(m) V_X(k-m) + \sum_{m=0}^{k} V_Y(m) V_Y(k-m) \tag{2-26}$$

$$V_t(k+1) = \frac{1}{2U(0)} V_t(k) - \frac{1}{2U(0)} \sum_{m=1}^{k-1} U(m) U(k-m) \tag{2-27}$$

## 2.3 Proposed Scheme for Dynamic Simulation

This section proposes a DT-based scheme for dynamic simulation using the DTs derived above for power system models and then illustrates the scheme using a single-machine-infinite-bus (SMIB) system.

### 2.3.1 DT-based Solution Scheme

The variables of power system model in this section are given in (2-28), which satisfy (2-7), (2-9), (2-11), (2-13)-(2-16) and (2-21)-(2-23).

$$\begin{aligned} x(t) &= \left[ p_{sv}, p_m, v_s, e_d', e_q', e_d'', e_q'', \delta, s_m, e_{fd}, v_f, v_r, v_{ts} \right] \\ y(t) &= \left[ i_d, i_q, p_e, i_x, i_y, v_d, v_q, v_x, v_y, s_e, v_t \right] \end{aligned} \tag{2-28}$$

In the proposed dynamic simulation scheme, these variables are approximated by the $K^{\text{th}}$ order power series. After obtaining the coefficients, the trajectory can be

displayed by evaluating (2-29). Thus, the key is to calculate the coefficients $X(k), Y(k), k = 0,1,2\ldots K$ defined by (2-30).

$$\begin{aligned} x(t) &= X(0) + X(1)t + X(2)t^2 + \ldots X(K)t^K \\ y(t) &= Y(0) + Y(1)t + Y(2)t^2 + \ldots Y(K)t^K \end{aligned} \tag{2-29}$$

$$\begin{aligned} X(k) &= \left\{P_{sv}, P_m, V_s, E_d^{'}, E_q^{'}, E_d^{''}, E_q^{''}, \Delta, S_m, E_{fd}, V_f, V_r, V_{ts}\right\} \\ Y(k) &= \left\{I_d, I_q, P_e, I_x, I_y, V_d, V_q, V_x, V_y, S_e, V_t\right\} \end{aligned} \tag{2-30}$$

At each simulation step, $\{X(0), Y(0)\}$ are known and $\{X(k), Y(k), k = 1,2,\ldots K\}$ are to be solved. Algorithm 1 is designed to calculate the $(k+1)^{th}$ order coefficients from the coefficients of orders $0,1,\ldots k$. The algorithm is executed $K$-1 times to obtain all the coefficients starting from $\{X(0), Y(0)\}$. Since the operations of DT are purely linear, the calculation process is explicit. For example, the coefficients $P_{sv}(k+1)$ explicitly depend on the lower order coefficients as shown in (2-8).

The DT-based simulation scheme is illustrated below using an SMIB system given in (2-31) with parameters $H$=3 s, $D$=3 p. u., $w_s = 2p \times 60$ rad/s, $P_{max}$=1.7 p.u., $P_m$=0.44 p.u. and the initial state $d(0) = 0.26$ rad and $w(0) = 0.002$ p.u.

$$\begin{cases} \dot{d} = w_s w \\ \dot{w} = \dfrac{1}{2H}\left(P_m - P_{\max}\sin d - Dw\right) \end{cases} \tag{2-31}$$

To obtain the trajectories of rotor angle and rotor speed, the first step is to apply DT to (31). Similar to the steps described in the previous section, its DT is given in (2-32)-(2-33):

**Algorithm 1**

**Input**: $X(0:k), Y(0:k)$

**Output**: $X(0:k+1), Y(0:k+1)$

1. Calculate $P_{sv}(k+1)$ from (2-8).
2. Calculate $P_m(k+1)$ from (2-10).
3. Calculate $E_d^{'}(k+1), E_q^{'}(k+1), E_d^{''}(k+1), E_q^{''}(k+1), \Delta(k+1), S_m(k+1)$ from (2-17).
4. Calculate $I_d(k), I_q(k), P_e(k)$ from (2-18).
5. Calculate $V_x(k), V_y(k)$ from (2-19).
6. Calculate $\Phi(k), \Upsilon(k)$ from (2-3).
7. Calculate $V_d(k), V_q(k), I_x(k), I_y(k)$ from (2-20).
8. Calculate $S_e(k), V_t(k)$ from (2-25)-(2-27).
9. Calculate $E_{fd}(k+1), V_r(k+1), V_f(k+1), V_{ts}(k+1)$ from (2-24).
10. Calculate $V_s(k+1)$ from (2-12).

$$(k+1)\Delta(k+1) = w_s W(k) \tag{2-32}$$

$$2H(k+1)W(k+1) = P_m \tilde{n}(k) - P_{\max}\Phi(k) - DW(k) \tag{2-33}$$

With the values of parameters plugged in, the recursive formula to calculate the coefficients becomes

$$\begin{aligned} \Delta(k+1) &= \frac{377}{k+1} W(k) \\ W(k+1) &= \frac{1}{k+1}\left[0.073\tilde{n}(k) - 0.283\Phi(k) - 0.5W(k)\right] \end{aligned} \tag{2-34}$$

The coefficients $\Delta(0), W(0), \Phi(0), \Upsilon(0)$, are obtained in (2-35) using Proposition 1-a.

$$
\begin{aligned}
\mathrm{D}(0) &= d(0) = 0.26 \\
W(0) &= w(0) = 2 \times 10^{-3} \\
\mathrm{F}(0) &= \sin(d(0)) = 0.2571 \\
\mathrm{Y}(0) &= \cos(d(0)) = 0.9664
\end{aligned}
\tag{2-35}
$$

The values of $\mathrm{D}(k), W(k)$ for arbitrary $k$ can be calculated starting from (2-35). For example, the coefficients with $k$=1 are calculated in (2-36) with $\tilde{n}(0) = 1$.

$$
\begin{aligned}
\mathrm{D}(1) &= 377W(0) = 0.754 \\
W(1) &= 0.073 - 0.283\mathrm{F}(0) - 0.5W(0) = -7.6 \times 10^{-4} \\
\mathrm{F}(1) &= \mathrm{Y}(0)\mathrm{D}(1) = 0.7287 \\
\mathrm{Y}(1) &= -\mathrm{F}(0)\mathrm{D}(1) = -0.1938
\end{aligned}
\tag{2-36}
$$

Such a recursive process continues until $k$ reaches a desired order $K$. The rotor angle and rotor speed expressions for $K$=3 are approximated by

$$
\begin{aligned}
\delta(t) &= \Delta(0) + \Delta(1)t + \Delta(2)t^2 + \Delta(3)t^3 \\
w(t) &= W(0) + W(1)t + W(2)t^2 + W(3)t^3
\end{aligned}
\tag{2-37}
$$

**2.3.2 Remarks**

In recent literature, other methods have been applied to obtain an approximate solution of a nonlinear differential equation in the form of a polynomial function of time, such as the Adomian decomposition method (ADM) [32]-[33] and a Power Series based method (for short, PSM) [35][36]. The approximate solutions of these two methods and the proposed DT method all converge to the true solution in the power series form when the number of terms increases to infinity [32]-[36], [50], [58]-[59]. However, when finite terms are taken, their solutions are not identical term by term. The main reason lies in their different approximation and truncation mechanisms in generating each specific term: the ADM generates each term as a polynomial of time [32][33] composed of monomials of different orders, so the later terms can also change the coefficients of some

of low order monomials when terms grow; the other two methods generate each term as a single monomial of time; the PSM needs to transform all analytical nonlinear functions, e.g. sine and cosine functions, first into truncated Taylor series so as to introduce truncation errors, while the DT method directly uses the differential transforms of nonlinear functions instead of their Taylor expansions.

The solutions of all the above methods are accurate within a limited time window, whose length in general increases with the number of terms. However, their different generation mechanisms make the DT be easier to implement in practice with a large number of terms than the ADM and PSM. The ADM and PSM have to explicitly calculate high order derivatives to obtain Adomian polynomials or truncated Taylor series [32]-[36], [58]-[59]. The complexity of high order derivatives increases significantly at a large number of terms. Comparatively, the DT adopts transform rules to avoid computing high order derivatives. Ref. [37] proposes a Pade Approximants based method to transform a polynomial-form solution, which can be from any of these three methods, into a fractional form to further improve its accuracy, and hence can be applied as a post-processing technique together with each of these three methods.

## 2.4 Case Study

In this section, the proposed simulation scheme is tested on the IEEE 10-machine 39-bus system and Polish 327-machine 2383-bus system [60]. The accuracy of the DT approach is validated by various disturbances. The numerical stability, accuracy and time performance are compared with five commonly used numerical methods [7][11]: the modified Euler method (ME), 4th order Runge-Kutta method (RK4), Gill’s version of

Runge-Kutta method (RKG), Trapezoidal method and Gear method. In all subsections, the benchmark result is given by the RK4 method with a very small time step of 0.3ms. Errors of each method are considered its differences from the benchmark result. Simulations are performed in MATLAB R2017a on a computer with i5-7200U CPU.

### 2.4.1 Scanning Various Contingencies

For the 39-bus system, three stability scenarios, i.e., one stable case, one marginally stable case, and one unstable case, are simulated for 20 seconds. The stable case has a three-phase fault at the bus 3 that starts at $t$=1s and is cleared after 5 cycles by tripping the line 3-4. The marginal stable and unstable cases are created by clearing the fault after 12.935 cycles and 12.940 cycles, respectively. Fig. 1 to Fig. 3 show the trajectories of rotor angles from both the DT and benchmark RK4 methods as well as the rotor angle errors of the DT method. The machine 1 at bus 30 is selected as the reference. The results of the DT method with $K$=8 and the time step length of 0.05s accurately match the benchmark results for all three scenarios. The maximum errors of all state variables (including rotor angles, rotor speeds, transient and sub-transient voltages in $d$-axis and $q$-axis, field voltages, and all other variables) over the entire 20-second simulation period are $1.48\times10^{-4}$ p.u., $1.00\times10^{-3}$ p.u. and $1.30\times10^{-3}$ p.u. for the three scenarios, respectively.

The same tests are also conducted on the Polish 2383-bus system and the DT method can still assess stability accurately. Fig. 4 to Fig. 6 show the trajectories of rotor angles from both the DT and RK4 methods as well as rotor angle errors of the DT method using the time step length of 0.016s. The machine 1 at bus 10 is selected as the

reference. The maximum errors of all state variables over the entire 20-second simulation period are $5.2\times10^{-3}$ p.u., $3.9\times10^{-3}$ p.u. and $3.8\times10^{-3}$ p.u. for the three scenarios, respectively.

In addition, the impacts of various factors on the simulation of the DT method are studied, including: fault types; fault locations; the modeling of PSS; and the modeling of saliency. Apply these two types of faults to all buses and lines of the 39-bus system: 1) temporary three-phase short-circuit fault lasting for 5 cycles on each bus (totally, 39 faults); 2) permanent three-phase short-circuit line fault near each end of every line that is tripped by opening the line after 5 cycles (totally, 46×2=92 faults). Fig. 7 and Fig. 8 show that the maximum errors are within $1.1\times10^{-3}$ p.u. for any state variables of the system in all cases. Besides, a 830 MW generator trip on bus 38 is simulated and the maximum error is $2.26\times10^{-4}$ p.u..

Also, the impact of the PSS is simulated by many cases. Fig. 9 shows the maximum errors are within $3.0\times10^{-4}$ p.u. and are not affected much by the modeling of PSS. Meanwhile, the impact of the saliency is studied. Fig. 10 shows the maximum errors are within $2.6\times10^{-4}$ p.u. and the modeling of saliency does not introduce additional errors.

### 2.4.2 Comparison of Numerical Stability

To study the error propagation and numerical stability, three scenarios are designed for the 39-bus system with three different time step lengths: 0.050s, 0.067s and 0.100s. The simulated disturbance is a three-phase fault at bus 3 cleared after 5 cycles by tripping the branch 3-4. Fig. 11 a)-c) shows the error propagation with the three scenarios where all methods start from the same initial state at $t$=0s. To take a detailed look at the

transient dynamics right after the fault, the figure focuses on the first 1 second. The logarithmic vertical axis is the maximum error of all state variables at the end of each time step. The errors do not propagate much for the four methods when the time step length is 0.050s. Meanwhile, Fig. 11 b) shows the error of the ME method propagates when the time step length is increased to 0.067s and it approaches $10^2$ p.u. at $t$=1.0s, indicating divergence of the ME method. In Fig. 11 c), it shows that the errors of ME, RK4 and RKG methods increase along time steps significantly, indicating the tendency toward numerical instability. Table 1 summarizes whether the four methods tend to be numerically stable or unstable under three scenarios. All methods work well when the time step length is 0.050s. The ME method has numerical instability issue when the time step length is 0.067s. Only the DT method is numerically stable under all three scenarios.

Table 2 summarizes the maximum time step lengths of the four methods to maintain the numerical stability for both the 39-bus system and the 2383-bus system, by extensive simulations with gradually increased time step lengths until numerical instability occurs. The DT approach can maintain numerical stability using much larger time step lengths than the ME, RK4 and RKG methods. But it will diverge when the time step length is larger than 0.125 s for the 39-bus system and 0.017 s for the 2383-bus system. By contrast, the implicit methods such as the Trapezoidal method and the Gear method are very stable, and there is no need to limit their time step lengths from the numerical stability perspective. These results indicate the numerical stability of the DT approach is better than the ME, RK4, and RKG methods, but weaker than the Trapezoidal method and the Gear method.

### 2.4.3 Comparison of Accuracy and Time Performance

For both test systems, Table 3 shows the time step lengths of the ME, RK4, RKG and DT methods with three desired error tolerances, i.e., the maximum error of all state variables over the entire simulation period is in the magnitude of $10^{-2}$, $10^{-3}$ and $10^{-4}$ p.u., respectively. For the Trapezoidal and Gear methods, the average time step lengths are shown since they are implemented by MATLAB solvers ode23t and ode15s respectively with variable time step lengths.

For the 39-bus system, the time step lengths of the DT can be increased compared with the ME method to 7.5 times, 23.3 times and 50.0 times while still meeting the requirements of three error tolerances, respectively. Compared with both the RK4 and RKG methods, it is increased to 2.0 times, 2.7 times and 4.5 times. Also, the DT method increases the time step length to 4.5 times and 1.7 times compared to the Trapezoidal and Gear method respectively under the same error tolerance. Similar results are also observed on the 2383-bus system. These results show the DT approach can prolong the time step lengths with the other methods for the same level of error tolerance.

For both test systems, Table 4 shows the computation times of the six methods using the time step lengths in Table 3. For the 39-bus system, the time costs of the DT method are reduced by 33.3%, 66.4% and 86.8% compared to the ME method under three error tolerances respectively. It is reduced by 15.1%, 17.8%, and 43.2% compared with both the RK4 and RKG methods. Also, the DT reduces the time cost by 61.1% and 11.9% compared to the Trapezoidal and Gear methods respectively under the same error tolerances. Similar results are also observed on the 2383-bus system. These results show

the DT approach can reduce computation time compared with the other methods for the same level of error tolerance.

## 2.5 Conclusion

This chapter has proposed a DT-based approach for dynamic simulation of multi-machine power systems. This chapter derived the DTs for detailed power system models. Then, the simulation scheme using the derived DTs was tested on two test systems. The results show that the approach significantly increases the time step length by using more power series terms to approximate the solution so as to speed up simulation while keeping a comparable accuracy with traditional numerical integration. Therefore, the approach is of great potential for fast power system simulation.

Electricity utilities usually model a power system by a set of nonlinear differential algebraic equations (DAEs) for transient stability studies. In this chapter, impedance loads for each simulated system are assumed so as to reduce the original DAE model into a DE model by including loads into a reduced admittance matrix. The simulation results from such a DE model are different from simulation results of a DAE model if more complex loads exist. This chapter has focused on a proof of concept study on the DT method for simulating a large-scale system and several important aspects of handling DEs for power system simulation including transformation rules specifically for nonlinear power system models, practical DE models of generators and their controllers and detailed comparisons of the DT method with other numerical methods. Extending the proposed approach to solve power system DAEs will be the focus of next chapter.

# CHAPTER 3

# SOLVING DIFFERENTIAL ALGEBRAIC EQUATION MODEL USING DIFFERENTIAL TRANSFORMATION

This chapter proposes a novel non-iterative method to solve power system differential algebraic equations (DAEs) using the DT method, which has proved to be effective in solving state variables of nonlinear differential equations in previous chapter. This chapter further solves non-state variables, e.g. current injections and bus voltages, directly with a realistic DAE model of power grids. These non-state variables, nonlinearly coupled in network equations, are conventionally solved by numerical methods with time-consuming iterations, but their DTs are proved to satisfy formally linear equations in this chapter. Thus, a non-iterative algorithm is designed to analytically solve all variables of a power system DAE model with ZIP loads. From test results on a Polish 2383-bus system, the proposed method demonstrates fast and reliable time performance compared to traditional numerical approaches including the implicit trapezoidal rule method and a partitioned scheme using the explicit modified Euler method and Newton Raphson method.

## 3.1 Proposed Method for Solving Power System DAE Model Using DT

### 3.1.1 Conceptual Description of the Proposed Method

A power system DAE model in the state-space representation is given in (3-1), where $\boldsymbol{x}$ is the state vector, $\boldsymbol{v}$ is the vector of bus voltages, $\boldsymbol{f}$ represents a vector field determined by differential equations on dynamic devices such as synchronous generators

and associated controllers, $\boldsymbol{i}$ is the vector-valued function on current injections from all generators and load buses, , and $\boldsymbol{Y_{bus}}$ is the network admittance matrix.

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{v}) \\ \boldsymbol{Y}_{bus}\boldsymbol{v} &= \boldsymbol{i}(\boldsymbol{x}, \boldsymbol{v}) \end{aligned} \tag{3-1}$$

In the proposed method, the solution of both state variables and the non-state variables, bus voltages, are approximated by $K^{th}$ order power series in time in (3-2). The major task is to solve power series coefficients of orders from 0 to $K$. The two steps to obtain these coefficients are conceptually shown below and then elaborated in Chapter 3.2 and 3.3, respectively.

$$\begin{aligned} \boldsymbol{x} &= \sum_{0}^{K} \boldsymbol{X}(k) t^k \\ \boldsymbol{v} &= \sum_{0}^{K} \boldsymbol{V}(k) t^k \end{aligned} \tag{3-2}$$

### 3.1.2 Step 1: Deriving DTs of Power System DAE Model

The DTs of the DAE model (3-1) will be derived in Chapter 3.2 and have the general form in (3-3). Compared with the original DAE model (3-1), each variable or function $\boldsymbol{x}$, $\boldsymbol{v}$, $\boldsymbol{f}$, $\boldsymbol{i}$ are transformed to their power series coefficients $\boldsymbol{X}(k)$, $\boldsymbol{V}(k)$, $\boldsymbol{F}(k)$, $\boldsymbol{I}(k)$ (denoted by their corresponding capital letters), coupled by a new set of equations in (3-3). It can be observed that, the left-hand side (LHS) of (3-3) only contains the $(k+1)^{th}$ order coefficients of state variables and $k^{th}$ order coefficients of bus voltages, respectively, while the right-hand side (RHS) couples $0^{th}$ to $k^{th}$ order coefficients of both state variables and bus voltages by nonlinear functions $\boldsymbol{F}$ and $\boldsymbol{I}$.

$$\begin{aligned} (k+1)\boldsymbol{X}(k+1) &= \boldsymbol{F}(k) = \boldsymbol{F}(\boldsymbol{X}(l), \boldsymbol{V}(l)), l = 0 \ldots k \quad \text{(a)} \\ \boldsymbol{Y}_{bus}\boldsymbol{V}(k) &= \boldsymbol{I}(k) = \boldsymbol{I}(\boldsymbol{X}(l), \boldsymbol{V}(l)), l = 0 \ldots k \quad \text{(b)} \end{aligned} \tag{3-3}$$

### 3.1.3 Step 2: Solving Power Series Coefficients of State Variables and Bus Voltages

The main task in this step is to solve power series coefficients $\boldsymbol{X}(k)$ and $\boldsymbol{V}(k)$ ($k \geq 1$) from the $(k\text{-}1)^{\text{th}}$ order coefficients, as indicated by two circled numbers in Fig. 12. Thus, any order coefficients are solvable from $\boldsymbol{X}(0)$ and $\boldsymbol{V}(0)$.

Rewrite (3-3)-a as (3-4) by replacing $k$ by $k$-1. Note that $\boldsymbol{X}(k)$ only appears on the LHS and the RHS only contains coefficients up to order $k$-1. Therefore $\boldsymbol{X}(k)$ can be explicitly solved from calculated lower order coefficients.

$$\boldsymbol{X}(k) = \frac{1}{k}\boldsymbol{F}\left(\boldsymbol{X}(l), \boldsymbol{V}(l)\right), l = 0 \ldots k-1 \tag{3-4}$$

In contrast, from (3-3)-b, solving $\boldsymbol{V}(k)$ is not straightforward since it appears on both the LHS and RHS and the vector-valued function $\boldsymbol{I}(\cdot)$ is nonlinear. Later in Chapter 3.3, we will prove that the coefficients of current injection $\boldsymbol{I}(k)$ satisfy a formally linear equation (3-5) about $\boldsymbol{V}(k)$.

$$\begin{aligned} \boldsymbol{I}(k) &= \boldsymbol{A}\boldsymbol{V}(k) + \boldsymbol{B} \\ \boldsymbol{A} &= \boldsymbol{A}\left(\boldsymbol{X}(l_1), \boldsymbol{V}(l_2)\right), l_1 = 0 \ldots k \\ \boldsymbol{B} &= \boldsymbol{B}\left(\boldsymbol{X}(l_1), \boldsymbol{V}(l_2)\right), l_2 = 0 \ldots k-1 \end{aligned} \tag{3-5}$$

Note that the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ still contain nonlinear functions that only involve the $(k\text{-}1)^{\text{th}}$ and lower order coefficients on bus voltages, so they do not affect the solvability of $\boldsymbol{V}(k)$. Finally, $\boldsymbol{V}(k)$ is explicitly solved in (3-6) after substituting (3-5) into (3-3)-b. The detailed derivation of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is presented in Chapter 3.3.

$$\boldsymbol{V}(k) = \left(\boldsymbol{Y}_{bus} - \boldsymbol{A}\right)^{-1}\boldsymbol{B} \tag{3-6}$$

For complex variables and parameters in (3-1)- (3-6) such as current injection vector $\boldsymbol{i}$, bus voltage vector $\boldsymbol{v}$, DTs $\boldsymbol{I}(k)$ and $\boldsymbol{V}(k)$, and admittance matrix $\boldsymbol{Y}_{bus}$, their real and imaginary parts are separate as follows, where $N$ is the number of buses.

$$\boldsymbol{i} = \left[ i_{x,1}, i_{y,1} \ldots i_{x,N}, i_{y,N} \right]^T$$

$$\boldsymbol{v} = \left[ v_{x,1}, v_{y,1} \ldots v_{x,N}, v_{y,N} \right]^T$$

$$\boldsymbol{I}(k) = \left[ I_{x,1}(k), I_{y,1}(k) \ldots I_{x,N}(k), I_{y,N}(k) \right]^T$$

$$\boldsymbol{V}(k) = \left[ V_{x,1}(k), V_{y,1}(k) \ldots V_{x,N}(k), V_{y,N}(k) \right]^T$$

$$\boldsymbol{Y}_{bus} = \begin{bmatrix} \boldsymbol{Y}_{11} & \ldots & \boldsymbol{Y}_{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Y}_{N1} & \ldots & \boldsymbol{Y}_{NN} \end{bmatrix} \text{ where } \boldsymbol{Y}_{ij} = \begin{bmatrix} G_{ij} & B_{ij} \\ -B_{ij} & G_{ij} \end{bmatrix}$$

**Remark:** There are two important observations: 1) from (3-5) that current injections and bus voltages, which are coupled by nonlinear network equations in (3-1), turn out to have linear relationships in terms of their coefficients after DT; 2) coefficients on bus voltages can be explicitly solved by (3-1) and then used to calculate bus voltages by (3-2) in a straightforward manner, which is different from using a conventional power flow solver to calculate bus voltages by numerical iterations. The proposed DT based method for solving DAEs differentiates itself from the traditional solution schemes which rely on iterative numerical methods such as the family of Newton Raphson (NR) methods.

## 3.2 DTs of Power System DAE Model

Typically, a power system DAE model contains differential equations for each generator and its controllers, current injection equations for all generator and load buses, and the transmission network equation. DTs of differential equations are provided in

previous chapter and DTs of current injection equations and the network equation are derived in this section.

### 3.2.1 Vectorized Transformation Rules

In power system DAE model, currents and voltages are usually written as matrix forms using rectangular coordinates. To make the expression of the derived DT more compact, this section extends the existing transformation rules for scalar valued functions to vectorized transformation rules so as to be directly applied to a vector valued function without expanding it into many scalar valued functions first. The proposition 1 provides six vectorized transformation rules in (3-7) for vector or matrix operations that often appear in a power system DAE model. These rules can be easily obtained by applying the existing transformation rules to each element of the vector valued function and their proofs are omitted.

**Proposition 1:** Given $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ as vector-valued functions having DTs as $\boldsymbol{X}(k)$ and $\boldsymbol{Y}(k)$, $h(t)$ and $H(k)$ as a scalar function and its DT, and $\boldsymbol{c}$ and $\boldsymbol{d}$ are constant matrices, the transformation rules in (3-7) hold.

$$\text{i) } \boldsymbol{x}(t) \pm \boldsymbol{y}(t) \;\text{®}\; \boldsymbol{X}(k) \pm \boldsymbol{Y}(k); \quad \text{ii) } \boldsymbol{x}(t)^T \;\text{®}\; \boldsymbol{X}(k)^T$$

$$\text{iii) } \boldsymbol{c}\boldsymbol{x}(t) \;\text{®}\; \boldsymbol{c}\boldsymbol{X}(k); \quad \text{iv) } \boldsymbol{x}(t)\boldsymbol{d} \;\text{®}\; \boldsymbol{X}(k)\boldsymbol{d}$$

$$\text{v) } \boldsymbol{x}(t)\boldsymbol{y}(t) \;\text{®}\; \boldsymbol{X}(k) \text{Ä} \boldsymbol{Y}(k) = \sum_{m=0}^{k} \boldsymbol{X}(m)\boldsymbol{Y}(k-m) \tag{3-7}$$

$$\text{vi) } \frac{1}{h(t)}\boldsymbol{y}(t) \;\text{®}\; \frac{1}{H(0)}\left(\boldsymbol{Y}(k) - \sum_{m=0}^{k-1} H(k-m)\boldsymbol{Z}(m)\right)$$

### 3.2.2 DTs of the Current Injection Equation of Generators

Consider the detailed 6th order synchronous generator model in. The current injection using the *d*-*q* coordinate system is given in (3-8). The coordination transformation between *d*-*q* and *x*-*y* coordinate system is given in (3-9). Variables $i_d$, $i_q$ are the *d*-axis and *q*-axis stator currents; $e''_d$, $e''_q$ are *d*-axis and *q*-axis sub-transient voltages; $v_d$ and $v_q$ are the *d*-axis and *q*-axis terminal voltages; $\delta$ is the rotor angle. Parameters $x''_d$ , $x''_q$ and $r_a$ are the *d*-axis and *q*-axis sub-transient reactance and internal resistance, respectively.

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \boldsymbol{y_a} \left( \begin{bmatrix} e''_d \\ e''_q \end{bmatrix} - \begin{bmatrix} v_d \\ v_q \end{bmatrix} \right) \text{ where } \boldsymbol{y_a} = \begin{bmatrix} r_a & -x''_q \\ x''_d & r_a \end{bmatrix}^{-1} \tag{3-8}$$

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \boldsymbol{r} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \quad \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \boldsymbol{r} \begin{bmatrix} v_d \\ v_q \end{bmatrix} \text{ where } \boldsymbol{r} = \begin{bmatrix} \sin d & \cos d \\ -\cos d & \sin d \end{bmatrix} \tag{3-9}$$

The current injection under the *x*-*y* axis is given in (3-10) by combining (3-8)-(3-9).

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \tau \begin{bmatrix} e''_d \\ e''_q \end{bmatrix} - \lambda \begin{bmatrix} v_x \\ v_y \end{bmatrix} \text{ where } \begin{cases} \tau = \boldsymbol{r}\boldsymbol{y_a} \\ \lambda = \tau \boldsymbol{r}^{\boldsymbol{T}} \end{cases} \tag{3-10}$$

The DT of (3-10) is given in (3-11).

$$\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = \mathrm{G}(k) \ddot{A} \begin{bmatrix} E''_d(k) \\ E''_q(k) \end{bmatrix} - \mathrm{L}(k) \ddot{A} \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} \tag{3-11}$$

For details of the derivation, the RHS of (3-11) is obtained using rules i) and v), where the $\boldsymbol{\Gamma}(k)$ and $\boldsymbol{\Lambda}(k)$ are respectively DTs of $\tau$ and $\lambda$, given by rules ii), iv) and v) as follows. $\boldsymbol{R}(k)$ is the DT of the matrix $\boldsymbol{r}$, where $\Phi(k)$ and $\Psi(k)$ are DTs of sine and cosine functions, respectively.

$$\tau = \boldsymbol{r}\boldsymbol{y_a} \;\; ® \;\; \mathrm{G}(k) = \boldsymbol{R}(k)\boldsymbol{y_a}$$

$$l = t\boldsymbol{r}^{T} \Rightarrow \mathrm{L}(k) = \mathrm{G}(k) \otimes \boldsymbol{R}(k)^{T}$$

$$\boldsymbol{R}(k) = \begin{bmatrix} \mathrm{F}(k) & \mathrm{Y}(k) \\ -\mathrm{Y}(k) & \mathrm{F}(k) \end{bmatrix}$$

Eq. (3-11) contains the convolution of a 2×2 matrix and a 2×1 vector, and its calculation is the same as the convolution of two matrices in the rule v). The detailed expression is following.

$$\begin{bmatrix} J_x(k) \\ J_y(k) \end{bmatrix} = \sum_{m=0}^{k} \mathrm{G}(m) \begin{bmatrix} E_d''(k-m) \\ E_q''(k-m) \end{bmatrix} - \sum_{m=0}^{k} \mathrm{L}(m) \begin{bmatrix} V_x(k-m) \\ V_y(k-m) \end{bmatrix}$$

### 3.2.3 DTs of the Current Injection Equation of Loads

Consider the ZIP load model in (3-12) where $p$ and $q$ are the active and reactive power loads, respectively; $v_t$ is the bus voltage magnitude defined in (3-13) and $u$ equals its square; $p_0$, $q_0$ and $v_{t0}$ are the steady state active power, reactive power and bus voltage; $a_p$ and $a_q$ are the percentages of constant impedance load; $b_p$ and $b_q$ are the percentages of constant current load; and $c_p$ and $c_q$ are the percentages of constant power load. There are $a_p + b_p + c_p = 1$ and $a_q + b_q + c_q = 1$.

$$\begin{cases} p = p_0\left(a_p\left(\dfrac{v_t}{v_{t0}}\right)^2 + b_p\left(\dfrac{v_t}{v_{t0}}\right) + c_p\right) \\ q = q_0\left(a_q\left(\dfrac{v_t}{v_{t0}}\right)^2 + b_q\left(\dfrac{v_t}{v_{t0}}\right) + c_q\right) \end{cases} \tag{3-12}$$

$$v_t = \sqrt{u}, \quad u = v_x^2 + v_y^2 \tag{3-13}$$

The current injected to the network can be calculated from the active and reactive power injections, and is written in matrix forms in (3-14) where $\boldsymbol{\beta}_a$, $\boldsymbol{\beta}_b$ and $\boldsymbol{\beta}_c$ are constant matrices.

$$\begin{bmatrix} \dot{I}_x \\ \dot{I}_y \end{bmatrix} = \begin{bmatrix} \dot{I}_x \\ \dot{I}_y \end{bmatrix}_a + \begin{bmatrix} \dot{I}_x \\ \dot{I}_y \end{bmatrix}_b + \begin{bmatrix} \dot{I}_x \\ \dot{I}_y \end{bmatrix}_c$$
$$= \frac{1}{v_{t0}^2} b_a \begin{bmatrix} V_x \\ V_y \end{bmatrix} + \frac{1}{v_{t0}} \frac{1}{v_t} b_b \begin{bmatrix} V_x \\ V_y \end{bmatrix} + \frac{1}{u} b_c \begin{bmatrix} V_x \\ V_y \end{bmatrix} \tag{3-14}$$

$$b_a = \begin{bmatrix} p_0 a_p & q_0 a_q \\ q_0 a_q & p_0 a_p \end{bmatrix} \quad b_b = \begin{bmatrix} p_0 b_p & q_0 b_q \\ q_0 b_q & p_0 b_p \end{bmatrix} \quad b_c = \begin{bmatrix} p_0 c_p & q_0 c_q \\ q_0 c_q & p_0 c_p \end{bmatrix}$$

DTs of $u$ and $v_t$ are given in (3-15)-(3-16). Then, DTs of the RHS terms in (3-14) can be obtained using rules in (3-7) as explained in the following.

$$U(k) = V_x(k) \ddot{A} V_x(k) + V_y(k) \ddot{A} V_y(k) \tag{3-15}$$

$$V_t(k) = \frac{1}{2V_t(0)} U(k) - \frac{1}{2V_t(0)} \sum_{m=1}^{k-1} V_t(m) V_t(k-m) \tag{3-16}$$

The first term in RHS of (3-14) is the current injection of constant impedance load. It is the product of a constant number $1/v_{t0}^2$, a constant matrix $\boldsymbol{\beta_a}$ and a vector valued function $[v_x, v_y]^T$. Therefore, its DT is given in (3-17) using the rule iii).

$$\begin{bmatrix} \dot{I}_x \\ \dot{I}_y \end{bmatrix} = \frac{1}{v_{t0}^2} b_a \begin{bmatrix} V_x \\ V_y \end{bmatrix} \circledR \begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = \frac{1}{v_{t0}^2} b_a \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} \tag{3-17}$$

The second term in RHS of (3-14) is the current injection of constant current load with DT in (3-18). It is transformed by three steps. First, the product of the constant matrix $\boldsymbol{\beta_b}$ and the vector valued function $[v_x, v_y]^T$ is transformed using the rule iii). Then, the division of the vector valued function $\boldsymbol{\beta_b}[v_x, v_y]^T$ and the scalar valued function $v_t$ is transformed using the rule vi). Finally, the product of the constant number $1/v_{t0}$ and the vector valued function $1/v_t\, \boldsymbol{\beta_b}[v_x, v_y]^T$ is transformed using the rule iii).

$$\begin{bmatrix} \dot{I}_x \\ \dot{I}_y \end{bmatrix} = \frac{1}{v_{t0}} \frac{1}{v_t} b_b \begin{bmatrix} V_x \\ V_y \end{bmatrix} \circledR$$
$$\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = \frac{1}{v_{t0}} \frac{1}{V_t(0)} \left( b_b \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} - \sum_{m=0}^{k-1} V_t(k-m) \begin{bmatrix} I_x(m) \\ I_y(m) \end{bmatrix} \right) \tag{3-18}$$

The third term in RHS of (3-14) is the current injection of constant power load. It contains the product of a constant matrix $\boldsymbol{\beta_c}$ and a vector valued function $[v_x, v_y]^T$, then divided by a scalar valued function $u$. Similar with the constant current load, its DT is given in (3-19) using rules iii) and vi).

$$\begin{aligned} \begin{bmatrix} I_x \\ I_y \end{bmatrix} &= \frac{1}{u} b_c \begin{bmatrix} v_x \\ v_y \end{bmatrix} \\ \begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} &= \frac{1}{U(0)} \left( b_c \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} - \sum_{m=0}^{k-1} U(k-m) \begin{bmatrix} I_x(m) \\ I_y(m) \end{bmatrix} \right) \end{aligned} \tag{3-19}$$

Finally, the DT of current injection equation (3-14) is given in (3-20) by summing DTs of each term (3-17)-(3-19) using the rule i).

$$\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = \begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix}_1 + \begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix}_2 + \begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix}_3 \tag{3-20}$$

### 3.2.4 DTs of the Network Equation

The network equation is in (3-21), which couples the current injections of all generators and loads. Its DT is given in (3-22).

$$\boldsymbol{i} = \boldsymbol{Y_{bus}} \boldsymbol{v_{bus}}$$

$$\begin{bmatrix} i_{x,m} \\ i_{y,m} \end{bmatrix} = \begin{cases} \text{RHS of } (10) \text{ for generator buses} \\ -\text{ RHS of } (14) \text{ for load buses} \\ \text{RHS of (10) - RHS of (14), for buses with} \\ \qquad \text{both generators and loads} \end{cases} \tag{3-21}$$

$$\boldsymbol{I}(k) = \boldsymbol{Y_{bus}} \boldsymbol{V}(k)$$

$$\begin{bmatrix} I_{x,m}(k) \\ I_{y,m}(k) \end{bmatrix} = \begin{cases} \text{RHS of } (11) \text{ for generator buses} \\ -\text{ RHS of } (20) \text{ for load buses} \\ \text{RHS of } (11) - \text{ RHS of } (20) \text{ for buses with} \\ \qquad \text{both generators and loads} \end{cases} \tag{3-22}$$

## 3.3 Solving Power Series Coefficients of State Variables and Bus Voltages

### 3.3.1 Linear Relationship Between Current Injection and Bus Voltage in terms of Power Series Coefficients

**Proposition 2**: The transformed current injections in (3-11) and (3-20) for generators and loads respectively satisfy equations (3-23) and (3-24), which are formally linear.

$$\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = \boldsymbol{A_g} \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} + \boldsymbol{B_g} \tag{3-23}$$

$$\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = \boldsymbol{A_l} \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} + \boldsymbol{B_l} \tag{3-24}$$

The proofs are given below and the detailed expressions of matrices $\boldsymbol{A_g}$, $\boldsymbol{B_g}$, $\boldsymbol{A_l}$ and $\boldsymbol{B_l}$ are in (3-25) and (3-29).

**Proof of (3-23) in Proposition 2:** Rewrite equation (3-11) as follows. Define $\boldsymbol{A_g}$ and $\boldsymbol{B_g}$ as (3-25). It is easy to confirm that $\boldsymbol{A_g}$ and $\boldsymbol{B_g}$ do not depend on $V_x(k)$ and $V_y(k)$.

$$\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = G(k) \cdot \begin{bmatrix} E''_d(k) \\ E''_q(k) \end{bmatrix} - \sum_{m=0}^{k-1} L(k-m) \begin{bmatrix} V_x(m) \\ V_y(m) \end{bmatrix} - L(0) \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix}$$

$$\boldsymbol{A_g} = -L(0), \quad \boldsymbol{B_g} = G(k) \cdot \begin{bmatrix} E''_d(k) \\ E''_q(k) \end{bmatrix} - \sum_{m=0}^{k-1} L(k-m) \begin{bmatrix} V_x(m) \\ V_y(m) \end{bmatrix} \tag{3-25}$$

**Proof of (3-24) in Proposition 2:** To prove (3-24), we only need to prove each component of the ZIP load in (3-17)-(3-19) can be written into following three equations, respectively.

$$\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = \boldsymbol{A_z} \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} + \boldsymbol{B_z},$$

$$\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = \boldsymbol{A_i} \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} + \boldsymbol{B_i}$$

$$\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} = \boldsymbol{A_p} \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} + \boldsymbol{B_p}$$

**Part a)** The DT of current injections of constant impedance load (3-17) can be easily written into above forms, by defining $\boldsymbol{A_z}$, $\boldsymbol{B_z}$ as (3-26).

$$\boldsymbol{A_z} = \frac{1}{v_{t0}^2} b_a, \quad \boldsymbol{B_z} = \boldsymbol{0}_{2\times 2} \tag{3-26}$$

**Part *b***) The DT of current injections of constant current load (3-18) is rewritten as follows.

$$\begin{aligned}
\begin{bmatrix} I_x(k) \\ I_y(k) \end{bmatrix} &= \frac{1}{v_{t0}^2} \left( b_b \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} - \sum_{m=0}^{k-1} V_t(k-m) \begin{bmatrix} I_x(m) \\ I_y(m) \end{bmatrix} \right) \\
&= \frac{1}{v_{t0}^2} b_b \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} - \frac{1}{v_{t0}^2} \sum_{m=1}^{k-1} V_t(k-m) \begin{bmatrix} I_x(m) \\ I_y(m) \end{bmatrix} - \frac{1}{v_{t0}^2} V_t(k) \begin{bmatrix} I_x(0) \\ I_y(0) \end{bmatrix} \\
&@ \boldsymbol{A_{i,1}} \begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} + \boldsymbol{B_{i,1}} - \frac{1}{v_{t0}^2} V_t(k) \begin{bmatrix} I_x(0) \\ I_y(0) \end{bmatrix}
\end{aligned}$$

The third term can be further written as follows after substituting $V_t(k)$ in (3-16).

$$\begin{aligned}
-\frac{1}{v_{t0}^2} V_t(k) \begin{bmatrix} I_x(0) \\ I_y(0) \end{bmatrix} &= -\frac{1}{2v_{t0}^3} \left( U(k) - \sum_{m=1}^{k-1} V_t(m) V_t(k-m) \right) \begin{bmatrix} I_x(0) \\ I_y(0) \end{bmatrix} \\
&@ -\frac{1}{2v_{t0}^3} U(k) \begin{bmatrix} I_x(0) \\ I_y(0) \end{bmatrix} + \boldsymbol{B_{i,2}}
\end{aligned}$$

The first term can be further written as follows after substituting $U(k)$ in (3-15).

$$
\begin{aligned}
&-\frac{1}{2v_{t0}^{3}}U(k)\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix} = -\frac{1}{2v_{t0}^{3}}\{V_x(k)\ddot{V}_x(k)+V_y(k)\ddot{V}_y(k)\}\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}\\
&= -\frac{1}{2v_{t0}^{3}}\left\{\sum_{m=1}^{k-1}V_x(m)V_x(k-m)+\sum_{m=1}^{k-1}V_y(m)V_y(k-m)\right\}\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}\\
&\quad -\frac{1}{v_{t0}^{3}}\{V_x(0)V_x(k)+V_y(0)V_y(k)\}\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}\\
&\equiv \boldsymbol{B}_{i,3}+\boldsymbol{A}_{i,2}\begin{bmatrix}V_x(k)\\ V_y(k)\end{bmatrix} \text{ where } \boldsymbol{A}_{i,2} = -\frac{1}{v_{t0}^{3}}\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}\begin{bmatrix}V_x(0), V_y(0)\end{bmatrix}
\end{aligned}
$$

Finally, define $\boldsymbol{A_i}$ and $\boldsymbol{B_i}$ as (3-27). It is easy to confirm that $\boldsymbol{A_i}$ and $\boldsymbol{B_i}$ do not depend on $V_x(k)$ and $V_y(k)$.

$$
\boldsymbol{A_i} = \boldsymbol{A}_{i,1}+\boldsymbol{A}_{i,2}, \quad \boldsymbol{B_i} = \boldsymbol{B}_{i,1}+\boldsymbol{B}_{i,2}+\boldsymbol{B}_{i,3} \tag{3-27}
$$

**Part *c***) Equation (3-19) is written as follows.

$$
\begin{aligned}
&\begin{bmatrix}\Gamma_x(k)\\ \Gamma_y(k)\end{bmatrix} = \frac{1}{v_{t0}^{2}}\left(b_c\begin{bmatrix}V_x(k)\\ V_y(k)\end{bmatrix}-\sum_{m=0}^{k-1}U(k-m)\begin{bmatrix}\Gamma_x(m)\\ \Gamma_y(m)\end{bmatrix}\right)\\
&= \frac{1}{v_{t0}^{2}}b_c\begin{bmatrix}V_x(k)\\ V_y(k)\end{bmatrix}-\frac{1}{v_{t0}^{2}}\sum_{m=1}^{k-1}U(k-m)\begin{bmatrix}\Gamma_x(m)\\ \Gamma_y(m)\end{bmatrix}-\frac{1}{v_{t0}^{2}}U(k)\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}\\
&\equiv \boldsymbol{A}_{p,1}\begin{bmatrix}V_x(k)\\ V_y(k)\end{bmatrix}+\boldsymbol{B}_{p,1}-\frac{1}{v_{t0}^{2}}U(k)\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}
\end{aligned}
$$

The third term can be further written as follows after substituting $U(k)$ in (3-15). It is easy to confirm that $\boldsymbol{A_p}$ and $\boldsymbol{B_p}$ defined in (3-28) do not depend on $V_x(k)$ and $V_y(k)$.

$$
\begin{aligned}
&-\frac{1}{v_{t0}^{2}}U(k)\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}\\
&= -\frac{1}{v_{t0}^{2}}\left\{\sum_{m=1}^{k-1}V_x(m)V_x(k-m)+\sum_{m=1}^{k-1}V_y(m)V_y(k-m)\right\}\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}\\
&\quad -\frac{2}{v_{t0}^{2}}\{V_x(0)V_x(k)+V_y(0)V_y(k)\}\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}\\
&\equiv \boldsymbol{B}_{p,2}+\boldsymbol{A}_{p,2}\begin{bmatrix}V_x(k)\\ V_y(k)\end{bmatrix} \text{ where } \boldsymbol{A}_{p,2} = -\frac{2}{v_{t0}^{2}}\begin{bmatrix}\Gamma_x(0)\\ \Gamma_y(0)\end{bmatrix}\begin{bmatrix}V_x(0), V_y(0)\end{bmatrix}
\end{aligned}
$$

$$\boldsymbol{A_p} = \boldsymbol{A_{p,1}} + \boldsymbol{A_{p,2}}, \quad \boldsymbol{B_p} = \boldsymbol{B_{p,1}} + \boldsymbol{B_{p,2}} \tag{3-28}$$

Finally, (3-24) is proved by defining $\boldsymbol{A_l}$ and $\boldsymbol{B_l}$ as (3-29).

$$\begin{aligned} \boldsymbol{A_l} &= \boldsymbol{A_z} + \boldsymbol{A_i} + \boldsymbol{A_p} \\ \boldsymbol{B_l} &= \boldsymbol{B_z} + \boldsymbol{B_i} + \boldsymbol{B_p} \end{aligned} \tag{3-29}$$

Using this proposition, current injections of all buses can be written as (3-5) with $\boldsymbol{A}$ and $\boldsymbol{B}$ in (3-30).

$$\begin{aligned} \boldsymbol{A} &= diag\left(\boldsymbol{A}_1, \boldsymbol{A}_2 \ldots \boldsymbol{A}_N\right), \boldsymbol{A}_n = \begin{cases} \boldsymbol{A}_g, \text{for generator buses} \\ -\boldsymbol{A}_l, \text{for load buses} \\ \boldsymbol{A}_g - \boldsymbol{A}_l, \text{for buses with both generators and loads} \end{cases} \\ \boldsymbol{B} &= \left[\boldsymbol{B}_1^T, \boldsymbol{B}_2^T \ldots \boldsymbol{B}_N^T\right]^T, \boldsymbol{B}_n = \begin{cases} \boldsymbol{B}_g, \text{for generator buses} \\ -\boldsymbol{B}_l, \text{for load buses} \\ \boldsymbol{B}_g - \boldsymbol{B}_l, \text{for buses with both generators and loads} \end{cases} \end{aligned} \tag{3-30}$$

### 3.3.2 Non-iterative Algorithm to Solve Power Series Coefficients

Following the basic idea in Fig. 12, **Algorithm 2** is further designed to solve power series coefficients of both state variables and bus voltages up to any desired order. Note that all the coefficients are explicitly calculated with no iteration.

---

**Algorithm 2: Solve Coefficients**

**Input**: initial values of state variables and bus voltages $\boldsymbol{x}_0, \boldsymbol{v}_0$

**Output**: any order coefficients $\boldsymbol{X}(k), \boldsymbol{V}(k), k = 0 \ldots K$

Initialization: $\boldsymbol{X}(0) = \boldsymbol{x}_0, \boldsymbol{V}(0) = \boldsymbol{v}_0$

1. Calculate the matrix $\boldsymbol{A}$
    1.1 Calculate the matrix $\boldsymbol{A_g}$ for generators
    1.2 Calculate the matrix $\boldsymbol{A_l}$ for loads

2. Calculate the matrix $(\boldsymbol{Y}_{bus} - \boldsymbol{A})$ and solve $(\boldsymbol{Y}_{bus} - \boldsymbol{A})^{-1}$

**for** $k = 1 : K$

3. Solve $\boldsymbol{X}(k)$: $\boldsymbol{X}(k) = \frac{1}{k}\boldsymbol{F}(\boldsymbol{X}(l), \boldsymbol{V}(l)), l = 0 \ldots k-1$

3.1 Solve state variables of governors and turbines

3.2 Solve state variables of 6th order generator model

3.3 Solve state variables of IEEE Type I exciter model

4. Calculate the matrix $\boldsymbol{B}$

4.1 Calculate the matrix $\boldsymbol{B}_g$ for generators

4.2 Calculate the matrix $\boldsymbol{B}_l$ for loads

5. Solve $\boldsymbol{V}(k)$: $\boldsymbol{V}(k) = (\boldsymbol{Y}_{bus} - \boldsymbol{A})^{-1}\boldsymbol{B}$

**end**

### 3.3.3 Extension

This section further discusses the linear relationship among non-state variables for a frequency dependent load model. When considering the impact of frequency changes, the ZIP load model is changed to a set of DAEs in (3-31), where $\theta$ is the bus voltage angle, $\Delta f$ is the frequency change, $\boldsymbol{i}_l$ is the current injection of the ZIP load model, $\boldsymbol{i}_{l,f}$ is the current injection after considering the frequency change, and $d$ is a constant.

$$\begin{aligned} \dot{\theta} &= \Delta f \\ \boldsymbol{i}_{l,f} &= \boldsymbol{i}_l(1 + d\Delta f) \end{aligned} \tag{3-31}$$

The DT of (3-31) are given in (3-32)-(3-33). For the DT of the algebraic current injection equation (3-33), it is also proved to satisfy a formally linear equation in (3-34) with detailed proofs given below.

$$Q(k) = \frac{1}{k}\Delta F(k-1) \tag{3-32}$$

$$\boldsymbol{I}_{l,f}(k) = \boldsymbol{I}_l(k) + d\boldsymbol{I}_l(k) \ddot{A} \Delta F(k) \tag{3-33}$$

$$\boldsymbol{I}_{l,f}(k) = \begin{bmatrix} \boldsymbol{A}_l & \boldsymbol{A}_f \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_l(k) \\ \Delta F(k) \end{bmatrix} + \boldsymbol{B}_l \tag{3-34}$$

**Proof of (3-34):** Rewrite (3-33) as follows.

$$\boldsymbol{I}_{l,f}(k) = \boldsymbol{I}_l(k) + d\boldsymbol{I}_l(k) \text{ Ä } \mathrm{D}F(k)$$
$$= (1 + d\mathrm{D}F(0))\boldsymbol{I}_l(k) + d\boldsymbol{I}_l(0)\mathrm{D}F(k) + d\sum_{m=1}^{k-1} \boldsymbol{I}_l(m)\mathrm{D}F(k-m))$$

Since $\boldsymbol{I}_l(k) = \boldsymbol{A}_l\boldsymbol{V}_l(k) + \boldsymbol{B}_l$ has been proved for the ZIP load model, the above equation is further rewritten as follows.

$$\boldsymbol{I}_{l,f}(k) = (1 + d\mathrm{D}F(0))(\boldsymbol{A}_l\boldsymbol{V}_l(k) + \boldsymbol{B}_l) + d\boldsymbol{I}_l(0)\mathrm{D}F(k) + d\sum_{m=1}^{k-1} \boldsymbol{I}_l(m)\mathrm{D}F(k-m))$$
$$= (1 + d\mathrm{D}F(0))\boldsymbol{A}_l\boldsymbol{V}_l(k) + d\boldsymbol{I}_l(0)\mathrm{D}F(k) + (1 + d\mathrm{D}F(0))\boldsymbol{B}_l + d\sum_{m=1}^{k-1} \boldsymbol{I}_l(m)\mathrm{D}F(k-m))$$

Finally, by defining $\boldsymbol{A}_l¢, \boldsymbol{A}_f, \boldsymbol{B}_l¢$ in (3-35), the linear relationship (3-34) is satisfied, where $\boldsymbol{A}_l¢, \boldsymbol{A}_f, \boldsymbol{B}_l¢$ do not depend on $V_x(k)$ and $V_y(k)$.

$$\boldsymbol{A}_l¢ = (1 + d\mathrm{D}F(0))\boldsymbol{A}_l; \boldsymbol{A}_f = d\boldsymbol{I}_l(0);$$
$$\boldsymbol{B}_l¢ = (1 + d\mathrm{D}F(0))\boldsymbol{B}_l + d\sum_{m=1}^{k-1} \boldsymbol{I}_l(m)\mathrm{D}F(k-m)) \tag{3-35}$$

Two additional variables are introduced, i.e., the state variable $\theta$ and the non-state variable $\Delta f$, and their solutions are also approximated by power series of time in (3-36), where the coefficients $\Theta(k)$ are solved together with the coefficients of state variables $\boldsymbol{X}(k)$ and coefficients $\Delta F(k)$ are solved together with the coefficients of bus voltages $\boldsymbol{V}(k)$.

$$q = \sum_{0}^{K} \mathrm{Q}(k)t^k; \quad \mathrm{D}f = \sum_{0}^{K} \mathrm{D}F(k)t^k \tag{3-36}$$

## 3.4 Case Study

The proposed method is first illustrated on a 3-machine 9-bus power system. Then, to validate the accuracy, time performance and robustness of the proposed method on solving practical high-dimensional nonlinear DAEs, the 327-machine 2383-bus Polish system with detailed models on generators, exciters, governors, turbines, and ZIP loads are used. In the ZIP load model, the percentages of each component are 20%, 30% and 50% respectively.

Two widely used solution approaches are implemented for comparison: 1) **TRAP-NR** method where the differential equations are algebraized by implicit trapezoidal method (TRAP) first and then solved *simultaneously* with the network equations by Newton Raphson (NR) method. 2) **ME-NR** method using a partitioned scheme where the differential and network equations are *alternatively* solved by explicit modified Euler method (ME) and NR method respectively. The time step length of both the TRAP-NR method and ME-NR method is $1\times10^{-3}$ s, while the proposed method prolongs the time step length to 10 times and still achieves better accuracy.

For a fair comparison, the benchmark result is given by the TRAP-NR method using a small enough time step length of $1\times10^{-4}$ s and errors of the proposed method, the TRAP-NR method and the ME-NR method are calculated by their differences from the benchmark result. Simulations are conducted in MATLAB R2018b on a personal computer with i7-6700U CPU.

### 3.4.1 Illustration on a 3-machine 9-bus Power System

The 3-machine 9-bus power system in [61] is used to illustrate the proposed method. The system contains generators at buses 1 to 3 equipped with governors and exciters whose differential equations can be found in Chapter 2, ZIP loads at buses 5, 6 and 8 and transition buses 4, 7 and 9. Equation (3-37) shows the network equations on current injections:

$$
\begin{aligned}
\begin{bmatrix} i_{x,n} \\ i_{y,n} \end{bmatrix} &= t_n \begin{bmatrix} i''_{d,n} \\ i''_{q,n} \end{bmatrix} - l_n \begin{bmatrix} v_{x,n} \\ v_{y,n} \end{bmatrix} \qquad if\ n = 1,2,3 \\
\begin{bmatrix} i_{x,n} \\ i_{y,n} \end{bmatrix} &= \frac{1}{v_{t0,n}^2} b_{a,n} \begin{bmatrix} v_{x,n} \\ v_{y,n} \end{bmatrix} + \frac{1}{v_{t0,n}} \frac{1}{\sqrt{v_{x,n}^2 + v_{y,n}^2}} b_{b,n} \begin{bmatrix} v_{x,n} \\ v_{y,n} \end{bmatrix} \\
&\quad + \frac{1}{v_{x,n}^2 + v_{y,n}^2} b_{c,n} \begin{bmatrix} v_{x,n} \\ v_{y,n} \end{bmatrix} \qquad if\ n = 5,6,8 \\
\begin{bmatrix} i_{x,1} \\ \vdots \\ i_{y,9} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{Y}_{11} & \cdots & \boldsymbol{Y}_{19} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Y}_{91} & \cdots & \boldsymbol{Y}_{99} \end{bmatrix} \begin{bmatrix} v_{x,1} \\ \vdots \\ v_{y,9} \end{bmatrix}
\end{aligned}
\tag{3-37}
$$

Apply DT to (3-37) according to (3-11), (3-17)-(3-20). Coefficients of state variables and bus voltages, i.e. $\boldsymbol{X}(1)$, $\boldsymbol{V}(1)$,…, $\boldsymbol{X}(K)$,$\boldsymbol{V}(K)$, are recursively calculated starting from $\boldsymbol{X}(0)$ and $\boldsymbol{V}(0)$. For illustration purpose, calculation of $\boldsymbol{X}(1)$ and $\boldsymbol{V}(1)$ is explained as follows. First, calculate $\boldsymbol{X}(1)$ by (3-4) with $k$=1:

$$\boldsymbol{X}(1) = \boldsymbol{F}(\boldsymbol{X}(0), \boldsymbol{V}(0))$$

where detailed equations of $\boldsymbol{F}$ can be found in Chapter 2. Then, write $\boldsymbol{I}(1)$ of all generators and loads as a linear equation about $\boldsymbol{V}(1)$ so as to explicitly solve $\boldsymbol{V}(1)$. For instance, current injections from the constant power load component at bus 5 (i.e. the last term of the second equation in (3-37) can be calculated by the following detailed steps. The remaining current injections can be handled in the similar way.

$$\begin{bmatrix} I_{x,5}(1) \\ I_{y,5}(1) \end{bmatrix}_{\boldsymbol{p}} = \frac{1}{U_5(0)}\left(\boldsymbol{b}_{c,5}\begin{bmatrix} V_{x,5}(1) \\ V_{y,5}(1) \end{bmatrix} - U_5(1)\begin{bmatrix} I_{x,5}(0) \\ I_{y,5}(0) \end{bmatrix}_{\boldsymbol{p}}\right) \qquad (a)$$

$$= \frac{1}{U_5(0)}\left(\boldsymbol{b}_{c,5}\begin{bmatrix} V_x(1) \\ V_y(1) \end{bmatrix} - 2\left(V_{x,5}(0)V_{x,5}(1) + V_{y,5}(0)V_{y,5}(1)\right)\begin{bmatrix} I_{x,5}(0) \\ I_{y,5}(0) \end{bmatrix}_{\boldsymbol{p}}\right) \qquad (b)$$

$$= \frac{1}{U_5(0)}\left(\boldsymbol{b}_{c,5}\begin{bmatrix} V_x(1) \\ V_y(1) \end{bmatrix} - \begin{bmatrix} 2V_{x,5}(0)I_{x,5}(0) & 2V_{y,5}(0)I_{x,5}(0) \\ 2V_{x,5}(0)I_{y,5}(0) & 2V_{y,5}(0)I_{y,5}(0) \end{bmatrix}\begin{bmatrix} V_{x,5}(1) \\ V_{y,5}(1) \end{bmatrix}\right) \qquad (c)\ (3\text{-}38)$$

$$= \frac{1}{U_5(0)}\left(\boldsymbol{b}_{c,5} - \begin{bmatrix} 2V_{x,5}(0)I_{x,5}(0) & 2V_{y,5}(0)I_{x,5}(0) \\ 2V_{x,5}(0)I_{y,5}(0) & 2V_{y,5}(0)I_{y,5}(0) \end{bmatrix}\begin{bmatrix} V_{x,5}(1) \\ V_{y,5}(1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) \qquad (d)$$

$$= \begin{bmatrix} -0.8771 & 0.1566 \\ 0.1566 & 0.8771 \end{bmatrix}\begin{bmatrix} V_{x,5}(1) \\ V_{y,5}(1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} @ \boldsymbol{A}_{\boldsymbol{p},5}\begin{bmatrix} V_{x,5}(1) \\ V_{y,5}(1) \end{bmatrix} + \boldsymbol{B}_{\boldsymbol{p},5} \qquad (e)$$

For load bus 5, the first order coefficient $[I_{x,5}(1), I_{y,5}(1)]_{\boldsymbol{p}}^{\mathrm{T}}$ for constant power load is given in (3-38)-a from (3-19). After substituting the expression of $U(1)$ given in (3-15) and simple matrix operations, it turns to (3-38)-b and (3-38)-c, respectively. Then, all terms containing $[V_{x,5}(1), V_{y,5}(1)]^{\mathrm{T}}$ are combined as a group versus the remaining as another group in (3-38)-d. Since $\boldsymbol{\beta}_{c,5}$ is a constant matrix and all variables, $U_5(0)$, $V_{x,5}(0)$, $V_{y,5}(0)$, $I_{x,5}(0)$, $I_{y,5}(0)$, except for $[V_{x,5}(1), V_{y,5}(1)]^{\mathrm{T}}$ have been known, their values are directly substituted into the equation to have (3-38)-e. Again, $[I_{x,5}(1), I_{y,5}(1)]_{\boldsymbol{p}}^{\mathrm{T}}$ is formally linear about $[V_{x,5}(1), V_{y,5}(1)]^{\mathrm{T}}$ with coefficient matrices denoted by $\boldsymbol{A}_{\boldsymbol{p},5}$ and $\boldsymbol{B}_{\boldsymbol{p},5}$.

After obtaining the linear forms for all current injections, we can combine them into the matrix representation (3-39). For instance, $\boldsymbol{A}_1$ and $\boldsymbol{B}_1$ are equal to the $\boldsymbol{A}_{\boldsymbol{g},1}$ and $\boldsymbol{B}_{\boldsymbol{g},1}$ respectively at generator bus 1.

$$\begin{bmatrix} I_{x,1}(1) \\ \vdots \\ I_{y,9}(1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & & \\ & \mathrm{O} & \\ & & \boldsymbol{A}_9 \end{bmatrix}\begin{bmatrix} V_{x,1}(1) \\ \vdots \\ V_{y,9}(1) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \vdots \\ \boldsymbol{B}_9 \end{bmatrix} \qquad (3\text{-}39)$$

Finally, combining (3-39) with the DTs of the network equation (3-22), $\boldsymbol{V}(1)$ is explicitly solved in (3-40). By recursively conducting this process, coefficients of bus voltages and state variables with any order $k$ can be obtained.

$$\begin{bmatrix} V_{x,1}(1) \\ \vdots \\ V_{y,9}(1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_{11} & \cdots & \boldsymbol{Y}_{19} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Y}_{91} & \cdots & \boldsymbol{Y}_{99} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}_1 & & \boldsymbol{B}_1 \\ & \ddots & \vdots \\ & \boldsymbol{A}_9 & \boldsymbol{B}_9 \end{bmatrix} \tag{3-40}$$

In each time step, solutions of involved variables are approximated by power series of time in (3-2). By performing the above process in multiple time steps, the solutions over a desired simulation range is obtained. Fig. 13a gives the transient voltage trajectories at bus 1 and bus 5 after a large disturbance using both the proposed method with $K$=8 and time step length of 0.01 s, and the TRAP-NR method with time step length of 0.001 s. Fig. 13b further provides the maximum voltage errors of all 9 buses for both methods compared with the benchmark result. It shows that the error of the proposed method is reduced by one order of magnitude compared to that of the TRAP-NR method over the entire simulation period despite the 10 times prolonged time step length.

### 3.4.2 Accuracy and Time Performance

Both stable and unstable scenarios are simulated for the Polish system to validate the accuracy and time performance of the proposed method. Respectively for two scenarios, Fig. 14 and Fig. 15 respectively show the transient responses of rotor angles, rotor speeds and bus voltages simulated by the proposed method. The machine 1 is selected as the reference to calculate relative rotor angles. The maximum errors of rotor angles, rotor speeds, and bus voltages compared with the benchmark results are $3.02\times10^{-5}$ degree, $4.27\times10^{-7}$ Hz, $3.33\times10^{-7}$ p.u. for stable scenario and $2.02\times10^{-5}$ degree, $3.00\times10^{-}$

$^{7}$ Hz, $2.78\times10^{-7}$ p.u. for unstable scenario respectively. It shows the proposed method can accurately simulate both stable and unstable contingencies in the transient stability simulation.

Table 5 gives the maximum errors of all state variables (including rotor angles, rotor speeds, transient and sub-transient voltages, field voltages, etc.) and bus voltages respectively, as well as the computation time per 1-second simulation. The errors of the state variables and the bus voltages using the proposed method are respectively two orders of magnitude lower and one order of magnitude lower than those using the TRAP-NR method and the ME-NR method. Also, the computation speed of the proposed method is around 10 times faster than the other two methods. These results show the proposed method is more efficient and accurate.

Fig. 16 gives the error propagation along the simulation process for four scenarios with the time step length increased to 0.02 s, 0.05s, 0.10s and 0.20s respectively starting from the same initial states at $t$=0. It shows that the error does not accumulate much when the time step length is 0.02 s and 0.05 s. The maximum errors are in the order of magnitude of $10^{-5}$ p.u. and $10^{-3}$ p.u.. respectively. For larger time step lengths, the error becomes unneglectable when the time step length is 0.10 s and even reaches $10^{5}$ p.u. when the time step length is 0.20 s, indicting divergence of the solution.

In this section, the $K$ is determined by gradually increasing its value until the maximum error of all variables satisfies a pre-defined requirement. Table 6 gives the error and the computation time with different values of $K$. It shows that the errors of state variables and bus voltages are decreased when $K$ increases from 2 to 8. And, keeping

increasing $K$ does not further improve the accuracy much but brings more computation burden. Therefore, $K$=8 is selected throughout the case study to meet the accuracy requirement, where the maximum error of all variables is in the order of magnitude of $10^{-6}$ p.u...

Since a large computation burden with transient stability simulation lies in solving linear equations, both sparse matrix and LU factorization techniques are implemented in this work for the DT method, TRAP-NR method and ME-NR method. Table 7 compares the total number $N_{LU}$ of times of LU factorization with three methods in a 1-second simulation. It is calculated by $N_{LU}=n_{LU}\times M$, where $n_{LU}$ is the number of times of LU factorization within each time step and $M$ is the total number of time steps. Within each time step, both the TRAP-NR and the ME-NR method need to perform LU factorization for each iteration unless a so-called very dishonest NR method is applied, but the DT method only needs to perform LU factorization once. Also, the DT method only takes 1/10 of time steps of the other two methods. Therefore, the DT method can significantly reduce the number of times of LU factorization in a simulation.

### 3.4.3 Robustness

The robustness of the proposed method is validated in three sets of cases: 1) stable and unstable scenarios, 2) disturbances with different severities, and 3) different percentages of constant power load.

By comparing the results of stable and unstable scenarios in Table 5, it is observed that the TRAP-NR method and ME-NR method are less accurate and slower in simulating the unstable scenario than in the stable scenario, but the proposed DT-based

method performs almost the same in both scenarios. This is because the system states change significantly in the unstable scenario and the NR method takes more iterations to converge. At each time step, the TRAP-NR method takes averagely 3.004 iterations in the stable scenario and 3.118 iterations in the unstable scenario. For ME-NR method, it takes 2.060 and 2.132 iterations, respectively. In contrast, the proposed method does not require iterations in the solving process, thus having better robustness on unstable scenarios.

Fig. 17 gives the time performance and the average number of iterations of the NR method under different disturbances with increasing severities using the three methods. It shows the computation time of the proposed method is almost the same for different disturbances, but both the TRAP-NR method and ME-NR method take longer time when simulating larger disturbances due to the increased number of iterations.

The time performance and the average number of iterations of the NR method under different percentages of constant power load is in Fig. 18. The higher percentage of constant power load brings stronger nonlinearity to the DAEs, thus making the NR method more difficult or fail to converge. In Fig. 18, the computation time of both the TRAP-NR method and ME-NR method increase significantly with the higher percentage of the constant power load. But the proposed method does not need iterations and its computation time is not affected much, showing it is more robust to handle the strong nonlinearity caused by constant power load.

## 3.5 Conclusion

In this chapter, a DT based non-iterative method is proposed for solving power system DAEs. Current injections and bus voltages coupled by nonlinear network equations in the original state space representation are proved to satisfy a formally linear equation in terms of their power series coefficients after DT. Benefiting from this proposition, solutions of both state variables and non-state voltages are calculated by power series of time whose coefficients are explicitly solved using the designed algorithm with no iteration. Simulation results shows the proposed method effectively reduces the computation burden compared to traditional numerical methods and demonstrates reliable time performance when solving DAEs under large disturbances or with strong nonlinearities.

# CHAPTER 4

# A DYNAMIZED POWER FLOW METHOD BASED ON DIFFERENTIAL TRANSFORMATION

This chapter proposes a novel method for solving and tracing power flow solutions with changes of a loading parameter. Different from the conventional continuation power flow method [62]-[66], which repeatedly solves static AC power flow equations, the proposed method extends the power flow model into a fictitious dynamic system by adding a differential equation on the loading parameter. As a result, the original solution curve tracing problem is converted to solving the time domain trajectories of the reformulated dynamic system. A non-iterative algorithm based on DT is proposed to analytically solve the aforementioned dynamized model in form of power series of time. This chapter proves that the nonlinear power flow equations in the time domain are converted to formally linear equations in the domain of the power series order after the DT, thus avoiding numerical iterations. Case studies on several test systems including a 2383-bus system show the merits of the proposed method.

## 4.1 Problem statement

The conventional power flow equations are given in (4-1)-a where $\bar{S}$ is a vector of the complex power injections, $\bar{V}$ is a vector of bus voltage phasor, and $\mathbf{Y_{bus}}$ is the bus admittance matrix. By adding the product of a loading parameter $\lambda$ and a constant vector $\bar{b}$ to the left-hand side, a general continuum of power flow equations is given in (4-1)-b.

$$\begin{aligned} \bar{S} &= \bar{V}(Y_{bus}\bar{V})^* \qquad \text{(a)} \\ \bar{S} + l\,\bar{b} &= \bar{V}(Y_{bus}\bar{V})^* \qquad \text{(b)} \end{aligned} \tag{4-1}$$

Note that the vector $\bar{b}$ is defined to reflect an arbitrarily direction of load changes, for example, uniform increases of all generation and load, or increases of generation and load at certain buses or zones. Meanwhile, practical operating constraints such as the reactive power limit of generators can be considered during the load change.

Equation (4-1)-b is further written as the general form in (4-2) where $\boldsymbol{g}$ is a nonlinear vector field; $\boldsymbol{y}$ is the bus voltage vector under rectangular coordinates defined as $\boldsymbol{y}=[\boldsymbol{e}^{\mathrm{T}},\boldsymbol{f}^{\mathrm{T}}]^{\mathrm{T}}$, where $\boldsymbol{e}= [e_1,\ldots, e_N]^{\mathrm{T}}$ and $\boldsymbol{f}= [f_1,\ldots, f_N]^{\mathrm{T}}$ are respectively the real and imaginary parts of the bus voltage phasor; $N$ is the total number of buses; $\lambda$ is the loading parameter.

$$0 = \boldsymbol{g}(\boldsymbol{y}, l) \tag{4-2}$$

The goal is to determine how power flow solution $\boldsymbol{y}$ changes with loading parameter $\lambda$, shown in (4-3). After (4-3) is obtained, the other system variables (such as voltage magnitude and power injections) are easily calculated to draw P-V curves.

$$\boldsymbol{y} = \boldsymbol{y}(l) \tag{4-3}$$

Generally, analytical expression of (4-3) is unavailable due to the nonlinearity of g in (4-3). Therefore, a prediction-correction scheme and numerical iterations are needed in conventional CPF method.

## 4.2 Proposed Dynamized Power Flow Method

### 4.2.1 Idea of the proposed method

The proposed method has following four steps.

First, the algebraic equation (4-2) is extended to a set of DAEs by introducing a fictitious time $t$ and adding two new equations, i.e., (4-4)-a and (4-4)-b. Differential equation (4-4)-a is a dynamic system to trace the changes of system variables such as

power or voltages, where $x(t)$ is a state variable and $f(\cdot)$ is a vector field. Algebraic equation (4-4)-b is an ancillary equation that builds the relationship between the newly introduced variable $x(t)$ and the original variables $\boldsymbol{y}(t)$ and $\lambda(t)$. Note that the ancillary function $h$ may not be needed if $x(t)$ is selected from one of the variables in $\boldsymbol{y}(t)$ and $\lambda(t)$.

$$\begin{aligned} \dot{x}(t) &= f(x(t), \boldsymbol{y}(t), l(t)) && (a) \\ 0 &= h(x(t), \boldsymbol{y}(t), l(t)) && (b) \\ 0 &= \boldsymbol{g}(\boldsymbol{y}(t), l(t)), i.e., \ Equ.\ (4.2) && (c) \end{aligned} \tag{4-4}$$

Second, the DTs of (4-4) are derived in (4-5), using the transformation rules. Specifically, the nonlinear power flow equation (4-4)-c is converted to a new set of equations (4-5)-c that couples the power series coefficients of $\boldsymbol{y}(t)$ and $\lambda(t)$ in all orders, i.e., $\boldsymbol{Y}(0)\ldots \boldsymbol{Y}(k)$, $\Lambda(0)\ldots \Lambda(k)$.

$$\begin{aligned} (k+1)X(k+1) &= F(X(0:k), \boldsymbol{Y}(0:k), \mathrm{L}(0:k)) && (a) \\ 0 &= H(X(0:k), \boldsymbol{Y}(0:k), \mathrm{L}(0:k)) && (b) \\ 0 &= \boldsymbol{G}(\boldsymbol{Y}(0:k), \mathrm{L}(0:k)) && (c) \end{aligned} \tag{4-5}$$

Third, we prove that both (4-5)-c and (4-5)-b satisfy formally linear equations about the $k^{\text{th}}$ order power series coefficients $\boldsymbol{Y}(k)$ and $\Lambda(k)$, as shown in (4-6)-a and (4-6)-b, respectively, where $\boldsymbol{A}$ matrices are functions of $\boldsymbol{Y}(0)$ and $\Lambda(0)$ and $\boldsymbol{B}$ matrices are functions of $\boldsymbol{Y}(0{:}k\text{-}1)$ and $\Lambda(0{:}k\text{-}1)$. As a result, $\boldsymbol{Y}(k)$, i.e., the $k^{\text{th}}$ order power series coefficient of bus voltage vector, is analytical solved from $\boldsymbol{Y}(0{:}k\text{-}1)$ and $\Lambda(0{:}k\text{-}1)$, either by (4-7) or by (4-8), depending on if the ancillary function $h$ is needed when designing the differential equation in (4-4).

$$\begin{aligned} 0 &= \boldsymbol{A}_{gy}\boldsymbol{Y}(k) + \boldsymbol{A}_{gl}\mathrm{L}(k) + \boldsymbol{B}_g && (a) \\ 0 &= \boldsymbol{A}_{hy}\boldsymbol{Y}(k) + \boldsymbol{A}_{hl}\mathrm{L}(k) + \boldsymbol{B}_h && (b) \end{aligned} \tag{4-6}$$

$$\boldsymbol{Y}(k) = -\boldsymbol{A}_{gy}^{-1}(\boldsymbol{A}_{gl}\mathrm{L}(k) + \boldsymbol{B}_g) \tag{4-7}$$

$$\begin{bmatrix} \boldsymbol{Y}(k) \\ \Lambda(k) \end{bmatrix} = -\begin{bmatrix} \boldsymbol{A}_{gy} & \boldsymbol{A}_{gl} \\ \boldsymbol{A}_{hy} & \boldsymbol{A}_{hl} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{B}_{g} \\ \boldsymbol{B}_{h} \end{bmatrix} \tag{4-8}$$

Finally, we design a non-iterative algorithm based on (4-5)-a and (4-7) or (4-8)to solve power series coefficients $X(k)$, $\boldsymbol{Y}(k)$ and $\Lambda(k)$ from $k$=0 to any order $K$ in a recursively manner, and approximate variables $x(t)$, $\boldsymbol{y}(t)$ and $\lambda(t)$ as power series of time, shown in (4-9). After $\boldsymbol{y}(t)$ and $\lambda(t)$ are solved, the solution curves of power flow equations are directly obtained, as illustrated in the case study.

$$\begin{aligned} x(t) &= X(0) + X(1)t + X(2)t^2 + \ldots X(K)t^K \\ \boldsymbol{y}(t) &= \boldsymbol{Y}(0) + \boldsymbol{Y}(1)t + \boldsymbol{Y}(2)t^2 + \ldots \boldsymbol{Y}(K)t^K \\ l(t) &= \mathrm{L}(0) + \mathrm{L}(1)t + \mathrm{L}(2)t^2 + \ldots \mathrm{L}(K)t^K \end{aligned} \tag{4-9}$$

Among the above four steps, only the last step needs to be performed online, while the first three steps can be conducted in the offline stage because they are mainly used to derive expressions of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in (4-6) and function $F$ in (4-5)-a, which is a one-time effort.

**Remarks**: there are two important observations: 1) from (4-6)-a that the nonlinear power flow equation (4-2) about $\boldsymbol{y}(t)$ are converted to a formally linear equation about power series coefficients $\boldsymbol{Y}(k)$ after DT; 2) coefficients on bus voltages are explicitly solved by (4-7) or (4-8) and then used to calculate bus voltages by (4-9) in a straightforward manner, which is different from using a conventional power flow solver to calculate bus voltages by numerical iterations. The proposed DT based method for solving solution curves of power flow equations differentiates itself from the traditional continuation power flow method that relies on iterative numerical methods such as the family of Newton Raphson methods.

### 4.2.2 Step 1: Dynamized Power Flow Equation

Two formulations of (4-4) are proposed to dynamize the power flow equation (4-2), shown in (4-10) and (4-11) respectively, where $C_1$ and $C_2$ are constants and $v_l(t)$ is the voltage magnitude of a load bus $l$. In (4-10), there is no ancillary equation (8b) because the selected state variable $\lambda(t)$ has existed in (4-2). In (4-11), the ancillary equation gives the relationship between bus voltage magnitude and the rectangular coordinate components.

Formulation 1:

$$\begin{aligned} \dot{l}(t) &= C_1 \\ 0 &= \boldsymbol{g}(\boldsymbol{y}(t), l(t)), i.e., Equ.(2) \end{aligned} \tag{4-10}$$

Formulation 2:

$$\begin{aligned} \dot{v}_l(t) &= C_2 \\ 0 &= v_l(t)^2 - e_l(t)^2 - f_l(t)^2 \\ 0 &= \boldsymbol{g}(\boldsymbol{y}(t), l(t)), i.e., Equ.(2) \end{aligned} \tag{4-11}$$

For Formulation 1, its purpose is to characterize how the power changes with time, i.e., the power increases with time when $C_1>0$ and decreases with time when $C_1<0$. It can be used to trace curve segment in various shapes, either monotonically or non-monotonically, such as the curves (a), (b) and (c) in Fig. 19. For Formulation 2, its purpose is to characterize how the voltage magnitude changes with time, i.e., the voltage magnitude increases with time when $C_2>0$ and decreases with time when $C_2<0$. It can also be used to trace either monotonical or non-monotonical curve segments such as (a), (b) and (d) in Fig. 19.

The above two formulations can be flexibly used to trace the full solution curve of a power flow equation. For example, the high voltage solutions in a P-V curve can be traced by Formulation 1 with $C_1>0$, the low voltage solutions can be traced by Formulation 1 with $C_1<0$, and the solution curves near the nose point can be traced by Formulation 2 with $C_2<0$.

### 4.2.3. Step 2: Deriving Different Transformation

1) DTs of Nonlinear Power Flow Equation

The nonlinear power flow equation (4-2) is written into (4-12)-(4-15) under rectangular coordinates, where $\Omega_{PQ}$, $\Omega_{PV}$, $\Omega_{REF}$ are the set of PQ buses, PV buses and reference bus respectively, $p$ and $q$ are active and reactive power, $e$ and $f$ are the real and imaginary parts of bus voltages, $g$ and $b$ are real and imaginary parts of the admittance, $v$ is the voltage magnitude, superscript $sp$ means the value is specified, subscript $i$ and $j$ are the index of buses.

$$p_i^{sp} = g_p(\boldsymbol{y}, l) = -l\,\Delta p_i + \sum_{j=1}^{N} g_{ij}\left(e_i e_j + f_i f_j\right) + \sum_{j=1}^{N} b_{ij}\left(f_i e_j - e_i f_j\right) \quad \text{if } i \in \Omega_{PQ} \cup \Omega_{PV}. \tag{4-12}$$

$$q_i^{sp} = g_q(\boldsymbol{y}, l) = -l\,\Delta q_i - \sum_{j=1}^{N} b_{ij}\left(e_i e_j + f_i f_j\right) + \sum_{j=1}^{N} g_{ij}\left(f_i e_j - e_i f_j\right) \quad \text{if } i \in \Omega_{PQ}. \tag{4-13}$$

$$(v_i^{sp})^2 = g_v(\boldsymbol{y}) = e_i^2 + f_i^2, \text{if } i \in \Omega_{PV} \tag{4-14}$$

$$\begin{aligned} e_i^{sp} &= g_e(\boldsymbol{y}) = e_i, \text{if } i \in \Omega_{REF} \\ f_i^{sp} &= g_f(\boldsymbol{y}) = f_i, \text{if } i \in \Omega_{REF} \end{aligned} \tag{4-15}$$

The DTs of (4-12)-(4-15) are derived in (4-16)-(4-19), respectively.

$$
\begin{aligned}
p_i^{sp} d(k) &= G_p(\boldsymbol{Y}, \mathrm{L}) \\
&= -\mathrm{D}p_i \mathrm{L}(k) + \sum_{j=1}^{N} g_{ij}\left(E_i(k)\,\mathrm{Ä}\,E_j(k) + F_i(k)\,\mathrm{Ä}\,F_j(k)\right) \\
&\quad + \sum_{j=1}^{N} b_{ij}\left(F_i(k)\,\mathrm{Ä}\,E_j(k) - E_i(k)\,\mathrm{Ä}\,F_j(k)\right) \\
&\quad \text{if } i\ \hat{\mathrm{I}}\ \mathrm{W}_{PQ}\ \grave{\mathrm{E}}\ \mathrm{W}_{PV}
\end{aligned} \tag{4-16}
$$

$$
\begin{aligned}
q_i^{sp} d(k) &= G_q(\boldsymbol{Y}, \mathrm{L}) \\
&= -\mathrm{D}q_i \mathrm{L}(k) - \sum_{j=1}^{N} b_{ij}\left(E_i(k)\,\mathrm{Ä}\,E_j(k) + F_i(k)\,\mathrm{Ä}\,F_j(k)\right) \\
&\quad + \sum_{j=1}^{N} g_{ij}\left(F_i(k)\,\mathrm{Ä}\,E_j(k) - E_i(k)\,\mathrm{Ä}\,F_j(k)\right) \\
&\quad \text{if } i\ \hat{\mathrm{I}}\ \mathrm{W}_{PQ}
\end{aligned} \tag{4-17}
$$

$$
\begin{aligned}
(v_i^{sp})^2 d(k) &= G_v(\boldsymbol{Y}) \\
&= E_i(k)\,\mathrm{Ä}\,E_i(k) + F_i(k)\,\mathrm{Ä}\,F_i(k), \text{ if } i\ \hat{\mathrm{I}}\ \mathrm{W}_{PV}
\end{aligned} \tag{4-18}
$$

$$
\begin{aligned}
e_i^{sp} d(k) &= G_e(\boldsymbol{Y}) = E_i(k), \text{ if } i\ \hat{\mathrm{I}}\ \mathrm{W}_{REF} \\
f_i^{sp} d(k) &= G_f(\boldsymbol{Y}) = F_i(k), \text{ if } i\ \hat{\mathrm{I}}\ \mathrm{W}_{REF}
\end{aligned} \tag{4-19}
$$

2) DTs of the Designed Differential Equations

For the differential equation in **Formulation** 1, i.e., (4-10), its DT is in (4-20).

$$\mathrm{L}(k) = C_1 d(k-1) \tag{4-20}$$

For **Formulation** 2 in (4-11), the DT of the differential equation is in (4-21); the DT of the ancillary equation is in (4-22).

$$V_l(k) = -C_2 d(k-1) \tag{4-21}$$

$$V_l(k)\,\mathrm{Ä}\,V_l(k) = E_l(k)\,\mathrm{Ä}\,E_l(k) + F_l(k)\,\mathrm{Ä}\,F_l(k) \tag{4-22}$$

3) Proof of Formal Linearity of Nonlinear Power Flow Equation after DT

**Proposition 3**: The transformed power flow equations (4-16)-(4-19) respectively satisfy formally linear equations (4-23)-(4-26).

$$0 = \boldsymbol{a}_{\mathrm{P},i}\boldsymbol{Y}(k) - \mathrm{D}p_i \mathrm{L}(k) + e_i, \text{ if } i\ \hat{\mathrm{I}}\ \mathrm{W}_{PQ}\ \grave{\mathrm{E}}\ \mathrm{W}_{PV} \tag{4-23}$$

$$0 = \boldsymbol{a}_{\mathrm{Q},i}\boldsymbol{Y}(k) - \mathrm{D}q_i \mathrm{L}(k) + m_i, \text{ if } i\ \hat{\mathrm{I}}\ \mathrm{W}_{PQ} \tag{4-24}$$

$$0 = \boldsymbol{a}_{\mathrm{V},i}\boldsymbol{Y}(k) + 0\mathrm{L}(k) + V_i, \text{ if } i \in W_{PV} \tag{4-25}$$

$$\begin{aligned} 0 &= \boldsymbol{a}_{\mathrm{E},i}\boldsymbol{Y}(k) + 0\mathrm{L}(k) - e_i^{sp} d(k), \text{ if } i \in W_{REF} \\ 0 &= \boldsymbol{a}_{\mathrm{F},i}\boldsymbol{Y}(k) + 0\mathrm{L}(k) - f_i^{sp} d(k), \text{ if } i \in W_{REF} \end{aligned} \tag{4-26}$$

where $\boldsymbol{Y}(k) \in \mathbb{R}^{2N+1}$ and $\Lambda(k) \in \mathbb{R}$ are variables representing the DT of $\boldsymbol{y}$ and $\lambda$ respectively; $\boldsymbol{a}_{\mathrm{P},i}$, $\boldsymbol{a}_{\mathrm{Q},i}$, $\boldsymbol{a}_{\mathrm{V},i}$ $\boldsymbol{a}_{\mathrm{E},i}$, $\boldsymbol{a}_{\mathrm{F},i} \in \mathbb{R}^{1\times 2N}$ and $\varepsilon_i$, $\mu_i$, $\zeta_i \in \mathbb{R}$ are parameters given in (4-29)-(4-34) respectively. The detailed proof of Proposition 3 is below.

To make the proofs more compact, the following Lemma is first proved. In the Lemma, the transformation of multiplication operation from time domain to the convolution operation in the domain of power series orders is well-known in many DT literatures, however, the resulted linear relationship in (4-27)-(4-28), despite their simplicity and being straightforward, are rarely noticed and exploited as far as the authors know.

**Lemma:** The DT of $z(t)=x(t)y(t)$, satisfies a formally linear equation in (4-27). Especially, when $x(t)=y(t)$, (4-28) holds.

$$Z(k) = X(k) \otimes Y(k) = aX(k) + bY(k) + c \tag{4-27}$$

$$Z(k) = X(k) \otimes X(k) = 2aX(k) + c \tag{4-28}$$

Proof of Lemma:

$$\begin{aligned} Z(k) &= X(k) \otimes Y(k) = \sum_{m=0}^{k} X(m)Y(k-m) \\ &= X(0)Y(k) + X(k)Y(0) + \sum_{m=1}^{k-1} X(m)Y(k-m) \end{aligned}$$

Therefore, (4-27) holds with $a$, $b$ and $c$ given below.

$$a = Y(0), b = X(0), c = \sum_{m=1}^{k-1} X(m)Y(k-m)$$

**Proof of Proposition 3:** Use (4-23) as an example. The RHS of (4-16) is rewritten as

$$\begin{aligned}
\text{RHS} = & -\Delta p_i L(k) + \underbrace{g_{ii}\left(E_i(k) \ddot{A} E_i(k) + F_i(k) \ddot{A} F_i(k)\right)}_{\text{Term 1}} \\
& + \underbrace{\sum_{\substack{j=1 \\ j \neq i}}^{N} g_{ij}\left(E_i(k) \ddot{A} E_j(k) + F_i(k) \ddot{A} F_j(k)\right)}_{\text{Term 2}} \\
& + \underbrace{\sum_{j=1}^{N} b_{ij}\left(F_i(k) \ddot{A} E_j(k) - E_i(k) \ddot{A} F_j(k)\right)}_{\text{Term 3}}
\end{aligned}$$

According to the Lemma, the three terms are rewritten as:

$$\begin{aligned}
\text{Term 1} = & \; 2g_{ii}E_i(0)E_i(k) + 2g_{ii}F_i(0)F_i(k) \\
& + g_{ii}\sum_{m=1}^{k-1} E_i(m)E_i(k-m) + g_{ii}\sum_{m=1}^{k-1} F_i(m)F_i(k-m)
\end{aligned}$$

$$\begin{aligned}
\text{Term 2} = & \sum_{\substack{j=1, \\ j \neq i}}^{N} g_{ij}\left(E_j(0)E_i(k) + E_i(0)E_j(k)\right) + \sum_{\substack{j=1 \\ j \neq i}}^{N} g_{ij}\left(F_j(0)F_i(k) + F_i(0)F_j(k)\right) \\
& + \sum_{\substack{j=1 \\ j \neq i}}^{N} g_{ij}\left(\sum_{m=1}^{k-1} E_i(m)E_j(k-m) + \sum_{m=1}^{k-1} F_i(m)F_j(k-m)\right)
\end{aligned}$$

$$\begin{aligned}
\text{Term 3} = & \sum_{j=1}^{N} b_{ij}\left(E_j(0)F_i(k) + F_i(0)E_j(k)\right) - \sum_{j=1}^{N} b_{ij}\left(F_j(0)E_i(k) + E_i(0)F_j(k)\right) \\
& + \sum_{j=1}^{N} b_{ij}\left(\sum_{m=1}^{k-1} F_i(m)E_j(k-m) - \sum_{m=1}^{k-1} E_i(m)F_j(k-m)\right)
\end{aligned}$$

Finally, (4-23) is obtained by summating the above three terms, with vector $\boldsymbol{a}_{\mathrm{P},i}$ and parameter $\varepsilon_i$ in (4-29) and (4-34). Similarly, (4-24)-(4-26) can be proved with vectors $\boldsymbol{a}_{\mathrm{P},i}$, $\boldsymbol{a}_{\mathrm{Q},i}$, $\boldsymbol{a}_{\mathrm{V},i}$, $\boldsymbol{a}_{\mathrm{E},i}$, $\boldsymbol{a}_{\mathrm{F},I}$ and parameters $\varepsilon_i$, $\mu_i$, $\zeta_i$ in (4-29)-(4-36).

$$\boldsymbol{a}_{\mathrm{P},i} = \begin{bmatrix} a_{i1} & b_{i1} & \mathrm{L} & a_{ij} & b_{ij} & \mathrm{L} \end{bmatrix} \text{ where}$$

$$a_{ij} = g_{ij}E_i(0) + b_{ij}F_i(0),\ b_{ij} = g_{ij}F_i(0) - b_{ij}E_i(0), \text{if } j \neq i$$

$$a_{ii} = \sum_{j=1}^{N} \left(g_{ij}E_j(0) - b_{ij}F_j(0)\right) + g_{ii}E_i(0) + b_{ii}F_i(0) \tag{4-29}$$

$$b_{ii} = \sum_{j=1}^{N} \left(b_{ij}E_j(0) + g_{ij}F_j(0)\right) - b_{ii}E_i(0) + g_{ii}F_i(0)$$

$$\boldsymbol{a}_{\mathrm{Q},i} = \begin{bmatrix} f_{i1} & y_{i1} & \mathrm{L} & f_{ij} & y_{ij} & \mathrm{L} \end{bmatrix} \text{ where}$$

$$f_{ij} = -b_{ij}E_i(0) + g_{ij}F_i(0),\ y_{ij} = -b_{ij}F_i(0) - g_{ij}E_i(0), \text{if } j \neq i$$

$$f_{ii} = -\sum_{j=1}^{N} \left(b_{ij}E_j(0) + g_{ij}F_j(0)\right) - b_{ii}E_i(0) + g_{ii}F_i(0) \qquad \text{.....} \tag{4-30}$$

$$y_{ii} = \sum_{j=1}^{N} \left(g_{ij}E_j(0) - b_{ij}F_j(0)\right) - g_{ii}E_i(0) - b_{ii}F_i(0)$$

$$\boldsymbol{a}_{\mathrm{V},i} = \begin{bmatrix} 0 & \mathrm{L} & 0 & 2E_i(0) & 2F_i(0) & 0 & \mathrm{L} & 0 \end{bmatrix} \tag{4-31}$$

$$\boldsymbol{a}_{\mathrm{E},i} = \begin{bmatrix} 0 & \mathrm{L} & 0 & 1 & 0 & 0 & \mathrm{L} & 0 \end{bmatrix} \tag{4-32}$$

$$\boldsymbol{a}_{\mathrm{F},i} = \begin{bmatrix} 0 & \mathrm{L} & 0 & 0 & 1 & 0 & \mathrm{L} & 0 \end{bmatrix} \tag{4-33}$$

$$e_i = \sum_{j=1}^{N} g_{ij}c_{ij} + \sum_{j=1}^{N} b_{ij}d_{ij} - p_i d(k), \text{where}$$

$$c_{ij} := \sum_{m=1}^{k-1} E_i(m)E_j(k-m) + \sum_{m=1}^{k-1} F_i(m)F_j(k-m) \tag{4-34}$$

$$d_{ij} := \sum_{m=1}^{k-1} F_i(m)E_j(k-m) - \sum_{m=1}^{k-1} E_i(m)F_j(k-m)$$

$$m_i = -\sum_{j=1}^{N} b_{ij}c_{ij} + \sum_{j=1}^{N} g_{ij}d_{ij} - q_i d(k) \tag{4-35}$$

$$V_i = c_{ii} - v_i^2 d(k) \tag{4-36}$$

Proof of (4-6)-b from (4-22): From the Lemma, there are

$$E_l(k) \ddot{A} E_l(k) = 2E_l(0)E_l(k) + \sum_{m=1}^{k-1} E_l(m)E_l(k-m)$$

$$F_l(k) \ddot{A} F_l(k) = 2F_l(0)F_l(k) + \sum_{m=1}^{k-1} F_l(m)F_l(k-m)$$

Then, (4-22) is rewritten as:

$$\begin{aligned} V_l(k) \ddot{A} V_l(k) = {} & 2E_l(0)E_l(k) + 2F_l(0)F_l(k) \\ & + \sum_{m=1}^{k-1} E_l(m)E_l(k-m) + \sum_{m=1}^{k-1} F_l(m)F_l(k-m) \end{aligned}$$

Finally, (4-6)-b is obtained with $\boldsymbol{a}_l$ and $\xi_l$ given in (4-37).

$$\begin{aligned} \boldsymbol{a}_l &= \begin{bmatrix} 0 & \cdots & 0 & 2E_l(0) & 2F_l(0) & 0 & \cdots & 0 \end{bmatrix} \\ x_l &= \sum_{m=1}^{k-1} E_l(m)E_l(k-m) + \sum_{m=1}^{k-1} F_l(m)F_l(k-m) - V_l(k) \ddot{A} V_l(k) \end{aligned} \tag{4-37}$$

From the **Proposition**, DTs (4-5)-c of the nonlinear power flow equation satisfy formally linear equation (4-6)-a with matrices $\boldsymbol{A}_{gy}$, $\mathbf{A}_{g\lambda}$, and $\boldsymbol{B}_g$ given by (4-38). For notation simplicity, here we let buses 1 to $M$ be PQ buses, buses $M$+1 to $N$-1 be PV buses and bus $N$ be the reference bus.

$$\boldsymbol{A}_{gy} = \begin{bmatrix} \boldsymbol{A}_{y,\mathrm{PQ}} \\ \boldsymbol{A}_{y,\mathrm{PV}} \\ \boldsymbol{A}_{y,\mathrm{REF}} \end{bmatrix} \quad \boldsymbol{A}_{gl} = \begin{bmatrix} \boldsymbol{A}_{l,\mathrm{PQ}} \\ \boldsymbol{A}_{l,\mathrm{PV}} \\ \boldsymbol{A}_{l,\mathrm{REF}} \end{bmatrix} \quad \boldsymbol{B}_g = \begin{bmatrix} \boldsymbol{B}_{\mathrm{PQ}} \\ \boldsymbol{B}_{\mathrm{PV}} \\ \boldsymbol{B}_{\mathrm{REF}} \end{bmatrix} \tag{4-38}$$

$$\boldsymbol{A}_{y,\mathrm{PQ}} = \begin{bmatrix} \boldsymbol{a}_{\mathrm{P},1} \\ \boldsymbol{a}_{\mathrm{Q},1} \\ \vdots \\ \boldsymbol{a}_{\mathrm{P},M} \\ \boldsymbol{a}_{\mathrm{Q},M} \end{bmatrix} \quad \boldsymbol{A}_{l,\mathrm{PQ}} = -\begin{bmatrix} \Delta p_1 \\ \Delta q_1 \\ \vdots \\ \Delta p_M \\ \Delta q_M \end{bmatrix} \quad \boldsymbol{B}_{\mathrm{PQ}} = \begin{bmatrix} e_1 \\ m_1 \\ \vdots \\ e_M \\ m_M \end{bmatrix}$$

$$\boldsymbol{A}_{y,\mathrm{PV}} = \begin{bmatrix} \boldsymbol{a}_{\mathrm{P},M+1} \\ \boldsymbol{a}_{\mathrm{V},M+1} \\ \vdots \\ \boldsymbol{a}_{\mathrm{P},N-1} \\ \boldsymbol{a}_{\mathrm{V},N-1} \end{bmatrix} \quad \boldsymbol{A}_{l,\mathrm{PV}} = -\begin{bmatrix} \Delta p_{M+1} \\ 0 \\ \vdots \\ \Delta p_{N-1} \\ 0 \end{bmatrix} \quad \boldsymbol{B}_{\mathrm{PV}} = \begin{bmatrix} e_{M+1} \\ z_{M+1} \\ \vdots \\ e_{N-1} \\ z_{N-1} \end{bmatrix}$$

$$\boldsymbol{A}_{y,\mathrm{REF}} = \begin{bmatrix} \boldsymbol{a}_{\mathrm{E},N} \\ \boldsymbol{a}_{\mathrm{F},N} \end{bmatrix} \quad \boldsymbol{A}_{l,\mathrm{REF}} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{B}_{\mathrm{REF}} = \begin{bmatrix} -e_N^{sp} d(k) \\ -f_N^{sp} d(k) \end{bmatrix}$$

Besides, the DT (4-22) of the ancillary equation in Formulation II also satisfies a formally linear equation in (4-6)-b with proof in the Appendix.

### 4.2.4. Step 3: Recursively Solving Variables As Power Series Of Time

Following the basic idea in Chapter 4.2.1, two algorithms are designed to solve power series coefficients $X(k)$, $\Lambda(k)$, $\boldsymbol{Y}(k)$ up to any desired order, as shown by Algorithm 3 and Algorithm 4, using Formulation I and Formulation II respectively. Note that these coefficients are explicitly calculated with no numerical iteration.

After the power series coefficients are calculated, $\boldsymbol{y}(t)$ and $\lambda(t)$ are calculated by evaluating the power series of time in (4-9) and the solution curves are directly obtained. In practical, the multi-time window strategy can be used to extend the convergence region of power series of time and ensure the accuracy. The time step length as well as the order $K$ of the power series of time are usually selected from trial simulations, and the impact of $K$ and time step length are also studied in Chapter 2 and 3.

| **Algorithm 3:** Solve Coefficients Using Formulation I | |
|---|---|
| **Input** | Initial values $x(0), \boldsymbol{y}(0), l(0)$ |
| **Output** | Any order coefficients $X(k), \boldsymbol{Y}(k), \Lambda(k), k = 0 \ldots K$ |
| 1 | Initialization: $X(0) = x(0), \boldsymbol{Y}(0) = \boldsymbol{y}(0), \Lambda(0) = l(0)$ |
| 2 | Calculate matrices $\boldsymbol{A}_{gy}$ using (4-38) |

| | |
|---|---|
| 3 | Calculate $\boldsymbol{A}_{gy}^{-1}$ |
| 4 | **for** $k = 1 : K$ **do** |
| 5 | Calculate matrix $\boldsymbol{B_g}$ using (4-38) |
| 6 | Solve $\mathrm{L}(k)$ using $\mathrm{L}(k) = C_1 d(k-1)$ |
| 7 | Solve $\boldsymbol{Y}(k)$ using $\boldsymbol{Y}(k) = -\boldsymbol{A}_{gy}^{-1}(\boldsymbol{A}_{gl}\mathrm{L}(k) + \boldsymbol{B}_g)$ |
| 8 | **end for** |

| | |
|---|---|
| **Algorithm 4:** Solve Coefficients Using Formulation II | |
| **Input** | Initial values $x(0), \boldsymbol{y}(0), l(0)$ |
| **Output** | Any order coefficients $X(k), \boldsymbol{Y}(k), \mathrm{L}(k), k = 0 \ldots K$ |
| 1 | Initialization: $X(0) = x(0), \boldsymbol{Y}(0) = \boldsymbol{y}(0), \mathrm{L}(0) = l(0)$ |
| 2 | Calculate matrices $\boldsymbol{A_{gy}}, \boldsymbol{A_{g\lambda}}, \boldsymbol{A_{hy}}, \boldsymbol{A_{h\lambda}}$ using (4-38) |
| 3 | Calculate $\begin{bmatrix} \boldsymbol{A}_{hy} & \boldsymbol{A}_{hl} \\ \boldsymbol{A}_{gy} & \boldsymbol{A}_{gl} \end{bmatrix}^{-1}$ |
| 4 | **for** $k = 1 : K$ **do** |
| 5 | Calculate matrix $\boldsymbol{B_g}, \boldsymbol{B_h}$ using (4-38) |
| 6 | Solve $V_l(k)$ using $V_l(k) = -C_2 d(k-1)$ |
| 7 | Solve $\boldsymbol{Y}(k), \mathrm{L}(k)$ using $\begin{bmatrix} \boldsymbol{Y}(k) \\ \mathrm{L}(k) \end{bmatrix} = -\begin{bmatrix} \boldsymbol{A}_{gy} & \boldsymbol{A}_{gl} \\ \boldsymbol{A}_{hy} & \boldsymbol{A}_{hl} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{B}_g \\ \boldsymbol{B}_h \end{bmatrix}$ |
| 8 | **end for** |

The proposed method can also be applied to more complicated power system models such as 1) considering reactive power limit of generators, 2) ZIP load model. First, to consider the reactive power limit of generators, the proposed method can be slightly modified as follows: if a generator meets the reactive power limit, then it is changed from a PV bus to a PQ bus; correspondingly, the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ need to be re-calculated using (4-38). Later in the case study, we demonstrate the proposed method for tracing PV curves considering reactive power limits. Second, for a nonlinear power flow

equation with ZIP loads, we proved in [47] that its DTs still satisfy formally linear equations. Therefore, the proposed method can be directly applied with slight modification on matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

Regarding the computational complexity, the proposed method has two unique features: First, it shifts most of the computation burden to the offline stage, i.e., deriving the equation for calculating matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, which is a one-time effort (the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ derived in this chapter can be directly used by others without deriving them again); and the online stage only involves explicit matrix operation and evaluation of analytical solutions, which do not require any numerical iterations. Second, the proposed method can reduce the frequency of solving linear equations compared with the CPF method, thus having better computational efficiency. This is because the CPF method needs to solve a linear equation in every iteration and every prediction-correction step, while the proposed method only needs to solve linear equations once in each time step, and the total number of time steps are greatly reduced benefiting from the high order approximation.

## 4.3 Case Studies

The proposed DPF method is first tested on the IEEE 9-bus system to demonstrate the basic idea, the impact of load change directions, and the impact of reactive power limit of generators. Then, the accuracy, computational complexity, and computation time are compared with the CPF method in MATPOWER using several large systems including the IEEE 39-bus system, IEEE 57-bus system and a Polish 2383-bus test system [60]. At last, the proposed approach is applied to $N$-1 contingency analysis.

Simulations are conducted in MATLAB R2017a on a personal computer with i5-8250U CPU. Without specification, generations and loads of all buses are uniformly increased. For the CPF method, various simulation control parameters are adjusted for the best time performance, including using the pseudo arc-length for parameterization, enabling adaptive step size, increasing the maximum allowed step size and disabling the incremental curve plotting in each iteration, etc. For the DPF method, parameters $C_1$ and $C_2$ are set as 1, $K$ is set as 6 from trail simulations, and the time step length is fixed at 0.05s for 2383-bus system and 0.1s for other systems.

**4.3.1. Demonstration on the 9-bus Power System**

To demonstrate the idea of the proposed method, Fig. 20 and Fig. 21 respectively give the time domain trajectories of the solved dynamized power flow model and the obtained PV curve. In the first 1.63s, the loading parameter λ increases with time in a constant rate while the voltage magnitude of bus 9 drops from 0.9956 p.u. to 0.6268 p.u., indicating high voltage solutions. During the time period between t=1.63s and t=1.68s, the voltage is decreased from 0.6268 p.u. to 0.5439 p.u., while the loading parameter is first increased from 1.63 to reach the maximum value 1.64 and then decreased to 1.63, indicating the dynamic process of passing the nose point. Finally, both the loading parameter and the bus voltage are decreased after t=1.68s, indicating the low voltage solutions. The obtained loading limit 1.64 is the same as the limit from the CPF method.

Two scenarios are designed to demonstrate the capability of the proposed method on handling load changes with 1) non-uniform directions and 2) reactive power limits. Fig. 22a shows the PV curve of load bus 9 when increasing generation at bus 3 and load

at bus 7 by 50 MW in active power and 10 MVar in reactive power. Fig. 22b further shows the PV curve of the same bus when reactive power limit of generators is considered. It shows the calculated maximum loading limits are reduced from 8.17 to 7.79 due to the reactive power limit. These results demonstrate the performance of the proposed method on practical power system models and applications.

### 4.3.2. Accuracy, Computation Complexity, And Time Performance On Large Systems

Respectively for the 39-bus system, the 57-bus system and the 2383-bus system, the proposed DPF method is compared with the CPF method. In all following studies, the CPF method is tested using the commercial MATPOWER package while the proposed DPF method is tested using our research code. Fig. 23 to Fig. 25 show the PV curves of three load buses, obtained by both the proposed method and the CPF method. Respectively for the three test systems, the calculated loading limits are 1.12, 0.88 and 0.89 for DPF method, and 1.13, 0.89 and 0.89 for the CPF method. These results demonstrate the accuracy of the proposed method.

For both the CPF method and the DPF method, a major computation burden is in solving linear equations. Table 8 gives how many times linear equations are solved for both methods. It shows that the proposed approach is 10 times fewer than the CPF method for all the three test systems. This is because the CPF method needs to solve a linear equation in each iteration and for every prediction-correction step while the proposed method only solves a linear equation once in each time step.

Table 9 further gives the computation times of both methods. It shows the proposed DPF method is around 9 times, 12 times, and 2 times faster than the CPF method, respectively, for the three test systems. The speed up on the 2383-bus system is less than speedups on the other two smaller systems because our current academic research code that implements the DPF method in MATLAB has not been optimized to as efficiently handle large-scale matrix operations as the commercial CPF solver in the MATPOWER. However, these test results do demonstrate the potential of the proposed DPF method for online power flow solution tracing and voltage stability assessment.

### 4.3.3. Application To N-1 Contingency Analysis

The proposed approach is further applied to screen *N*-1 contingencies. For the 39-bus system, 46 contingencies are created each with the loss of each single branch. Fig. 26 shows the maximum loading condition identified by both methods. Using the CPF results as benchmarks, the DPF method is accurate and reliable for all the contingencies. Regarding the computation time, the CPF method and the DPF method respectively takes 12.0 s and 1.4 s, showing that the DPF method can identify insecure contingencies much faster than the CPF method, and thus can scan more contingencies than the CPF method within limited time in the real-time environment.

## 4.4 Conclusion

In this chapter, a novel dynamized power flow method has been proposed to efficiently trace solution curves of power flow equations. The original curve tracing problem for steady-state power flow solutions is converted to a time domain simulation problem about a dynamized model after adding a differential equation on changes of the

operating condition. An DT-based approach is proposed for efficiently solving the dynamized model without numerical iterations. Simulation results have shown high accuracy, reduced computational complexity and improved time performance of the proposed DPF method compared with a CPF solver in MATPOWER. Besides, the proposed method can deal with practical engineering constraints such as the non-uniform load change directions and reactive power limits of generators.

# CHAPTER 5

# A DT BASED ADAPTIVE PARAREAL METHOD FOR POWER SYSTEM SIMULATION

This chapter proposes a DT- based variable-order variable-step variable-window adaptive parareal method for temporal parallelization of power system simulation with greatly enhanced convergence performance and efficiency. The proposed method integrates the temporal parallelization capability of the Parareal method and the highly adaptive feature of the DT method. Extensive simulations on a 39-bus system and a 2383-bus system demonstrate the effectiveness of the proposed approach.

## 5.1 Proposed Method

Consider the power system DAE model below

$$\begin{cases} \dot{x}(t) = f(x(t), v(t)) \\ 0 = g(x(t), v(t)) \end{cases}, \begin{cases} x(0) = x_0 \\ v(0) = v_0 \end{cases} \tag{5-1}$$

where $x$ is the vector of state variables, $v$ is the vector of bus voltages, and $f$ and $g$ are functions in differential equations and algebraic equations, respectively. For the ease of presentation, let $\boldsymbol{x}(t) = [x(t)^T, v(t)^T]^T, \boldsymbol{x}_0 = [x_0^T, v_0^T]^T$ and rewrite the DAE model as $0 = \boldsymbol{f}(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)), \boldsymbol{x}(0) = \boldsymbol{x}_0$.

**Notation**:

$m, M$ index and total number of iterations in Parareal

$k, K$ index and total order of DT

$n, N$ index and total number of time step in coarse solver

$w, c, f$ superscripts of Parareal window, coarse step, fine step

### 5.1.1. Conventional Numerical-based Parareal Method

Fig. 27 illustrates the basic idea of the Parareal method to achieve temporal parallelization. First, the desired simulation length [$t_0$, $t_N$] is decomposed into $N$ coarse steps with length $h_c$, and each coarse interval is further decomposed into many fine steps with length $h_f$. Then, the block of **initial coarse evaluation** computes the initial coarse solutions $\mathbf{x}_n^{c,0}$ by performing the simulation over [$t_0$, $t_N$] in serial using a coarse operator $\mathrm{C}$ with the coarse step length $h_c$. After that, the iteration process is performed between the blocks of fine evaluation and the coarse solution update. The block of **fine evaluation** is to provide a better solution $\mathbf{x}_n^{f,0}$ at each interval by simulating each coarse interval [$t_n$, $t_{n+1}$] in parallel using a fine operator $\mathrm{F}$ with the fine step length $h_f$. The block of **coarse solution update** is to correct the coarse solution at each coarse interval. The iteration process continues until the stopping criteria is met, i.e., the differences of coarse solutions between $m^{th}$ iteration and $(m+1)^{\text{th}}$ iteration are smaller than a pre-defined threshold, or the maximum number of iterations is reached.

Algorithm 5 shows the detailed procedure of the Parareal method, where the lines from 1 to 2 correspond to the block of initial coarse evaluation in Fig. 27; the lines from 4 to 6 correspond to the block of fine evaluation; the lines from 7 to 10 correspond to the coarse solution update; and the lines from 11 to 13 are the stopping criterion. In the algorithm, $\boldsymbol{x}_n^{c,m}, \boldsymbol{x}_n^{f,m}, \boldsymbol{x}_n^{*,m}$ respectively mean the coarse solution, fine solution and corrected coarse solution at time instant $t_n$ in the $m^{\text{th}}$ iteration; *tol* is a pre-defined

threshold; $M$ is the recorded number of iterations for Parareal algorithm to converge; and $m_{max}$ is the pre-defined maximum number of iterations.

**Algorithm 5: Parareal method**

**input**: $t_0, t_N, h_c, h_f, \boldsymbol{x}_0$
**output**: $\boldsymbol{x}_n, n = 1,2,\ldots N;\ M$

```
x_0^{*,0} = x_0, x_0^{*,1} = x_0^{*,0}
x_n^{*,0} = x_n^{c,0} ¬ C(t_0, t_n, h_c, x_0), n = 1,2...N // initial coarse evaluation
    for m = 1 : m_max
        for n = m : N // fine evaluation in parallel
            x_n^{f,m} ¬ F(t_n, t_{n+1}, h_f, x_{n-1}^{*,m-1})
        end
        for n = m : N // coarse solution update in serial
            x_n^{c,m} ¬ C(t_{n-1}, t_n, h_c, x_{n-1}^{*,m})
            x_n^{*,m} ¬ x_n^{c,m} + x_n^{f,m} - x_n^{c,m-1}
        end
        if || x_n^{*,m} - x_n^{*,m-1} ||_1 < tol, "n = 1,2,...N ; // stopping criterion
            x_n ¬ x_n^{*,m}, n = 1,2,...N ; M ¬ m
        end
end
```

The convergence performance of the Parareal algorithm (quantified by the number of iterations) is of critical importance to the overall efficiency of the Parareal algorithm. Many factors could impact the convergence performance of the Parareal algorithm, including 1) the selection of the coarse operator $\mathcal{C}$, 2) the coarse step length $h_c$, 3) the length of a simulation window $h_w = t_N - t_0$, and 4) the coarse solution update strategy. For example, a larger $h_w$ or $h_c$ may require more iterations or cause the Parareal algorithm to diverge. Based on these factors, we propose four corresponding techniques in the following sections to improve the convergence performance of the Parareal

algorithm, as shown in Fig. 28. After that, an integrated approach based on the four techniques is presented.

### 5.1.2. DT-based Parareal Method

The selection of the coarse operator plays a significant role in the convergence performance and computational efficiency of the Parareal algorithm. The coarse operator needs to have sufficient accuracy to ensure the convergence of the Parareal method on the one hand and needs to be fast enough to achieve considerable speedup on the other hand. The latter is because the coarse operator evaluation is in serial, which can offset of the efficiency gain from the parallel computing and should be minimized.

This section proposes a DT-based Parareal method which replaces the coarse solver $\mathcal{C}$ in Algorithm 5 with a DT solver. Fig. 29 illustrates the basic idea of the DT method, which derives semi-analytical solutions of differential equations or differential algebraic equations in the form of high-order power series of time. The DT method provides various transformation rules for numerous generic functions (both linear and nonlinear functions; both simple and compositional functions) such that a differential equation in a continuous set about the variable $t$ (time) is converted to a new set of difference equations in a discrete set about the variable $k$ (the power series order). The derivation of the difference equation is performed in the offline stage, which is a one-time effort. Based on the obtained difference equation, the power series coefficients up to any desired order could be efficiently computed in a recursive manner in the online stage. Using the single- machine- infinite- bus (SMIB) system as an example, the high-order power series solution is derived for illustration, and the rotor angle trajectory obtained

from the power series solution with different orders is compared with the benchmark solution obtained by the RK4 method with a tiny time step length. It shows the convergence region of the DT method increases with the order of the power series solution. Especially, the solution with an order of 15 is accurate enough up to 0.35 seconds, which is more than half period of the rotor angle trajectory.

Algorithm 6 shows the detailed procedure of the DT method, where the offline stage derives a recursive equation about power series coefficients; the line 1 in the online stage is the initialization of variables; the lines from 3 to 5 are the online evaluation of power series coefficients; the line 6 is to compute $\mathbf{x}(t_{n+1})$ by summing the power series terms from $0^{th}$ to $K^{th}$ orders; and the line 7 moves the simulation to the next time step.

**Algorithm 6: DT Method**

**input**: $t_0, t_{end}, \boldsymbol{x}_0, h, K$
**output**: $\boldsymbol{x}(t_n),\ t_0 \text{ £ } t_n \text{ £ } t_{end}$
**offline stage**: derive recursive equation $\boldsymbol{X}(k+1) = \boldsymbol{F}(\boldsymbol{X}(0:k))$
**online stage**:

1. $\boldsymbol{x}(t_0) \neg \boldsymbol{x}_0,\ t \neg t_0,\ \boldsymbol{X}(0) \neg \boldsymbol{x}_0,\ n \neg 0$
2. **while** $t + h < t_{end}$
3. **for** $k = 0 : K$ // computer power series coefficients
4. $\boldsymbol{X}(k+1) = \boldsymbol{F}(\boldsymbol{X}(0:k))$;
5. **end**
6. $\boldsymbol{x}(t_{n+1}) \neg$ sum of $\boldsymbol{X}(k)h^k$ over $k = 0,\ldots K$
7. $t \neg t + h, n \neg n + 1$
8. **end**

The proposed DT-based Parareal method has the following advantages: 1) the DT method can be more efficient than numerical methods such that it can reduce the sequential time and improve the overall speedup, 2) the DT method can be more accurate than numerical methods such that the coarse solution by the DT method is closer to the

true solution and less iterations are needed to make the Parareal algorithm converge, and 3) the DT method is highly flexible such that the order and time step can be adjusted adaptively to balance the accuracy and efficiency of coarse solver during the simulation, thus improving the overall performance.

### 5.1.3. VOVS-DT Method

Algorithm 7 shows the proposed VOVS-DT method, where the outputs include the not only the trajectory of state variables, but also the variable time step lengths $\boldsymbol{h}(t_n)$ and variable orders $\boldsymbol{K}(t_n)$ during the simulation; the lines from 1 to 7 are almost the same with the DT method except for the initialization of additional variables associated with the time step length and the order of DT; the line 8 is the estimation of error using the $(K+1)^{\text{th}}$ power series term; and the line from 9 to 12 are the adaptive change of time step length and order of DT based on the error estimation. In the Algorithm 7, $e_1, e_2, e_3, e_4$ are pre-defined error threshold, which takes the value of $10^{-8}$, $10^{-4}$, $10^{-8}$, and $10^{-4}$ respectively in this chapter; $q_1, q_2$ are the factors to adjust the time step length, which takes the value of 1.2 and 0.8 respectively; $\mathrm{D}K$ is the incremental to adjust the order of DT, which takes the value of 2. $h_{\max}, h_{\min}$ are the pre-defined maximum and minimum time step length, which take the value of 0.2 seconds and 0.0001 seconds respectively; $K_{\max}, K_{\min}$ are the pre-defined maximum and minimum order, which take the value of 20 and 2 respectively.

**Algorithm 7: VOVS-DT Method**

**input**: $t_0, t_{end}, \boldsymbol{x}_0, h_0, K_0$
**output**: $\boldsymbol{x}(t_n), \boldsymbol{h}(t_n), \boldsymbol{K}(t_n),\ t_0 \text{ £ } t_n \text{ £ } t_{end}$
**offline stage**: derive recursive equation $\boldsymbol{X}(k+1) = \boldsymbol{F}(\boldsymbol{X}(0:k))$

**online stage**:

1. $\boldsymbol{x}(t_0) \neg \boldsymbol{x}_0, \boldsymbol{h}(t_0) \neg h_0, \boldsymbol{K}(t_0) \neg K_0$
2. $t \neg t_0, h \neg h_0, K \neg K_0, \boldsymbol{X}(0) \neg \boldsymbol{x}_0, n \neg 0$
3. **while** $t + h < t_{end}$
4. **for** $k = 0 : K$ // computer power series coefficients
5. $\boldsymbol{X}(k+1) = \boldsymbol{F}(\boldsymbol{X}(0:k))$;
6. **end**
7. $\boldsymbol{x}(t_{n+1}) \neg$ sum of $\boldsymbol{X}(k)h^k$ over $k = 0,\ldots K$
8. $err \neg \| \boldsymbol{X}(K+1)h^{K+1} \|_{¥}$ // estimation of error
9. **if** $err < e_1$ ; $h \neg \min(q_1 h, h_{\max})$; **end**
10. **if** $err > e_2$ ; $h \neg \max(q_2 h, h_{\min})$; **end**
11. **if** $err < e_3$ ; $K \neg \max(K - \mathrm{D}K, K_{\min})$; **end**
12. **if** $err > e_4$ ; $K \neg \min(K + \mathrm{D}K, K_{\max})$; **end**
13. $\boldsymbol{h}(t_{n+1}) \neg h, \boldsymbol{K}(t_{n+1}) \neg K$, $t \neg t + h, n \neg n + 1$
14. **end**

The proposed VOVS-DT method has the following advantages when serving as the coarse solver in the Parareal method: 1) it increases the order and reduces the time step length of the DT-based coarse solver when the error is larger than a threshold during the simulation, which can reduce the needed number of iterations for the Parareal method to converge; 2) it decreases the order and increases the time step length of the DT-based coarse solver when the error is smaller than a threshold during the simulation, which could slightly increase the number of iterations but can avoid unnecessary computation burden and improve the overall efficiency.

### 5.1.4. VW-Parareal

To overcome the divergence issue of the Parareal method under long simulation length, the VW-Parareal method is proposed. As shown in Fig. 30, the total simulation length is decomposed into multiple windows and the parareal method is applied to each window in a sequential manner. Moreover, a variable window strategy is proposed that

can change the window size adaptively according to the needed number of iterations in the previous window.

Algorithm 8 shows the detailed procedure of the proposed VW-Parareal method, where the outputs include not only the trajectory of state variables but also the variable window lengths $\mathbf{H}$ and the number of iterations $\mathbf{M}$ in each window during the simulation; line 1 is the initialization of variables; line 3 defines a simulation window; line 4 performs Parareal simulation over this window; line 5 and line 6 adjusts the window length of the next window according to the number of iterations in the current window; line 7 saves the results. In Algorithm 8, $M_1, M_2$ are the pre-defined threshold of the number of iterations, which both take the value of 3 in this paper; $q_3, q_4$ are the factors to adjust the window length, which take the value of 1.2 and 0.5 respectively; $h_{w,\max}, h_{w,\min}$ are the pre-defined maximum and minimum window length, respectively, which take the value of 2 seconds and 0.2 seconds.

**Algorithm 8: VW-Parareal Method**

**input**: $t_0, t_{end}, h_w, \boldsymbol{x}(t_0)$
**output**: $\boldsymbol{x}(t), t_0 \leq t \leq t_{end}; \boldsymbol{H}, \boldsymbol{M}$

1. $t \leftarrow t_0, \mathbf{H} \leftarrow h_w, \boldsymbol{M} \leftarrow \varnothing, \boldsymbol{x}(t) \leftarrow \boldsymbol{x}(t_0)$
2. **while** $t + h_w < t_{end}$
3. $\quad t_1^w \leftarrow t, t_2^w \leftarrow t + h_w$
4. $\quad$ call **Algorithm 5**: Parareal method to compute $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_N; M$
5. $\quad$ **if** $M < M_1$, **then** $h_w \leftarrow \min(q_3 h_w, h_{w,\max})$
6. $\quad$ **if** $M > M_2$, **then** $h_w \leftarrow \max(q_4 h_w, h_{w,\min})$
7. $\quad t \leftarrow t + h_w$, $\boldsymbol{x}(t) \leftarrow [\boldsymbol{x}(t), \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_N]$, $\mathbf{H} \leftarrow [\mathbf{H}, h_w], \boldsymbol{M} \leftarrow [\boldsymbol{M}, M]$
8. **end**

The proposed VW-Parareal method has the following advantages. First, the usage of multi-window strategy could make the Parareal algorithm easier to converge because the length of a window is smaller than the total simulation length. Second, the proposed variable window strategy could provide more suitable window sizes during the simulation than a pre-defined fixed window size, because the Parareal algorithm under a fixed window size may diverge when window size is too large and may be inefficient when the window size is too small. Therefore, the proposed VW-Parareal method is able to balance the convergence performance and efficiency performance.

### 5.1.5. Improved Coarse Update Strategy

The coarse update strategy $\boldsymbol{x}_n^{*,m} \neq \boldsymbol{x}_n^{f,m} + (\boldsymbol{x}_n^{c,m} - \boldsymbol{x}_n^{c,m-1})$ in Algorithm 5 could be interpreted as a first-order Newton iteration method to solve the multi-shooting problem, where the Jacobian matrix in the Newton iteration method is approximated by the coarse solutions [67]. Following the interpretation in [67], this section extends the Parareal method to the second-order Newton iteration method to solve the multi-shooting problem, where both the Jacobian matrix in the first-order term and the Hessian matrix in the second-order term are approximated by the coarse solutions. The procedure is similar with [67] and the details are omitted.

$$\boldsymbol{x}_n^{*,m} \neq \boldsymbol{x}_n^{f,m} + \underbrace{(\boldsymbol{x}_n^{c,m} - \boldsymbol{x}_n^{c,m-1})}_{\text{first-order term}} + \underbrace{0.5(\boldsymbol{x}_n^{c,m+1} - 2\boldsymbol{x}_n^{c,m} + \boldsymbol{x}_n^{c,m-1})}_{\text{second-order term}}$$

Here, the value of $\boldsymbol{x}_n^{c,m+1}$ is unknown in the $m^{\text{th}}$ iteration and is approximated by $\boldsymbol{x}_n^{c,m+1} \approx \boldsymbol{x}_n^{c,m}$, which results in the proposed improved coarse update strategy:

$$\boldsymbol{x}_n^{*,m} \neg \boldsymbol{x}_n^{f,m} + 0.5(\boldsymbol{x}_n^{c,m} - \boldsymbol{x}_n^{c,m-1})$$

The approximation is valid because the differences between $\boldsymbol{x}_n^{c,m+1}$ and $\boldsymbol{x}_n^{c,m}$ decrease with the number of iteration if the Paraeral algorithm is converged. Therefore, this improved coarse update strategy is expected to reduce the number of iterations is the Parareal algorithm is able to converge.

### 5.1.6. Integrated Approach

This section integrates the above four techniques using the two strategies below. 1) Integrated approach 1: a DT-based VW-Parareal method with improved coarse solution update strategy, and 2) Integrated approach 2: a VOVS-DT-based VW-Parareal method with improved coarse solution update strategy. Both approaches could greatly improve the convergence performance of the Parareal algorithm. Compared with approach 1 with a fixed-order fixed-step DT, approach 2 provides additional flexibility to adjust the order and time step of the DT method to improve the overall efficiency.

## 5.2 Case Study on a 39-bus system

This section demonstrates the characteristics of each proposed algorithm (including the DT-based Parareal method, the VW-Parareal method, the Improved coarse update strategy, and the VOVS-DT method) based on extensive simulations on a 39-bus system with classical generator models.

### 5.2.1. Conventional Numerical-based Parareal Method

In this case study, different algorithms are used to simulate the trajectory of a 39-bus system under three different contingencies with increased severity, including 1) a temporary fault at bus 8, cleared after 0.1 seconds, 2) a permanent fault at bus 8, cleared after 0.2 seconds by tripping the branch between bus 8 and bus 9, and 3) a permanent fault at bus 8, cleared after 0.3 seconds by tripping the branch between bus 8 and bus 9.

In this test, the conventional Parareal method with numerical methods are used, where the fine solver is selected as the RK4 method with a tiny step length of 0.001 seconds, and three different coarse solvers are selected using the RK4 method with different time step lengths, including: 0.100 seconds, 0.075 seconds, and 0.050 seconds respectively. For the three different coarse time steps, Table 10 gives the average number of iterations of the conventional numerical-based Parareal algorithm under different window lengths. It shows the conventional Parareal algorithm needs more iterations to converge or even diverge (i.e., reaches a pre-defined maximum number of iterations, set as 10) when the coarse time step length or window length is too large. Also, it becomes harder for the conventional Parareal algorithm to converge when simulating more severe contingencies with stronger transient dynamics.

### 5.2.2. DT-based Parareal Method

In this test, the DT-based Parareal method is used, where the fine solver is selected as the DT method with an order of 8 and a tiny step length of 0.001 seconds, and three different coarse solvers are selected using the DT method with an order of 8 and three different time step lengths, including: 0.100 seconds, 0.075 seconds, and 0.050

seconds respectively. Table 11 shows the average number of iterations using the DT-based Parareal algorithm. In all the scenarios, the PDT method is converged in 1 iteration only. This result demonstrates the effectiveness of the DT-based Parareal method on improving the convergence performance of the Parareal iteration process.

### 5.2.3. VW-Parareal

In this test, the VW-Parareal method based on numerical methods is used. Table12 and Table 13 respectively give the average number of iterations and the average window length of the proposed VW-Parareal algorithm under different coarse time steps and different window lengths. Table 12 shows the proposed VW-Parareal algorithm is converged in all the scenarios with less than 4 iterations averagely. This is because of the flexibility of the proposed VW-Parareal algorithm to adjust the window length adaptively to achieve the convergence, as shown in Table 13. For example, for the cases with $h_c$=0.010 seconds, the conventional fixed window Parareal algorithm with $h_w$=1.00 seconds fail to converge for all three fault scenarios, but the VW-Parareal algorithm is converged for the three scenarios benefiting from the decreased average window length from 1.00 seconds to be around 0.3200 seconds. Moreover, for the cases with $h_c$=0.050 seconds, where both conventional fixed window Parareal and the proposed VW-Parareal algorithms are converged in one iteration, the VW-Parareal algorithm allows greatly increased window lengths (e.g., 1.860 seconds) meanwhile maintaining the convergence of the Parareal iteration procedure. These results demonstrate the benefits of the proposed VW-Parareal algorithm in improving the convergence performance.

### 5.2.4. Improved Coarse Update Strategy

In this test, the improved coarse update strategy is further added to the VW-Parareal method. Table 14 and Table 15 respectively give the average number of iterations and the average window length of the this algorithm under different coarse time steps and different window lengths. Results show that the VW-Parareal method with the second-order Parareal update formula needs less iterations and longer Parareal window sizes, compared with Table 12 and Table 13 using VW-Parareal method with a first-order Parareal update.

### 5.2.5. VOVS-DT Method

The following four methods are tested, including 1) the conventional DT method with a fixed order of 8 and a fixed time step of 0.1 seconds, 2) the VO-DT method with variable orders but a fixed time step of 0.1 seconds, 3) the VS-DT method with variable steps but a fixed order of 8, and 4) the VOVS-DT method with both variable orders and variable time steps. Table 16 and Table 17 respectively gives the average order and average time steps of the four methods in the three scenarios. They show the average time step length is decreased and the average order is increased for more severe faults, because the transient dynamics are stronger and lasts longer than those of less severe faults. Also, the VOVS-DT method allows larger average time step lengths than the VS-DT method only (e.g., 0.088 seconds versus 0.069 seconds in the Scenario 3 of Table 16), and smaller average order than the VO-DT method only (e.g., 8.94 versus 9.33 in the Scenario 2 of Table 17). These results demonstrate the flexibility of the proposed VOVS-

DT method in adjusting orders and time step lengths simultaneously to adapt to different scenarios.

Take the scenario 2 as an example, Fig. 31 shows the trajectory of selected variables (including relative rotor angles and rotor speeds of all generators, and the voltage magnitude and electrical power outputs of all buses) over a 1-second pre-fault simulation, a 0.2-second fault-on simulation and a 10-second post-fault simulation. Fig. 32 shows the time step length and the order of the VOVS-DT method over the whole simulation. It is observed that low orders and larger time steps are selected automatically by the proposed algorithm in the first-second pre-fault steady-state simulation and the last 4 seconds in the post-fault simulation, where the oscillation has been well-damped. On the other hand, higher orders and smaller time steps are selected in the fault-on simulation and the first 6 seconds in the post-fault simulation, where there is strong transient dynamics. These results show the proposed algorithm is able to adjust orders and time steps adaptively during the simulation.

### 5.3 Case Study on a 2383-bus system

In this case study, the following three methods with the same parameters $h_c$=0.1 seconds and $h_w$=1 seconds are compared for simulating a Polish 2383-bus system under a permanent three-phase fault on bus 9 cleared after 0.3 seconds by tripping the line between bus 9 and bus 6: 1) conventional numerical-based Parareal method; 2) the integrated approach 1 using a DT-based VW-Parareal method with improved coarse solution update strategy; 3) the integrated approach 2: a VOVS-DT-based VW-Parareal method with improved coarse solution update strategy. Table 18 gives the average

number of iterations, the average window length, the average order and time step length in the DT or VOVS-DT method, and the CPU time of the three methods. It shows both the integrated approach 1 and 2 perform better than the conventional Parareal method, where the number of iterations and the CPU time is significantly reduced. Moreover, integrated approach 2 has better computational efficiency than integrated approach 1 benefiting from the smaller average time step length of DT than integrated approach 1, despite the slightly larger number of iterations and the average order of DT in this case.

For the integrated approach 2, Fig. 33 shows the time-step length and order of DT during the simulation, Fig. 34 shows the Parareal window size and number of iterations in each window during the simulation, and Fig. 35 further shows the time step length of the DT method, the order of the DT method, and the Parareal window size in a three-dimensional space, where the red dot represents the conventional Parareal method with a fixed order, fixed step size and fixed Parareal window size. The results show the proposed approach extends the conventional Parareal method by introducing flexibilities along three dimensions, i.e., the step size and order of the coarse solver and the Parareal window length, thus greatly enhancing the convergence performance and efficiency of the Parareal method.

## 5.4 Conclusion

This chapter proposes a DT- based variable-order variable-step variable-window adaptive parareal method for temporal parallelization of power system simulation with greatly enhanced convergence performance and efficiency. The proposed method integrates the temporal parallelization capability of the Parareal method and the highly

adaptive feature of the DT method. Extensive simulations on a 39-bus system and a 2383-bus system demonstrate the effectiveness of the proposed approach.

# CHAPTER 6

# DT BASED ADAPTIVE SWITCHING CONTROL OF WIND TURBINES

This chapter proposes a novel switching control strategy which predicts the safety of a frequency response right after a disturbance by evaluating the derived semi-analytical solutions of system frequency response model over a certain post-disturbance period of interest, and activates frequency support mode only when the frequency response is predicted as unsafe. The rationale of the proposed strategy is real time evaluation of an offline obtained semi-analytical solution on frequency responses using real-time measurements. Such a semi-analytical solution is in form of ultra-high order Taylor series derived by the DT method on the differential equation model of the system. The ultra-high order nature of the solution enables its largely extended convergence region to cover the frequency response period of interest, so that the frequency response of a WTG can be accurately predicted when a disturbance is detected and the WTG provides frequency support only for an unsafe response, thus avoiding the unnecessary switches. The case studies on a 4-bus power system and a New England 10-machine 39-bus system show the effectiveness of the proposed strategy.

## 6.1 Problem Description

The system frequency response can be assessed using three metrics: the rate of change of frequency (ROCOF), frequency nadir and primary settling frequency. The effect of increasing wind power penetration on system frequency response can be

illustrated in Fig.. 36, where a larger ROCOF, lower frequency nadir and lower primary setting frequency are observed.

To address the inadequate frequency response issue, the WTGs are also expected to operate at frequency support mode in addition to its normal mode. Despite the many frequency support functions proposed in the literature, the strategies on switching among these modes are not carefully studied. A conceptual system frequency response where WTGs do not provide frequency support is shown in Fig. 37 with the case details and actual/ideal WTG reaction shown in Table 19.

The WTG should activate its frequency support mode only for case 3 where the frequency deviation will otherwise surpass the nadir limit, and not react to the frequency deviations in case 1 and case 2. However, when the frequency surpasses the deadband but does not hit the nadir limit as in case 2, the frequency support will also be activated due to the deadband setting. The case 2 may occur frequently in WTG operations since the deadband setting is usually conservative to have some safety margin. These unnecessary frequency supports reduce the profits of wind farms and should be avoided. A straightforward way to reduce unnecessary switches is to set a larger deadband. However, a larger deadband can pose the system under higher risks of frequency excursions. Moreover, it is very hard to determine the critical deadband value with ensured safety. Therefore, the purpose of this work is to investigate new methods to avoid unnecessary switches meanwhile ensuring adequate frequency responses.

## 6.2 Proposed Adaptive Switching Control Strategy

Existing switching control strategies of WTGs fall into two categories, as shown in (6-1) and (6-2), respectively, where $\Delta w(t_0)$ is the frequency drop measured at $t=t_0$, $\Delta w_{db}$ is a preset deadband width on the frequency deviation, $\Delta w_{crt}(\Delta p_d)$ is the critical deadband width, and $\Delta p_d$ is a power imbalance assumed to be known under a disturbance.

$$\begin{aligned}&\text{if } \Delta w(t_0) < \Delta w_{db} : \text{in MPPT mode,}\\&\text{otherwise: in frequency support mode}\end{aligned} \tag{6-1}$$

$$\begin{aligned}&\text{if } \Delta w(t_0) < \Delta w_{crt}(\Delta p_d) : \text{in MPPT mode,}\\&\text{otherwise: in frequency support mode}\end{aligned} \tag{6-2}$$

The strategy in (6-1)is widely adopted by WTGs manufacturers due to its ease of implementation, e.g., the deadband width recommended by GE is $\Delta w_{db}$=0.15HZ for a certain frequency support mode [18]. However, this strategy is not flexible to adapt to disturbances and the deadband width is often conservatively small such that the unnecessary mode switching of a WTG could be triggered. In comparison, the strategy in (6-2) can overcome the conservativeness, but it has to calculate a critical deadband width for each disturbance (e.g., by extensive simulation or by solving an optimization problem [18]), which is a huge computation burden and makes it difficult for real-time implementation.

### 6.2.1. Proposed Switching Control Strategy

To overcome the conservativeness and the computation burden of existing strategies (6-1) and (6-2), this letter proposes a novel switching control strategy in (3), where $\Delta w(t)$, $t\in[t_0,T]$ is the frequency response predictor and $\Delta w_{lim}$ is the safety limit of

frequency drop. The detailed derivation of the frequency response predictor is in Chapter 6.2.3. It is observed from (6-3) that the proposed strategy uses frequency deviation at $t=t_0$, the time of detecting power imbalance such as loss of generation or load, but further predicts the frequency response over a future time period $(t_0,T]$. Also, the proposed strategy does not need a deadband since it directly compares the predicted frequency response with the safety limit.

$$\begin{aligned} &\text{if } \Delta w(t) < \Delta w_{\lim} \text{ for } \forall t \in [t_0,T]\text{: in MPPT mode,} \\ &\text{otherwise: in frequency support mode} \end{aligned} \tag{6-3}$$

### 6.2.2. System Model

Consider the augmented frequency response model [18] in (6-4)-(6-5), where (6-4) is the classical frequency response model based on center of inertia, and (6-5) is the WTG model.

$$\begin{aligned} \Delta\dot{w} &= \frac{w_s}{2H}\left(\Delta p_m - \Delta p_d + \sum_{i=1}^{N}\Delta p_{gen,i} - \frac{D}{w_s}\Delta w\right) \\ \Delta\dot{p}_m &= \frac{1}{t_{ch}}\left(\Delta p_v - \Delta p_m\right) \\ \Delta\dot{p}_v &= \frac{1}{t_g}\left(-\Delta p_v - \frac{1}{R}\Delta w\right) \end{aligned} \tag{6-4}$$

$$\begin{aligned} \Delta\dot{w}_{r,i} &= A_i\Delta w_{r,i} + B_{1,i}\Delta\dot{w} + B_{2,i}\Delta w \\ \Delta p_{gen,i} &= C_i\Delta w_{r,i} + D_{1,i}\Delta\dot{w} + D_{2,i}\Delta w \end{aligned} \tag{6-5}$$

In (6-4), $\Delta p_m$, $\Delta p_v$ are output power of turbines and governors; $\tau_{ch}$, $\tau_g$ are their time constants, respectively; $D$ and $R$ are damping and droop coefficients; $\omega_s$ is the nominal frequency; $N$ is the number of WTGs and $i$ is the index of WTGs. In (6-5), $\Delta w_{r,i}$ and $\Delta p_{gen,i}$ are the rotor speed deviation and the increased power generation. Coefficients $A_i$

and $C_i$ are parameters of the WTG model; $B_{1,i}$ and $D_{1,i}$ are parameters for inertia emulation control mode; $B_{2,i}$ and $D_{2,i}$ are parameters for primary frequency control mode. Parameters $B_{1,i}$, $D_{1,i}$, $B_{2,i}$ and $D_{2,i}$ are zero for the MPPT mode and non-zero for frequency support modes.

### 6.2.3. Frequency Response Predictor

Given system state variables $\mathbf{x}=[\Delta w,\ \Delta p_m,\ \Delta p_v,\ \Delta w_{r,1},\ldots,\ \Delta w_{r,N}]^{\mathrm{T}}$ and the disturbance at time instant $t=t_0$, this section aims at finding a function $F(\mathbf{x}(t_0))$ in (6-6) to predict the frequency response over a future time period $(t_0,T]$.

$$\boldsymbol{x}(t_0+\Delta t)=F(\boldsymbol{x}(t_0)),\ \forall \Delta t\in[0,T-t_0] \tag{6-6}$$

Since the analytical expression of (6-6) is generally unavailable, this letter seeks a semi-analytical, approximate solution about the frequency response in the form of power series of time in (6-7), where $f_k(\mathbf{x}(t_0))$ is the $k^{th}$ order power series coefficients to be solved.

$$\boldsymbol{x}(t_0+\Delta t)=\boldsymbol{x}(t_0)+f_1(\boldsymbol{x}(t_0))\Delta t+\ldots f_K(\boldsymbol{x}(t_0))\Delta t^K \tag{6-7}$$

To effectively solve the power series coefficients $f_k(\mathbf{x}(t_0))$, the DTM is adopted due to its proved accuracy and efficiency for solving detailed power system models. By applying the DT rules to each term in (6-4)-(6-5), a recursive equation about the $k^{\mathrm{th}}$ order power series coefficients $f_k(\mathbf{x}(t_0))=[\Delta w[k],\ \Delta p_m[k],\ \Delta p_v[k],\ \Delta w_{r,1}[k],\ldots,\ \Delta w_{r,N}[k]]^{\mathrm{T}}$ is obtained in (6-8)-(6-12).

$$(k+1)\Delta w[k+1] = \frac{w_s}{2H}\left(\Delta p_m[k] - \Delta p_d[k] + \sum_{i=1}^{N} \Delta p_{gen,i}[k] - \frac{D}{w_s}\Delta w[k]\right) \tag{6-8}$$

$$(k+1)\Delta p_m[k+1] = \frac{1}{t_{ch}}\left(\Delta p_v[k] - \Delta p_m[k]\right) \tag{6-9}$$

$$(k+1)\Delta p_v[k+1] = \frac{1}{t_g}\left(-\Delta p_v[k] - \frac{1}{R}\Delta w[k]\right) \tag{6-10}$$

$$(k+1)\Delta w_{r,i}[k+1] = A_i \Delta w_{r,i}[k] + B_{2,i}\Delta w[k] + B_{1,i} \times (k+1)\Delta w[k+1] \tag{6-11}$$

$$\Delta p_{gen,i}[k] = C_i \Delta w_{r,i}[k] + D_{2,i}\Delta w[k] + D_{1,i} \times (k+1)\Delta w[k+1] \tag{6-12}$$

With the derived recursive equations (6-8)-(6-12), the power series coefficients $f_k(\mathbf{x}(t_0))$ up to any desired order can be derived. Especially, the frequency response at time instant $t=t_0+\Delta t$ is predicted by power series of time in (6-13).

$$\Delta w(t_0 + \Delta t) = \Delta w[0] + \Delta w[1]\Delta t + \Lambda\ \Delta w[K]\Delta t^K \tag{6-13}$$

The predicted frequency response (6-13) is accurate within a certain time period whose length increases with the order $K$. To ensure the accuracy of the predicted frequency response over a longer time period of interest, arbitrary high order $K$ such as 100 to 200 can be easily derived by the DTM method. Besides, the multi-time window strategy can be used to further enhance the accuracy of (6-13). For illustration, the first three terms in the right-hand side of (6-13) are derived below.

First, initialize $f_0(\mathbf{x}(t_0))=[\Delta w[0], \Delta p_m[0], \Delta p_v[0], \Delta w_{r,1}[0],\ldots, \Delta w_{r,N}[0]]^T = [\Delta w(t_0), \Delta p_m(t_0), \Delta p_v(t_0), \Delta w_{r,1}(t_0),\ldots, \Delta w_{r,N}(t_0)]^T$. Second, $f_1(\mathbf{x}(t_0))=[\Delta w[1], \Delta p_m[1], \Delta p_v[1], \Delta w_{r,1}[1],\ldots, \Delta w_{r,N}[1]]^T$ is calculated by

$$\Delta w[1] = \frac{w_s}{2H}\left(\Delta p_m(t_0) - \Delta p_d + \sum_{i=1}^{N} \Delta p_{gen,i}[0] - \frac{D}{w_s}\Delta w(t_0)\right)$$

$$\Delta p_m[1] = \frac{1}{t_{ch}}\left(\Delta p_v(t_0) - \Delta p_m(t_0)\right)$$

$$\Delta p_v[1] = \frac{1}{t_g}\left(-\Delta p_v(t_0) - \frac{1}{R}\Delta w(t_0)\right)$$

$$\Delta w_{r,i}[1] = A_i \Delta w_{r,i}(t_0) + B_{2,i}\Delta w(t_0) + B_{1,i}\Delta w[1]$$

$$\Delta p_{gen,i}[0] = C_i \Delta w_{r,i}[0] + D_{2,i}\Delta w[0] + D_{1,i} \times \Delta w[1]$$

Then, $\Delta w[2]$ is calculated by

$$\Delta w[2] = \frac{w_s}{2H}\left(\Delta p_m[1] + \sum_{i=1}^{N} \Delta p_{gen,i}[1] - \frac{D}{w_s}\Delta w[1]\right)$$

Finally, $\Delta w(t_0 + \Delta t) = \Delta w[0] + \Delta w[1]\Delta t + \Delta w[2]\Delta t^2$.

## 6.3 Case Study

### 6.3.1. Test on a 4-bus System

To validate the effectiveness of the proposed strategy, a four-bus system in [18] is used, shown in Fig. 38. The synchronous generator at bus 1 represents a 600 MW conventional generation, aggregated from four thermal power plants with 150 MW capacity each. The WTG at bus 3 represents a 300 MW wind power generation, aggregated from 200 DFIGs with 1.5 MW capacity each. The parameters in the frequency response model and the WTG model are adopted from [18] where $A_i$ = -0.0723, $C_i$ = 0.0127, $H$=4, $\tau_{ch}$ = 0.3, $\tau_g$ = 0.1, $\omega_s$ = 60HZ, $D$ = 1, $R$ = 0.05, $B_{1,i}$=-0.6246, $B_{2,i}$=0.1874, $D_{1,i}$=-0.10, $D_{2,i}$=-0.03, $\Delta w_{db}$=0.2 HZ and $\Delta w_{lim}$=0.5 HZ.

The accuracy of the proposed DTM-based semi-analytical solution is tested under three scenarios with different disturbance severities, shown in Table 20. The commonly used numerical integration method RK4 is used as the benchmark. The system frequency response over a time window of 4 seconds after the disturbance occurs is studied. In the RK4 method, a time step of 0.04 second is used and 100 consecutive time windows are used to make the frequency response solution. In the semi-analytical solution in (9), $K = 200$ is selected. The accuracy of the semi-analytical solution is shown in Fig. 39. Under all three scenarios, the semi-analytical solution matches the RK4 solution up to $t = 4.4$ s, i.e. equal to 340 consecutive time intervals of RK4. The effect of order $K$ on the semi-analytical solution accuracy is also studied. The convergence region of the semi-analytical solution is drawn in Fig. 40. It shows the convergence region of the semi-analytical solution increases with its order. For the four-bus system, the frequency nadir limit usually happens before $t = 2$ s. Therefore, an order $K = 50$ is selected for the proposed control strategy.

To show the proposed adaptive switching strategy can check the safety in real time, the time cost of evaluating the analytical solution are compared with the time for frequency to reach the deadband in the worst-case scenario 3. In Fig. 39c), it takes 0.19 second for the frequency deviation to hit the deadband $\Delta w = 0.2$ Hz, while the time cost for evaluating the analytical solution with $K = 50$ at one time instant is only $6.04 \times 10^{-4}$ second. It means the analytical solution evaluation can be done at more than 300 time instants before the deadband is met. Moreover, evaluations of an analytical solution at different time instants are independent from each other and hence can be

conducted by parallel computers. Therefore, the proposed control strategy has enough time to check the safety of a frequency response and bypass the deadband for a safe response to avoid unnecessary switches.

To validate the proposed control strategy in avoiding unnecessary switches, the disturbances $\Delta p_d$ are changed from 10 MW to 150 MW with the step size of 10 MW shown in Fig. 41. Among the 15 cases, only 2 cases require WTGs to provide frequency support and they are successfully detected by the safety check in the proposed control strategy. Otherwise, the WTGs will be switched to frequency support modes in 10 cases, where 8 of them are switched unnecessarily.

The proposed adaptive switching control strategy not only avoids the unnecessary switches, but also lets the WTGs react correctly when the switching is indeed needed. For the worst-case scenario 3, the proposed strategy successfully sensed the unsafety, and the frequency support mode is activated timely. The frequency response when switching to different frequency support modes at $\mathrm{D}w = 0.2$ Hz is shown in Fig. 42, where mode 1 is the MPPT mode and the other modes represent frequency control modes with different control parameters $B_{1,i}$, $B_{2,i}$, $D_{1,i}$, $D_{2,i}$. It shows the safety is preserved for all frequency support modes using the proposed strategy.

### 6.3.2. Test on the 39-bus System

To test the performance of the proposed strategy on real-world power system, a 10-machine 39-bus system is used where five synchronous generators are replaced by WTGs. The parameters in (4)-(5) are the same with the 4-bus system.

Two scenarios, i.e., a safe scenario with power imbalance of -500 MW and an unsafe scenario with power imbalance of -1000MW, are tested. Fig. 43 give the frequency responses in the two scenarios. The proposed strategy activates the frequency support mode only for the unsafe scenario. In contrast, the deadband based strategy would activate frequency support for both the safe and unsafe scenarios. This result shows the proposed strategy can overcome conservativeness compared with the deadband based strategy.

## 6.4 Conclusion

This chapter proposes an analytical, adaptive switching control strategy to overcome the conservativeness of the deadband based strategy. For safe responses, the frequency support mode is not activated even if the deadband is met. For unsafe responses, the frequency support mode is activated immediately once unsafety is predicted. The strategy is shown effective to control WTGs for adequate frequency response.

# CHAPTER 7

# AN AFFINE RECURSION FORM OF NONLINEAR DAES

This chapter proposes an affine recursion form (ARF) of nonlinear DAEs. The advantage of ARF is that nonlinear DAEs are converted to formally linear equations about Taylor series coefficients, which enables straightforward calculation of semi-analytical solutions of the nonlinear DAEs in the form of arbitrary high-order Taylor series. Since practical DAE models often contain compositional functions with complicated compositional structures, this chapter first studies the ARF of generic compositional functions, including its existence, uniqueness, and propagation of the ARF over a compositional function structure to derive ARFs of compositional functions from simple functions. After that, this chapter derives the ARF of nonlinear DAEs and designs an ARF-based DAE solution algorithm. Finally, the chapter applies the proposed method to derive the ARF of a detailed power system DAE model.

## 7.1 ARF of Compositional Functions

### 7.1.1. Definition

Consider a vector-valued nonlinear compositional function $\boldsymbol{f} \circ \boldsymbol{x} : t \in \mathrm{T} \subset \mathbb{R} \mapsto (\boldsymbol{f} \circ \boldsymbol{x})(t) \in \mathrm{F} \subset \mathbb{R}^{q}$, which is composed of $\boldsymbol{x} : t \in \mathrm{T} \mapsto \boldsymbol{x}(t) \in \mathrm{X}_1 \subset \mathbb{R}^{d}$ and $\boldsymbol{f} : \boldsymbol{x} \in \mathrm{X}_2 \subset \mathbb{R}^{d} \mapsto \boldsymbol{f}(\boldsymbol{x}) \in \mathrm{F}$, where $\mathrm{T}, \mathrm{F}, \mathrm{X}_1, \mathrm{X}_2$ are connected sets, $d$ and $q$ are two positive integers.

**Assumption 1**: Assume $\mathrm{X}_1 \subset \mathrm{X}_2$ for function composition operations to be valid, and all components in $\boldsymbol{x}$ and $\boldsymbol{f}$ are analytical functions in their domain, i.e.,

$x_i \in C^w(\mathrm{T}), i = 1,\ldots d; f_i \in C^w(\mathrm{X}_2), i = 1,\ldots q$. Denote the set of such compositional functions as $\mathrm{C}$.

**Definition 2** *(Affine recursion form): The Affine recursion form of a vector-valued nonlinear compositional function* $\boldsymbol{f} \circ \boldsymbol{x} \in \mathrm{C} : t \in \mathrm{T} \subseteq \mathbb{R} \mapsto (\boldsymbol{f} \circ \boldsymbol{x})(t) \in \mathrm{F} \subseteq \mathbb{R}^q$ *at* $t = t_0$ *is a two-tuple,* $(\boldsymbol{A}_{t_0}, \boldsymbol{b}_{t_0})$*, consisting of a matrix-valued function* $\boldsymbol{A}_{t_0}$ *and a vector-valued function* $\boldsymbol{b}_{t_0}$ *that satisfy:*

$$(\boldsymbol{F} \circ \boldsymbol{X})_{t_0}(k) = \boldsymbol{A}_{t_0}\left(\boldsymbol{X}_{t_0}(0)\right)\boldsymbol{X}_{t_0}(k) + \boldsymbol{b}_{t_0}\left(\boldsymbol{X}_{t_0}(\mathrm{K})\right) \tag{7-1}$$

*where*

$$\boldsymbol{X}_{t_0} : k \in \mathbb{N}_0 = \mathbb{N} \cup \{0\} \mapsto \boldsymbol{X}_{t_0}(k) = \frac{1}{k!}\frac{d^k \boldsymbol{x}}{dt^k}\Big|_{t=t_0} \in \bar{\mathrm{X}} \subseteq \mathbb{R}^d$$

$$(\boldsymbol{F} \circ \boldsymbol{X})_{t_0} : k \in \mathbb{N}_0 \mapsto (\boldsymbol{F} \circ \boldsymbol{X})_{t_0}(k) = \frac{1}{k!}\frac{d^k(\boldsymbol{f} \circ \boldsymbol{x})}{dt^k}\Big|_{t=t_0} \in \bar{\mathrm{F}} \subseteq \mathbb{R}^q$$

$$\boldsymbol{A}_{t_0} : \boldsymbol{X}_{t_0}(k) \in \bar{\mathrm{X}} \subseteq \mathbb{R}^d \mapsto \boldsymbol{A}_{t_0}\left(\boldsymbol{X}_{t_0}(k)\right) \in \bar{\mathrm{A}} \subseteq \mathbb{R}^{q \times d}$$

$$\boldsymbol{b}_{t_0} : \boldsymbol{X}_{t_0}(\mathrm{K}) \in \bar{\mathrm{X}} \subseteq \mathbb{R}^{d \times k} \mapsto \boldsymbol{b}_{t_0}\left(\boldsymbol{X}_{t_0}(\mathrm{K})\right) \in \bar{\mathrm{B}} \subseteq \mathbb{R}^q$$

$$\boldsymbol{X}_{t_0}(0) = \boldsymbol{x}(t_0)$$

$$\mathrm{K} = \{0, 1, \ldots k-1\}$$

$$\boldsymbol{X}_{t_0}(\mathrm{K}) = \left\{\boldsymbol{X}_{t_0}(0), \boldsymbol{X}_{t_0}(1), \ldots \boldsymbol{X}_{t_0}(k-1)\right\}$$

**Remark 1***:* There are the following three observations from (7-1):

First, from Taylor's theorem, these hold for a neighborhood of $t_0$, i.e. $t \in \mathrm{I}(t_0)$:

$$\boldsymbol{x}(t) = \sum_{k=0}^{\infty} \boldsymbol{X}_{t_0}(k)(t - t_0)^k \tag{7-2}$$

$$\left(\boldsymbol{f} \circ \boldsymbol{x}\right)(t) = \sum_{k=0}^{\infty} (\boldsymbol{F} \circ \boldsymbol{X})_{t_0}(k)(t - t_0)^k \tag{7-3}$$

Therefore, eq. (7-1) builds upon the relationship between the $k^{th}$ order Taylor series coefficients $(\boldsymbol{F} \circ \boldsymbol{X})_{t_0}(k)$ of the compositional function $(\boldsymbol{f} \circ \boldsymbol{x})(t)$ and the $k^{th}$ order Taylor series coefficients $\boldsymbol{X}_{t_0}(k)$ of the function $\boldsymbol{x}(t)$ for any $k \in \mathbb{Z}_0$.

Second, eq. (7-1) decouples the $(\boldsymbol{F} \circ \boldsymbol{X})_{t_0}(k)$ into two terms, respectively related to $\boldsymbol{X}_{t_0}(k)$ and $\boldsymbol{X}_{t_0}(\mathrm{K})$. $\boldsymbol{A}_{t_0}\left(\boldsymbol{X}_{t_0}(0)\right)$ in the first term only depends on $\boldsymbol{X}_{t_0}(0)$ and $\boldsymbol{b}_{t_0}\left(\boldsymbol{X}_{t_0}(\mathrm{K})\right)$ in the second term only depends on $\boldsymbol{X}_{t_0}(0), \boldsymbol{X}_{t_0}(1), \ldots \boldsymbol{X}_{t_0}(k-1)$. Neither $\boldsymbol{A}_{t_0}\left(\boldsymbol{X}_{t_0}(0)\right)$ or $\boldsymbol{b}_{t_0}\left(\boldsymbol{X}_{t_0}(\mathrm{K})\right)$ needs to know $\boldsymbol{X}_{t_0}(k)$, so eq. (7-1) is a linear equation between $(\boldsymbol{F} \circ \boldsymbol{X})_{t_0}(k)$ and $\boldsymbol{X}_{t_0}(k)$. This indicates a recursive approach to the solution of a nonlinear system through calculating such linear equations related to Taylor series coefficients from low to high orders for the desired accuracy.

Third, although eq. (7-1) has a linear relationship between $(\boldsymbol{F} \circ \boldsymbol{X})_{t_0}(k)$ and $\boldsymbol{X}_{t_0}(k)$, $\boldsymbol{A}_{t_0}$ and $\boldsymbol{b}_{t_0}$ can include nonlinear functions about $\boldsymbol{X}_{t_0}(0)$ and $\boldsymbol{X}_{t_0}(\mathrm{K})$, respectively. Therefore, the nonlinearity in the compositional function $(\boldsymbol{f} \circ \boldsymbol{x})(t)$ is enclosed by the new form in $\boldsymbol{A}_{t_0}$ and $\boldsymbol{b}_{t_0}$ but not lost. Actually, $(\boldsymbol{f} \circ \boldsymbol{x})(t)$ or its truncated Taylor's expansion can be recovered from the ARF $(\boldsymbol{A}_{t_0}, \boldsymbol{b}_{t_0})$ by (7-3) up to any desired order.

### 7.1.2. Existence and Uniqueness

On the existence and uniqueness of the ARF, there is this theorem:

**Theorem 1**: *For any compositional function* $\boldsymbol{f} \circ \boldsymbol{x} \in C: t \in T \to (\boldsymbol{f} \circ \boldsymbol{x})(t) \in F \to \mathbb{R}^q$ *that satisfies Assumption 1, and for any* $t_0 \in T$, *if* $\boldsymbol{X}_{t_0}(k) \neq 0 \ \forall k \in \mathbb{N}_0$, *then there exists a unique ARF* $(\boldsymbol{A}_{t_0}, \boldsymbol{b}_{t_0})$ *at* $t = t_0$.

*Proof*: We only need to show

$$\frac{d^k (f_i \circ \boldsymbol{x})(t)}{dt^k} = \boldsymbol{a}_i^T \frac{d^k \boldsymbol{x}(t)}{dt^k} + b_i \tag{7-4}$$

holds for $i = 1, \ldots, q$, where $\boldsymbol{a}_i$ is a function of $\boldsymbol{x}(t)$ and $b_i$ is a function of both $\boldsymbol{x}(t)$ and its derivatives $\boldsymbol{x}(t)$ up to the order of $k-1$. Then, the existence of the ARF $(\boldsymbol{A}_{t_0}, \boldsymbol{b}_{t_0})$ can be proved by defining $\boldsymbol{A}_{t_0} = \left[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_q\right]^T_{t=t_0}$ and $\boldsymbol{b}_{t_0} = \left[b_1, \ldots, b_q\right]^T_{t=t_0}$.

When $k = 1$, there is

$$\frac{d^k (f_i \circ \boldsymbol{x})(t)}{dt^k} = \sum_{j=1}^{d} \frac{\partial f_i}{\partial x_j} \frac{dx_j}{dt} = \left[\frac{\partial f_i}{\partial x_1} \ \cdots \ \frac{\partial f_i}{\partial x_m}\right]^T \frac{d\boldsymbol{x}(t)}{dt}$$

whose satisfaction to (7-4) can easily be verified with $\boldsymbol{a}_i = \left[\frac{\partial f_i}{\partial x_1} \ \cdots \ \frac{\partial f_i}{\partial x_d}\right]^T$, a function of $\boldsymbol{x}(t)$, and $b_i = 0$.

Assume there exists $\boldsymbol{a}_{i,n}$ as a function of $\boldsymbol{x}(t)$ and $b_{i,n}$ as a function of both $\boldsymbol{x}(t)$ and its derivatives up to the order of $n-1$ for $k = n$, such that

$$\frac{d^n\left(f_i \circ \boldsymbol{x}\right)(t)}{dt^n} = \boldsymbol{a}_{i,n}{}^T \frac{d^n \boldsymbol{x}\left(t\right)}{dt^n} + b_{i,n} .$$

Then, verify the existence of $\boldsymbol{a}_{i,n+1} = \boldsymbol{a}_{i,n}$ as a function of $\boldsymbol{x}(t)$ and $b_{i,n+1} = \frac{d\boldsymbol{a}_{i,n}{}^T}{dt}\frac{d^k \boldsymbol{x}\left(t\right)}{dt^k} + \frac{db_{i,n}}{dt}$ as a function of both $\boldsymbol{x}(t)$ and its derivatives up to the order of $n$ for $k = n + 1$, such that

$$\frac{d^{n+1}\left(f_i \circ \boldsymbol{x}\right)(t)}{dt^{n+1}} = \boldsymbol{a}_{i,n+1}{}^T \frac{d^{n+1} \boldsymbol{x}\left(t\right)}{dt^{n+1}} + b_{i,n+1} .$$

The uniqueness of the ARF $(\boldsymbol{A}_{t_0}, \boldsymbol{b}_{t_0})$ is obvious because if there exists another ARF $(\boldsymbol{A}_{t_0}^{¢}, \boldsymbol{b}_{t_0}^{¢})$ with $\boldsymbol{A}_{t_0}^{¢}{}^{1}\ \boldsymbol{A}_{t_0}$ or $\boldsymbol{b}_{t_0}^{¢}{}^{1}\ \boldsymbol{b}_{t_0}$, then there is at least one positive integer $k$ such that $\boldsymbol{A}_{t_0}^{¢}\left(\boldsymbol{X}_{t_0}(0)\right)\boldsymbol{X}_{t_0}(k) + \boldsymbol{b}_{t_0}^{¢}\left(\boldsymbol{X}_{t_0}(\mathrm{K})\right)^{1}\ \boldsymbol{A}_{t_0}\left(\boldsymbol{X}_{t_0}(0)\right)\boldsymbol{X}_{t_0}(k) \quad + \boldsymbol{b}_{t_0}\left(\boldsymbol{X}_{t_0}(\mathrm{K})\right)$. Then the coefficients $(\boldsymbol{F} \circ \boldsymbol{X})_{t_0}(k)$ in (7-3) of the function $(\boldsymbol{f} \circ \boldsymbol{x})(t)$ are not unique, which is contradictory to the uniqueness of the Taylor series of an analytical function. ■

**Notation**: for simplicity in the following sections, the subscript $t_0$ is omitted; $\boldsymbol{A}\left(\boldsymbol{X}(0)\right)$ and $\boldsymbol{b}\left(\boldsymbol{X}(\mathrm{K})\right)$ are written as $\boldsymbol{A}$ and $\boldsymbol{b}$ if not confusing.

### 7.1.3. Derivation

$\boldsymbol{y} = \boldsymbol{f} \circ \boldsymbol{x}$ defined on a connected set T Í ¡ in Definition 2 is a one-input one-output one-layer compositional function, which can be represented by a directed acyclic graph (DAG) [68] in Fig. 44a, where $\boldsymbol{u} = \boldsymbol{x}(t), \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{u})$. Its ARF $(\boldsymbol{A}, \boldsymbol{b})$ indicates

another compositional, dual function that exists on a discrete set $\Upsilon_0$, as shown in Fig. 44b, and has a linear form $\boldsymbol{Y}(k) = (\boldsymbol{F} \circ \boldsymbol{X})(k) = \boldsymbol{A}\boldsymbol{X}(k) + \boldsymbol{b}$ .

Generally, a compositional function $\boldsymbol{f} \circ \boldsymbol{x}$ can be further comprised of multiple layers of subfunctions in the form of $\boldsymbol{f} \circ \boldsymbol{x} = \boldsymbol{f}_m \circ \boldsymbol{f}_{m-1} \circ \cdots \boldsymbol{f}_0 \circ \boldsymbol{x} : \Upsilon \to \Upsilon^q$ whose DAG is in Fig. 45a, where the domains and ranges of $\boldsymbol{f}_i, i \in \{0 \ldots m\}$ and $\boldsymbol{x}$ are compatible for composition operations, and $\boldsymbol{u}_0 = \boldsymbol{x}(t), \boldsymbol{u}_{i+1} = \boldsymbol{f}_i(\boldsymbol{u}_i), i = 0 \ldots m-1, \boldsymbol{y} = \boldsymbol{f}_m(\boldsymbol{u}_m)$ . Then, the DAG for the transformed compositional function defined on $\Upsilon_0$ is shown in Fig. 45b, where for each function $\boldsymbol{f}_i, i \in \{0 \ldots m\}$, there exists a unique ARF denoted by $(\boldsymbol{A}_i, \boldsymbol{b}_i)$, such that

$$\begin{aligned} \boldsymbol{U}_{i+1}(k) &= (\boldsymbol{F}_i \circ \boldsymbol{U}_i)(k) = \boldsymbol{A}_i \boldsymbol{U}_i(k) + \boldsymbol{b}_i \\ &\qquad \text{for } i = 0 \ldots m-1 \quad . \\ \boldsymbol{Y}(k) &= (\boldsymbol{F}_m \circ \boldsymbol{U}_m)(k) = \boldsymbol{A}_m \boldsymbol{U}_m(k) + \boldsymbol{b}_m \end{aligned}$$

The following Theorem 2 gives the relationship between $(\boldsymbol{A}, \boldsymbol{b})$ in Fig. 44b and $(\boldsymbol{A}_i, \boldsymbol{b}_i)$ for $i \in \{0 \ldots m\}$ in Fig. 45b.

**Theorem 2**: *Let* $\boldsymbol{y} = \boldsymbol{f} \circ \boldsymbol{x} = \boldsymbol{f}_m \circ \boldsymbol{f}_{m-1} \circ \cdots \boldsymbol{f}_0 \circ \boldsymbol{x}$ . *If* $(\boldsymbol{A}, \boldsymbol{b})$ *is the ARF of* $\boldsymbol{f} \circ \boldsymbol{x}$ *and* $(\boldsymbol{A}_i, \boldsymbol{b}_i), i \in \{0 \ldots m\}$ *are the ARFs of* $\boldsymbol{u}_{i+1} = \boldsymbol{f}_i(\boldsymbol{u}_i), i = 0 \ldots m-1, \boldsymbol{y} = \boldsymbol{f}_m(\boldsymbol{u}_m)$. *There is*

$$\boldsymbol{A} = \prod_{i=0}^{m} \boldsymbol{A}_i, \quad \boldsymbol{b} = \boldsymbol{b}_m + \sum_{s=1}^{m} \prod_{i=s}^{m} \boldsymbol{A}_i \boldsymbol{b}_{s-1} \tag{7-5}$$

*Proof:* It is proved from the derivation below:

$$
\begin{aligned}
&(\boldsymbol{F}_m \circ \boldsymbol{F}_{m-1} \circ \cdots \boldsymbol{F}_0 \circ \boldsymbol{X})(k) \\
&= \boldsymbol{A}_m \boldsymbol{U}_m + \boldsymbol{b}_m \\
&= \boldsymbol{A}_m (\boldsymbol{A}_{m-1} \boldsymbol{U}_{m-1} + \boldsymbol{b}_{m-1}) + \boldsymbol{b}_m \\
&\cdots \qquad \blacksquare \\
&= \underbrace{\boldsymbol{A}_m \boldsymbol{A}_{m-1} \cdots \boldsymbol{A}_0}_{\boldsymbol{A}} \boldsymbol{X} \\
&+ \underbrace{\boldsymbol{b}_m + \boldsymbol{A}_m \boldsymbol{b}_{m-1} + \cdots + \boldsymbol{A}_m \boldsymbol{A}_{m-1} \cdots \boldsymbol{A}_1 \boldsymbol{b}_0}_{\boldsymbol{b}}
\end{aligned}
$$

More generally, each layer in Fig. 45 can further contain multiple nodes as shown in Fig. 46, where $r, s$ are respectively the numbers of nodes in the input layer and output layer, $n_i$ is the number of nodes in the $i^{\text{th}}$ layer, and

$$
\begin{aligned}
\boldsymbol{u}_{0j} &= \boldsymbol{x}_j(t), \forall j = 1, 2, \ldots r \\
\boldsymbol{u}_{i+1,j} &= \boldsymbol{f}_{i,j}(\boldsymbol{u}_{i,J_i}), \forall i = 0 \ldots m-1; \forall j = 1, 2, \ldots n_i \\
\boldsymbol{y}_i &= \boldsymbol{f}_{mi}(\boldsymbol{u}_{mJ_m}), \forall i = 1, 2, \ldots s \\
J_i &= 1, 2, \ldots n_i, \forall i = 0 \ldots m \\
\boldsymbol{u}_{i,J_i} &= \left\{ \boldsymbol{u}_{i1}, \ldots \boldsymbol{u}_{in_i} \right\} \forall i = 0 \ldots m
\end{aligned}
.
$$

Then, the DAG for the transformed compositional function defined on $\Psi_0$ is shown in Fig. 46b, where for each function $\boldsymbol{f}_{ij}, i \in \{0 \ldots m\}, j \in \{1, 2 \ldots, n_i\}$, there exists a unique ARF denoted by $(\boldsymbol{A}_{ij}, \boldsymbol{b}_{ij})$, such that

$$
\begin{aligned}
\boldsymbol{U}_{i+1,j}(k) &= \left(\boldsymbol{F}_{ij} \circ \boldsymbol{U}_{i\Omega_{ij}}\right)(k) = \boldsymbol{A}_{ij} \boldsymbol{U}_{i\Omega_{ij}}(k) + \boldsymbol{b}_{ij} \\
&\qquad \text{for } i \in \{0 \ldots m-1\}, j \in \{1, 2 \ldots, n_i\} \\
\boldsymbol{Y}_j(k) &= \left(\boldsymbol{F}_{mj} \circ \boldsymbol{U}_{m\Omega_{mj}}\right)(k) = \boldsymbol{A}_{mj} \boldsymbol{U}_{m\Omega_{mj}}(k) + \boldsymbol{b}_{mj} \\
&\qquad \text{for } j \in \{1, 2 \ldots, s\}
\end{aligned}
.
$$

where $\Omega_{ij} \subseteq \{1, 2 \ldots, n_{i-1}\}$ is a subset of indices for the nodes in layer $i$-1 that are connected to the node $\boldsymbol{f}_{ij}$.

Theorem 3 gives the relationship between $(\boldsymbol{A},\boldsymbol{b})$ in Fig. 44b and $(\boldsymbol{A}_{ij},\boldsymbol{b}_{ij})$ for $i \in \{0 \ldots m\}, j \in \{1,2\ldots,n_i\}$ in Fig. 46b. Before presenting Theorem 3, the following notations are used. Let $\mathrm{P}_{pq} = \boldsymbol{path}(\boldsymbol{x}_p,\boldsymbol{y}_q), p \in \{1,2,\ldots r\}, q \in \{1,2,\ldots s\}$ be the set of all paths between nodes $\boldsymbol{x}_p$ and $\boldsymbol{y}_q$ in the DAG in Fig. 46a; $\mathrm{P}_{uvq} = \boldsymbol{path}(\boldsymbol{f}_{uv},\boldsymbol{y}_q), u \in \{0 \ldots m\}, v \in \{1,2\ldots,n_i\}$ be the set of all paths between nodes $\boldsymbol{f}_{uv}$ and $\boldsymbol{y}_q$; $\mathrm{P}_l$ be the $l^{th}$ path in the set $\mathrm{P}_{pq}$ or $\mathrm{P}_{uvq}$.

**Theorem 3**: *Let* $\boldsymbol{y} = \boldsymbol{f} \circ \boldsymbol{x}$ *be a compositional function with compositional structure in Fig. 46a. If* $(\boldsymbol{A},\boldsymbol{b})$ *is the ARF of* $\boldsymbol{f} \circ \boldsymbol{x}$ *and* $(\boldsymbol{A}_{ij},\boldsymbol{b}_{ij})$, $i \in \{0 \ldots m\}, j \in \{1,2\ldots,n_i\}$ *are the ARFs of* $\boldsymbol{u}_{i+1,j} = \boldsymbol{f}_{i,j}(\boldsymbol{u}_{i,\mathrm{J}_i}), \forall i = 0 \ldots m-1; \forall j = 1,2,\ldots n_i$ *and* $\boldsymbol{y}_i = \boldsymbol{f}_{mi}(\boldsymbol{u}_{m\mathrm{J}_m}), \forall i = 1,2,\ldots s$, *then there is*:

$$\boldsymbol{A} = \begin{bmatrix} \bar{\boldsymbol{A}}_{11} & \bar{\boldsymbol{A}}_{12} & \ldots & \bar{\boldsymbol{A}}_{1r} \\ \bar{\boldsymbol{A}}_{21} & \bar{\boldsymbol{A}}_{22} & \ldots & \bar{\boldsymbol{A}}_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\boldsymbol{A}}_{s1} & \bar{\boldsymbol{A}}_{s2} & \ldots & \bar{\boldsymbol{A}}_{sr} \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} \bar{\boldsymbol{b}}_1 \\ \bar{\boldsymbol{b}}_2 \\ \vdots \\ \bar{\boldsymbol{b}}_s \end{bmatrix}$$

$$\bar{\boldsymbol{A}}_{qp} = \sum_{\mathrm{P}_l \in \mathrm{P}_{pq}} \prod_{\boldsymbol{f}_{ij} \in \mathrm{P}_l} \boldsymbol{A}_{ij} \tag{7-6}$$

$$\bar{\boldsymbol{b}}_q = \boldsymbol{b}_{mq} + \sum_{u=0}^{m} \sum_{v=1}^{n_m} \sum_{\mathrm{P}_l \in \mathrm{P}_{uvq}} \prod_{\boldsymbol{f}_{ij} \in \mathrm{P}_l} \boldsymbol{A}_{ij} \boldsymbol{b}_{uv}$$

*Proof:* It can be proved in a similar way to Theorem 2. ■

**Remark 2**: In Fig. 45a and 46a, the selection of subfunctions $\boldsymbol{f}_i$ or $\boldsymbol{f}_{ij}$ is not unique. For example, an alternative representation of $\boldsymbol{f}$ in Fig.45a could be $\boldsymbol{f} = \tilde{\boldsymbol{f}}_2 \circ \tilde{\boldsymbol{f}}_1 \circ \tilde{\boldsymbol{f}}_0$, where $\tilde{\boldsymbol{f}}_2 = \boldsymbol{f}_m \circ \boldsymbol{f}_{m-1}, \tilde{\boldsymbol{f}}_1 = \boldsymbol{f}_{m-2} \circ \mathrm{K} \circ \boldsymbol{f}_2, \tilde{\boldsymbol{f}}_0 = \boldsymbol{f}_1 \circ \boldsymbol{f}_0$. In this work, we assume $\boldsymbol{f}$ is comprised of multiple subfunctions, where each subfunction is simple enough so that their ARFs can be obtained from a set of fundamental rules that each can be regarded as a map from an analytical function $\boldsymbol{f}_i$ to its ARF $(\boldsymbol{A}_i, \boldsymbol{b}_i)$. Propositions 1 to 3 give the fundamental rules for mapping some generic nonlinear analytical functions to their ARFs. Proposition 4 provides the rules for function $(\boldsymbol{f} \circ \boldsymbol{x})(t)$ where $\boldsymbol{f}$ is a linear combination, multiplication, or division of $\boldsymbol{x}_1, \boldsymbol{x}_2$. Proposition 5 provides the rules for function $(\boldsymbol{f} \circ \boldsymbol{x})(t)$ where $\boldsymbol{f}$ is a trigonometric function, exponential function, or square root function of $\boldsymbol{x}$. Proposition 6 provides the rules for functions $(\boldsymbol{f} \circ \boldsymbol{x})(t)$ that skip $\boldsymbol{x}$ and directly depend on $t$. The detailed proofs of Propositions 4 to 6 are omitted, and these rules can be easily verified using Definition 2. The rules for other analytical functions can also be derived as needed. These rules make an extension of the differential transformation rules in our previous work.

**Proposition 4:** *Consider* $\boldsymbol{x}_1 : t \in \mathbb{R} \to \boldsymbol{x}_1(t) \in \mathbb{R}^d$ *and* $\boldsymbol{x}_2 : t \in \mathbb{R} \to \boldsymbol{x}_2(t) \in \mathbb{R}^q$. *Let* $a, b \in \mathbb{R}$ *be constant numbers.*

1. *Let* $d = q$ *, the ARF of* $(\boldsymbol{f} \circ \boldsymbol{x})(t) = a\boldsymbol{x}_1(t) + b\boldsymbol{x}_2(t)$ *is*

$(\boldsymbol{A},\boldsymbol{b}) = (a\boldsymbol{1}_d, b\boldsymbol{1}_d, \boldsymbol{0}_d)$, *where* $\boldsymbol{1}_d \in \mathbb{R}^d$ *is a vector with all elements being one.*

2. *Let* $d = q$ *, then the ARF of* $(\boldsymbol{f} \circ \boldsymbol{x})(t) = \boldsymbol{x}_1(t)^T \boldsymbol{x}_2(t)$ *is*

$$(\boldsymbol{A},\boldsymbol{b}) = \left(\boldsymbol{X}_2(0)^T, \boldsymbol{X}_1(0)^T, \sum_{m=1}^{k-1} \boldsymbol{X}_1(m)\boldsymbol{X}_2(k-m)\right).$$

3. *Let* $q = 1$, *then the ARF of* $(\boldsymbol{f} \circ \boldsymbol{x})(t) = \boldsymbol{x}_1(t)/\boldsymbol{x}_2(t)$ *is*

$$(\boldsymbol{A},\boldsymbol{b}) = \left(\frac{1}{\boldsymbol{X}_2(0)}, \frac{\boldsymbol{F}(0)^T}{\boldsymbol{X}_2(0)}, \frac{1}{\boldsymbol{X}_2(0)} \sum_{m=1}^{k-1} \boldsymbol{X}_2(m)\boldsymbol{F}(k-m)\right).$$

**Proposition 5:** *Consider* $\boldsymbol{x} : t \in \mathbb{R} \to \boldsymbol{x}(t) \in \mathbb{R}^d$.

1. *The ARF of* $\boldsymbol{f}_1 = \sin \boldsymbol{x}$ *and* $\boldsymbol{f}_2 = \cos \boldsymbol{x}$ *are respectively*

$$(\boldsymbol{A}_{f_1},\boldsymbol{b}_{f_1}) = \left(\boldsymbol{F}_2(0), \frac{1}{k}\sum_{m=1}^{k-1} \boldsymbol{F}_2(m)(k-m)\boldsymbol{x}(k-m)\right)$$
$$(\boldsymbol{A}_{f_2},\boldsymbol{b}_{f_2}) = \left(-\boldsymbol{F}_1(0), -\frac{1}{k}\sum_{m=1}^{k-1} \boldsymbol{F}_1(m)(k-m)\boldsymbol{x}(k-m)\right).$$

2. *The ARF of* $\boldsymbol{f} = e^{\boldsymbol{x}}$ *is*

$$(\boldsymbol{A},\boldsymbol{b}) = \left(\boldsymbol{F}(0), \frac{1}{k}\sum_{m=1}^{k-1} \boldsymbol{F}(m)(k-m)\boldsymbol{x}(k-m)\right).$$

3. *The ARF of* $\boldsymbol{f} = \sqrt{\boldsymbol{x}}$ *is*

$$(\boldsymbol{A},\boldsymbol{b}) = \left(\frac{1}{2\boldsymbol{F}(0)}, \frac{-1}{2\boldsymbol{F}(0)}\sum_{m=1}^{k-1} \boldsymbol{F}(m)\boldsymbol{x}(k-m)\right).$$

**Proposition 6:** *Consider* $\boldsymbol{x} : t \in \mathbb{R} \to \boldsymbol{x}(t) \in \mathbb{R}^d$. *Let* $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathbb{R}^d$ *be a constant vector and* $\boldsymbol{0}_d \in \mathbb{R}^d$ *be a zero vector.*

1. *The ARF of* $(\boldsymbol{f} \circ \boldsymbol{x})(t) = \boldsymbol{c}_1 t^n, n \in \mathbb{N}_0$ *is*

$$(\boldsymbol{A}, \boldsymbol{b}) = (\boldsymbol{0}_d, \boldsymbol{c}_1 d_{n,k}), d_{n,k} = \begin{cases} 1, n = k \\ 0, n \neq k \end{cases}.$$

2. *The ARF of* $(\boldsymbol{f} \circ \boldsymbol{x})(t) = \sin(\boldsymbol{c}_1 t + \boldsymbol{c}_2)$ *is*

$$(\boldsymbol{A}, \boldsymbol{b}) = (\boldsymbol{0}_d, \frac{1}{k!}\boldsymbol{c}_1^k \sin(\frac{1}{2}kp + \boldsymbol{c}_2)) .$$

3. *The ARF of* $(\boldsymbol{f} \circ \boldsymbol{x})(t) = \cos(\boldsymbol{c}_1 t + \boldsymbol{c}_2)$ *is*

$$(\boldsymbol{A}, \boldsymbol{b}) = (\boldsymbol{0}_d, \frac{1}{k!}\boldsymbol{c}_1^k \cos(\frac{1}{2}kp + \boldsymbol{c}_2)) .$$

4. *The ARF of* $(\boldsymbol{f} \circ \boldsymbol{x})(t) = e^{\boldsymbol{c}_1 t + \boldsymbol{c}_2}$ *is*

$$(\boldsymbol{A}, \boldsymbol{b}) = (\boldsymbol{0}_d, \frac{1}{k!}\boldsymbol{c}_1^k e^{\boldsymbol{c}_2}) .$$

Note that $\boldsymbol{c}_1^k, \sin(\frac{1}{2}kp + \boldsymbol{c}_2), \cos(\frac{1}{2}kp + \boldsymbol{c}_2), e^{\boldsymbol{c}_2}$ here have all component-wise operations without confusion.

## 7.2 ARF of DAEs

Consider nonlinear semi-explicit index-1 DAE system in the form of

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t)) \\ 0 &= \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t)) \end{aligned} \tag{7-7}$$

where $t \in \mathbb{R}$ is the independent variable, $\boldsymbol{x}(t) = [x_1(t), \ldots, x_m(t)]^T$ is the vector of differential variables with $x_i(t) : \mathbb{R} \to \mathbb{R}, i = 1, 2, \ldots m$, $\boldsymbol{y}(t) = [y_1(t), \ldots, y_n(t)]^T$ is the vector of algebraic variables with $y_i(t) : \mathbb{R} \to \mathbb{R}, i = 1, 2, \ldots n$, $\boldsymbol{f} = [f_1(\cdot), f_2(\cdot), \ldots f_m(\cdot)]^T$ is a nonlinear vector field for the differential equations with $f_i(\cdot) : \mathbb{R}^{1+m+n} \to \mathbb{R}, i = 1, 2, \ldots m$, $\boldsymbol{g} = [g_1(\cdot), g_2(\cdot), \ldots g_n(\cdot)]^T$ is a nonlinear vector field for the algebraic equations with $g_i(\cdot) : \mathbb{R}^{1+m+n} \to \mathbb{R}, i = 1, 2, \ldots n$. The index-1 indicates that $\det(\partial \boldsymbol{g} / \partial \boldsymbol{y}) \neq 0$ or the Jacobian $\partial \boldsymbol{g} / \partial \boldsymbol{y}$ is non-singular. We further assume $x_i(t), y_i(t), f_i(\cdot), g_i(\cdot)$ are analytical functions with arbitrary compositional structure.

From theorem 1, there exist unique ARFs $(\boldsymbol{A}_{11}, \boldsymbol{A}_{12}, \boldsymbol{b}_1)$ and $(\boldsymbol{A}_{21}, \boldsymbol{A}_{22}, \boldsymbol{b}_2)$ such that the DAE in (7-7) is converted to

$$\begin{aligned} (k+1)\boldsymbol{X}(k+1) &= \boldsymbol{A}_{11}\boldsymbol{X}(k) + \boldsymbol{A}_{12}\boldsymbol{Y}(k) + \boldsymbol{b}_1 \\ 0 &= \boldsymbol{A}_{21}\boldsymbol{X}(k) + \boldsymbol{A}_{22}\boldsymbol{Y}(k) + \boldsymbol{b}_2 \end{aligned} \tag{7-8}$$

**Definition 3** *(Affine recursion form of DAEs): Equation* (7-8) *is defined as the affine recursion form of nonlinear DAEs* (7-7).

Based on (7-7), there is (7-9) to solve $\boldsymbol{X}(k), \boldsymbol{Y}(k), \forall k \in \mathbb{N}$ in a recursive manner given $\boldsymbol{X}(0) = \boldsymbol{x}(t_0), \boldsymbol{Y}(0) = \boldsymbol{y}(t_0)$, and $\boldsymbol{x}(t_0 + h), \boldsymbol{y}(t_0 + h)$ in a neighborhood of $t = t_0$ is given by (7-10) with $K^{th}$ order accuracy, where $K$ is a positive integer, and $h$ is the step size. Details are presented in Algorithm 9.

$$\begin{aligned} \boldsymbol{Y}(k) &= -\boldsymbol{A}_{22}^{-1}\left(\boldsymbol{A}_{21}\boldsymbol{X}(k) + \boldsymbol{b}_2\right) \\ \boldsymbol{X}(k+1) &= \frac{1}{k+1}\left[\boldsymbol{A}_{11}\boldsymbol{X}(k) + \boldsymbol{A}_{12}\boldsymbol{Y}(k) + \boldsymbol{b}_1\right] \end{aligned} \tag{7-9}$$

$$\begin{aligned} \boldsymbol{x}(t) &= \sum_{k=0}^{K} \boldsymbol{X}(k)(t-t_0)^k + O(h^{K+1}) \\ \boldsymbol{y}(t) &= \sum_{k=0}^{K} \boldsymbol{Y}(k)(t-t_0)^k + O(h^{K+1}) \end{aligned} \tag{7-10}$$

---

**Algorithm 9: Solve DAEs using the proposed affine recursive form**

---

**Input**: initial values $\boldsymbol{x}(t_0), \boldsymbol{y}(t_0)$; desired order $K$

**Output**: $\boldsymbol{x}(t_0+h), \boldsymbol{y}(t_0+h)$

**Offline Stage:**

Derive matrices $\boldsymbol{A}_{11}, \boldsymbol{A}_{12}, \boldsymbol{A}_{21}, \boldsymbol{A}_{22}, \boldsymbol{b}_1, \boldsymbol{b}_2$ using (7-5)-(7-6)

**Online Stage**

1. Initialization: $\boldsymbol{X}(0) = \boldsymbol{x}_0, \boldsymbol{Y}(0) = \boldsymbol{v}_0$
2. Evaluate matrices $\boldsymbol{A}_{11}, \boldsymbol{A}_{12}, \boldsymbol{A}_{21}, \boldsymbol{A}_{22}$
3. Calculate matrix $\boldsymbol{A}_{22}^{-1}$

**for** $k = 1 : K$

   4. Calculate vectors $\boldsymbol{b}_1, \boldsymbol{b}_2$
   5. Calculate $\boldsymbol{X}(k)$: $\boldsymbol{X}(k) = \frac{1}{k}\left[\boldsymbol{A}_{11}\boldsymbol{X}(k-1) + \boldsymbol{A}_{12}\boldsymbol{Y}(k-1) + \boldsymbol{b}_1\right]$
   6. Calculate $\boldsymbol{Y}(k)$: $\boldsymbol{Y}(k) = -\boldsymbol{A}_{22}^{-1}\left(\boldsymbol{A}_{21}\boldsymbol{X}(k) + \boldsymbol{b}_2\right)$

**end**

7. Calculate $\boldsymbol{x}(t_0+h), \boldsymbol{y}(t_0+h)$ using (7-10)

---

From Algorithm 9, the major computation cost lies in the for loop of calculating $\boldsymbol{X}(k), \boldsymbol{Y}(k), k = 1, \ldots K$ from $\boldsymbol{x}(t_0), \boldsymbol{y}(t_0)$. Such an affine recursive problem is studied in [69] and its complexity is $O(Kn^3)$. Besides, the calculation of matrix inverse in step 3 is also $O(n^3)$ in a naive implementation and it only needs to be executed once, regardless of

the selection of $K$. The remaining steps 1, 2, 4, and 7 are neglectable. Therefore, the complexity of the proposed algorithm is $O(Kn^3)$.

## 7.3 Application to Power System DAEs

The widely used power system DAE model is written as

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{v}(t)) \\ \boldsymbol{Y}_{bus}\boldsymbol{v}(t) &= \boldsymbol{i}(\boldsymbol{x}(t),\boldsymbol{v}(t)) \end{aligned} \tag{7-11}$$

where $t \in \mathbb{R}$ is the independent variable with the physical meaning of time, $\boldsymbol{x} \in \mathbb{R}^m$ is the vector of state variables representing the dynamics of power system components such as synchronous generators, motors, and controllers, etc., $\boldsymbol{v} \in \mathbb{R}^n$ is the vector on real and imaginary parts of bus voltages, $\boldsymbol{h} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ is a vector-valued nonlinear function for the differential equations, $\boldsymbol{i} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ the vector-valued function on real and imaginary parts of current injections, and $\boldsymbol{Y}_{bus} \in \mathbb{R}^{n \times n}$ is the constant admittance matrix on the power network.

Note that a power system DAE model usually represents voltages and current injections at buses and admittances of network branches by phasors, i.e. complex numbers. However, the resulting complex equations of the model can be separated into equations respectively on the real and imaginary parts such that only real equations and real analytical functions are considered. Therefore, the $\boldsymbol{v}(t), \boldsymbol{i}(t), \boldsymbol{Y}_{bus}$ are defined below, where $\mathrm{N}, \mathrm{E}$ are the set of buses and branches respectively, $v_{x,i}, v_{y,i}$ are the real and

imaginary parts of voltages at the bus $i$, $i_{x,i}, i_{y,i}$ are the real and imaginary parts of currents at bus $i$ and $G_{ij}, B_{ij}$ are the conductance and susceptance of the branch $(i,j)$ .

$$\boldsymbol{v}(t) = \begin{bmatrix} v_{x,i}, v_{y,i} \end{bmatrix}_{i \in N}; \boldsymbol{i}(t) = \begin{bmatrix} i_{x,i}, i_{y,i} \end{bmatrix}_{i \in N}$$

$$\boldsymbol{Y}_{\boldsymbol{bus}} = \begin{bmatrix} \boldsymbol{Y}_{ij} \end{bmatrix}_{(i,j) \in E}, \text{where } \boldsymbol{Y}_{ij} = \begin{pmatrix} G_{ij} & B_{ij} \\ -B_{ij} & G_{ij} \end{pmatrix}$$

The equation (7-11) can be further written as (7-12)-(7-14), representing the three major components in a power system: the generators, loads, and the power transmission network, where $\boldsymbol{i}_g : \mathbb{R}^{m'} \times \mathbb{R}^{n_g} \to \mathbb{R}^{n_g}$ means the current injection from generators and $\boldsymbol{i}_l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$ represents the current injection from loads, subscripts $g$ and $l$ represent the generators and loads, respectively and $n_g + n_l = n$ .

The model of generators:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{v}_g(t)) \\ \boldsymbol{i}_g &= \boldsymbol{i}_g(\boldsymbol{x}(t), \boldsymbol{v}_g(t)) \end{aligned} \tag{7-12}$$

The model of loads:

$$\boldsymbol{i}_l = \boldsymbol{i}_l(\boldsymbol{v}_l(t)) \tag{7-13}$$

The model of the power transmission network:

$$\begin{aligned} \boldsymbol{Y}_{\boldsymbol{bus}} \boldsymbol{v}(t) &= \boldsymbol{i}(\boldsymbol{x}(t), \boldsymbol{v}(t)) \\ \boldsymbol{v}(t) = \begin{bmatrix} \boldsymbol{v}_g(t) \\ \boldsymbol{v}_l(t) \end{bmatrix}, \quad \boldsymbol{i}(\boldsymbol{x}(t), \boldsymbol{v}(t)) &= \begin{bmatrix} \boldsymbol{i}_g(\boldsymbol{x}(t), \boldsymbol{v}_g(t)) \\ \boldsymbol{i}_l(\boldsymbol{v}_l(t)) \end{bmatrix} \end{aligned} \tag{7-14}$$

The compositional structure of $\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{v}_g(t)),\boldsymbol{i}_g(\boldsymbol{x}(t),\boldsymbol{v}_g(t))$ and $\boldsymbol{i}_l(\boldsymbol{v}_l(t))$ are obtained as follows.

### 7.3.1. ARF of Load Model

For the widely used ZIP load model in transient stability study, the detailed expression of (7-13) is given below, where $v_x, v_y, i_x, i_y$ are algebraic variables, and $v_{t0}, p_0, q_0, a_p, a_q, b_p, b_q, c_p, c_q$ are either system parameters or can be regarded as constants.

$$
\begin{aligned}
i_x &= \frac{1}{v_{t0}^2}\left(p_0 a_p v_x + q_0 a_q v_y\right) + \frac{1}{v_{t0}}\frac{1}{\sqrt{v_x^{\,2} + v_y^{\,2}}}\left(p_0 b_p v_x + q_0 b_q v_y\right) \\
&\quad + \frac{1}{v_x^{\,2} + v_y^{\,2}}\left(p_0 c_p v_x + q_0 c_q v_y\right) \\
i_y &= \frac{1}{v_{t0}^2}\left(-q_0 a_q v_x + p_0 a_p v_y\right) + \frac{1}{v_{t0}}\frac{1}{\sqrt{v_x^{\,2} + v_y^{\,2}}}\left(-q_0 b_q v_x + p_0 b_p v_y\right) \\
&\quad + \frac{1}{v_x^{\,2} + v_y^{\,2}}\left(-q_0 c_q v_x + p_0 c_p v_y\right)
\end{aligned}
$$

Define

$$
\begin{aligned}
\boldsymbol{v}_l &= \left[v_{x,i}, v_{y,i}\right]_{i \in L} & \boldsymbol{i}_l &= \left[i_{x,i}, i_{y,i}\right]_{i \in L} \\
\boldsymbol{u}_{01} &= \boldsymbol{v}_l & \boldsymbol{u}_{11} &= \left[v_{x,i}^{\,2} + v_{y,i}^{\,2}\right]_{i \in L} \\
\boldsymbol{u}_{21} &= \left[\sqrt{v_{x,i}^{\,2} + v_{y,i}^{\,2}}\right]_{i \in L} & \boldsymbol{u}_{31} &= \left[\frac{1}{\sqrt{v_{x,i}^{\,2} + v_{y,i}^{\,2}}}\right]_{i \in L} \\
\boldsymbol{u}_{32} &= \left[\frac{1}{v_{x,i}^{\,2} + v_{y,i}^{\,2}}\right]_{i \in L}
\end{aligned}
$$

where $L$ is the set of load buses, then the compositional function structure of the load model is represented by the DAG in Fig. 47a, with the transformed function with linear structure in Fig. 47b.

Therefore, from Theorem 3, there exists an ARF $(\boldsymbol{A}_l, \boldsymbol{b}_l)$ for the load model such that

$$\boldsymbol{I}_l(\boldsymbol{V}_l(k)) = \boldsymbol{A}_l\boldsymbol{V}_l(k) + \boldsymbol{b}_l \tag{7-15}$$

where the expressions of the ARFs are derived using Theorem 3 with details below. From Fig. 47a, there are three paths between nodes $\boldsymbol{v}_l$ and $\boldsymbol{i}_l$ :

$$\boldsymbol{v}_l \ ® \ \{\boldsymbol{f}_{31}\} ® \ \boldsymbol{i}_l$$
$$\boldsymbol{v}_l \ ® \ \{\boldsymbol{f}_{01}, \boldsymbol{f}_{11}, \boldsymbol{f}_{21}, \boldsymbol{f}_{31}\} ® \ \boldsymbol{i}_l$$
$$\boldsymbol{v}_l \ ® \ \{\boldsymbol{f}_{01}, \boldsymbol{f}_{22}, \boldsymbol{f}_{31}\} ® \ \boldsymbol{i}_l$$

Therefore,

$$\boldsymbol{A}_l = \boldsymbol{A}_{31} + \boldsymbol{A}_{31}\boldsymbol{A}_{21}\boldsymbol{A}_{11}\boldsymbol{A}_{01} + \boldsymbol{A}_{31}\boldsymbol{A}_{22}\boldsymbol{A}_{01} \tag{7-16}$$

Similarly, we can verify there is one path between nodes $\boldsymbol{f}_{21}$ and $\boldsymbol{i}_l$ , one path between $\boldsymbol{f}_{22}$ and $\boldsymbol{i}_l$ , one path between $\boldsymbol{f}_{11}$ and $\boldsymbol{i}_l$ , two paths between $\boldsymbol{f}_{01}$ and $\boldsymbol{i}_l$ , shown below respectively.

$$\boldsymbol{f}_{21} \ ® \ \{\boldsymbol{f}_{31}\} ® \ \boldsymbol{i}_l$$
$$\boldsymbol{f}_{22} \ ® \ \{\boldsymbol{f}_{31}\} ® \ \boldsymbol{i}_l$$
$$\boldsymbol{f}_{11} \ ® \ \{\boldsymbol{f}_{21}, \boldsymbol{f}_{31}\} ® \ \boldsymbol{i}_l$$
$$\boldsymbol{f}_{01} \ ® \ \{\boldsymbol{f}_{11}, \boldsymbol{f}_{21}, \boldsymbol{f}_{31}\} ® \ \boldsymbol{i}_l$$
$$\boldsymbol{f}_{01} \ ® \ \{\boldsymbol{f}_{22}, \boldsymbol{f}_{31}\} ® \ \boldsymbol{i}_l$$

Therefore,

$$\begin{aligned}\boldsymbol{b}_l = \ & \boldsymbol{b}_{31} + \boldsymbol{A}_{31}\boldsymbol{b}_{21} + \boldsymbol{A}_{31}\boldsymbol{b}_{22} + \boldsymbol{A}_{31}\boldsymbol{A}_{21}\boldsymbol{b}_{11} \\ & + \boldsymbol{A}_{31}\boldsymbol{A}_{22}\boldsymbol{b}_{01} + \boldsymbol{A}_{31}\boldsymbol{A}_{21}\boldsymbol{A}_{11}\boldsymbol{b}_{01}\end{aligned} \tag{7-17}$$

In (7-16)-(7-17), the ARFs of simple functions at each node are obtained from the transformation rules in Proposition 1 to 3. After substituting their expressions into (7-16)-(7-17), it is easy to verify that the derived ARF $(\boldsymbol{A}_l, \boldsymbol{b}_l)$ coincides with the formally

linear equations derived in Chapter 3 for the same load model. Moreover, the ARF in (7-16)-(7-17) is easier to implement because it is more compact and does not involve tedious Taylor expansion expression of the compositional functions.

### 7.3.2. ARF of Generator Model

For the widely used six order generator model in power system transient stability study, the detailed expression of (7-12) is below, where $d, w, e_{d1}, e_{q1}, e_{d2}, e_{q2}$ are state variables, $i_d, i_q, v_d, v_q, p_e, v_x, v_y, i_x, i_y$ are algebraic variables, $H, T_{q1}, T_{d1}, T_{q2}, T_{d2}, D, x_d, x_q, x_{d1}, x_{q1}, x_{d2}, x_{q2}, p_m, e_{fd}$ are either system parameters, or can be regarded as constants.

$$\dot{d} = w$$
$$2H\dot{w} = p_m - p_e - Dw$$
$$T_{q1}\dot{e}_{d1} = -\frac{x_q - x_{q2}}{x_{q1} - x_{q2}} e_{d1} + \frac{x_q - x_{q1}}{x_{q1} - x_{q2}} e_{d2}$$
$$T_{d1}\dot{e}_{q1} = -\frac{x_d - x_{d2}}{x_{d1} - x_{d2}} e_{q1} + \frac{x_d - x_{d1}}{x_{d1} - x_{d2}} e_{q2} + e_{fd}$$
$$T_{q2}\dot{e}_{d2} = e_{d1} - e_{d2} + (x_{q1} - x_{q2}) i_q$$
$$T_{d2}\dot{e}_{q2} = e_{q1} - e_{q2} - (x_{d1} - x_{d2}) i_d$$
$$i_d = e_{q2} - v_q, \quad i_q = e_{d2} - v_d, \quad p_e = v_d i_d + v_q i_q$$
$$v_d = v_x \cos d + v_y \sin d, \quad v_q = v_x \sin d - v_y \cos d$$
$$i_x = i_d \sin d - i_q \cos d, \quad i_y = i_d \cos d + i_q \sin d$$

Define

$$\boldsymbol{x} = \left[\delta_i, \omega_i, e_{d1,i}, e_{q1,i}, e_{d2,i}, e_{q2,i}\right]^T \in G \qquad \boldsymbol{y} = \dot{\boldsymbol{x}}$$
$$\boldsymbol{v}_g = \left[v_{x,i}, v_{y,i}\right]^T \in G \qquad \boldsymbol{i}_g = \left[i_{x,i}, i_{y,i}\right]^T \in G$$
$$\boldsymbol{u}_{01} = \boldsymbol{x} \qquad \boldsymbol{u}_{02} = \boldsymbol{v}_g$$
$$\boldsymbol{u}_{11} = \left[\sin\delta_i, \cos\delta_i\right]^T \in G \qquad \boldsymbol{u}_{21} = \left[v_{d,i}, v_{q,i}\right]^T \in G$$
$$\boldsymbol{u}_{31} = \left[i_{d,i}, i_{q,i}\right]^T \in G \qquad \boldsymbol{u}_{41} = \left[p_{e,i}\right]^T \in G$$

where $G$ is the set of generator buses, then the compositional function structure of the generator model is represented by the DAG in Fig. 48a, with the transformed function with linear structure in Fig 4b. Therefore, from Theorem 3, there exists an ARF

$$\left(\boldsymbol{A}_g, \boldsymbol{b}_g\right) = \left(\begin{bmatrix} \bar{\boldsymbol{A}}_{11} & \bar{\boldsymbol{A}}_{12} \\ \bar{\boldsymbol{A}}_{21} & \bar{\boldsymbol{A}}_{22} \end{bmatrix}, \begin{bmatrix} \bar{\boldsymbol{b}}_1 \\ \bar{\boldsymbol{b}}_2 \end{bmatrix}\right)$$

for the generator model such that

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{X}(k), \boldsymbol{V}_g(k)) &= \bar{\boldsymbol{A}}_{11}\boldsymbol{X}(k) + \bar{\boldsymbol{A}}_{12}\boldsymbol{V}_g(k) + \bar{\boldsymbol{b}}_1 \\ \boldsymbol{I}_g(\boldsymbol{X}(k), \boldsymbol{V}_g(k)) &= \bar{\boldsymbol{A}}_{21}\boldsymbol{X}(k) + \bar{\boldsymbol{A}}_{22}\boldsymbol{V}_g(k) + \bar{\boldsymbol{b}}_2 \end{aligned} \tag{7-18}$$

where the expressions of the ARFs $\left(\boldsymbol{A}_g, \boldsymbol{b}_g\right)$ are derived using Theorem 3 with details omitted. It is easy to confirm that the ARF coincides with the formally linear equation in Chapter 3 for the same model, but with a much more compact structure.

### 7.3.3. ARF of Power System DAEs

Finally, after simple matrix operations, the power system model in (7-11) is transformed into the ARF in

$$\begin{aligned} (k+1)\boldsymbol{X}(k+1) &= \boldsymbol{C}_{11}\boldsymbol{X}(k) + \boldsymbol{C}_{12}\boldsymbol{V}(k) + \boldsymbol{d}_1 \\ \boldsymbol{Y}_{\boldsymbol{bus}}\boldsymbol{V}(k) &= \boldsymbol{C}_{21}\boldsymbol{X}(k) + \boldsymbol{C}_{22}\boldsymbol{V}(k) + \boldsymbol{d}_2 \end{aligned} \tag{7-19}$$

where $\boldsymbol{C}_{ij}, i, j \in \{1,2\}$ are functions of $\boldsymbol{X}(0), \boldsymbol{V}(0)$, $\boldsymbol{d}_1, \boldsymbol{d}_2$ are functions of $\boldsymbol{X}(\mathrm{K}) = [\boldsymbol{X}(0), \ldots \boldsymbol{X}(k-1)]$ and $\boldsymbol{V}(\mathrm{K}) = [\boldsymbol{V}(0), \ldots \boldsymbol{V}(k-1)]$ with expressions below.

$$\boldsymbol{C}_{11} = \bar{\boldsymbol{A}}_{11}; \boldsymbol{C}_{12} = [\bar{\boldsymbol{A}}_{12} \quad \boldsymbol{0}]; \boldsymbol{d}_1 = \bar{\boldsymbol{b}}_1$$

$$\boldsymbol{C}_{21} = \begin{bmatrix} \bar{\boldsymbol{A}}_{21} \\ \boldsymbol{0} \end{bmatrix}; \boldsymbol{C}_{22} = \begin{bmatrix} \bar{\boldsymbol{A}}_{22} \\ \boldsymbol{A}_l \end{bmatrix}; \boldsymbol{d}_2 = \begin{bmatrix} \bar{\boldsymbol{b}}_2 \\ \boldsymbol{b}_l \end{bmatrix}$$

## 7.4 Conclusion

This chapter presents a theoretical study of nonlinear DAEs. A novel affine recursion form of nonlinear DAEs is proposed in this chapter. The existence, uniqueness, and derivation of ARFs for nonlinear DAEs are studied and an ARF-based algorithm is designed to derive the semi-analytical solutions of nonlinear DAEs in the form of arbitrary high-order Taylor series. Application to power system DAEs demonstrates the potential of the ARF for the stability analysis and control of nonlinear DAEs in high-dimensional networked dynamical systems.

# CHAPTER 8

# CONCLUSION

This work aims at developing efficient and robust power system simulation algorithms. Specifically, this work aims at addressing the fundamental computational challenges of solving high dimensional nonlinear differential algebraic equation model in transient stability analysis, and its two variations, i.e., ordinary differential equation model in transient stability simulation under the assumption of constant impedance load, and the algebraic AC power flow equation model in voltage stability analysis. The contribution of this work are as follows.

First, this dissertation proposes a novel dynamic simulation approach based on DT, which is a mathematical tool and can effectively find an approximate solution of a set of nonlinear ODEs. This work, for the first time, introduces DT to power system studies. By assuming the solution of a set of power system DEs as a power series in time, the proposed approach utilizes DT to calculate series coefficients efficiently by means of a set of transform rules designed for nonlinear functions involved in power system models instead of directly computing the high-order derivatives in DEs. These transform rules are proved for representative power system models including a detailed synchronous machine model involving trigonometric functions and a practical exciter model with the exponential and square root functions. The dissertation also proposed a DT-based dynamic simulation scheme that allows significantly prolonged time steps to reduce the overall simulation time compared to a traditional numerical approach.

Second, this work proposes a novel non-iterative method to solve the DAE model of a large-scale power system using the DT method. First, we derive the DTs of the algebraic network equations with current injections. Then, we prove that current injections and bus voltages which are coupled by the original nonlinear network equations, satisfy a formally linear equation in terms of their power series coefficients after DT. Further, a non-iterative algorithm is designed to analytically solve both state variables and non-state variables by power series of time. Simulation results show the proposed method is fast and reliable compared to traditional methods.

Third, to trace solution curves of power flow equations more efficiently, this work proposes a novel dynamized power flow (DPF) method that extends the power flow model into a fictitious dynamic system, called a “dynamized” power flow model, by adding a differential equation about a fictitious time, and then solve the complete time-domain trajectory of the dynamic system instead of repeatedly solving power flow equations for a series of conditions. The DT method is applied to solve the dynamized model, named as dynamized power flow method. This work proves that the nonlinear AC power flow equations are converted to formally linear equations after DT, and further designs an efficient algorithm to solve the time domain trajectory without numerical iterations. Case studies on several test systems including a 2383-bus system demonstrate the accuracy, computational complexity and time performance of the proposed approach compared with a CPF solver.

Fourth, this dissertation examined the feasibility of combining the DT method and the Parareal algorithm to further speed up the dynamic simulation. A DT- based variable-

order variable-step variable-window adaptive parareal method is proposed for temporal parallelization of power system simulation with greatly enhanced convergence performance and efficiency. The proposed method integrates the temporal parallelization capability of the Parareal method and the highly adaptive feature of the DT method. Extensive simulations on a 39-bus system and a 2383-bus system demonstrate the effectiveness of the proposed approach.

Fifth, this work proposes a novel switching control strategy which predicts the safety of a frequency response right after a disturbance by evaluating the derived semi-analytical solutions of system frequency response model over a certain post-disturbance period of interest, and activates frequency support mode only when the frequency response is predicted as unsafe. The rationale of the proposed strategy is real time evaluation of an offline obtained semi-analytical solution on frequency responses using real-time measurements. Such a semi-analytical solution is in form of ultra-high order Taylor series derived by the DT method on the differential equation model of the system. The ultra-high order nature of the solution enables its largely extended convergence region to cover the frequency response period of interest, so that the frequency response of a WTG can be accurately predicted when a disturbance is detected and the WTG provides frequency support only for an unsafe response, thus avoiding the unnecessary switches. The case studies on a 4-bus power system and a New England 10-machine 39-bus system show the effectiveness of the proposed strategy.

Finally, this work proposes a DT-based affine recursion form (ARF) of general nonlinear DAEs. The advantage of ARF is that nonlinear DAEs are converted to formally

linear equations about Taylor series coefficients, which enables straightforward calculation of semi-analytical solutions of the nonlinear DAEs in the form of arbitrary high-order Taylor series. Since practical DAE models often contain compositional functions with complicated compositional structures, this work first studies the ARF of generic compositional functions, including its existence, uniqueness, and propagation of the ARF over a compositional function structure to derive ARFs of compositional functions from simple functions. After that, this work derives the ARF of nonlinear DAEs and designs an ARF-based DAE solution algorithm. Finally, the work applies the proposed method to derive the ARF of a detailed power system DAE model.

# LIST OF REFERENCES


[1] J. H. Eto and R. J. Thomas, "Computational needs for the next generation electric grid," *Department of Energy*, vol. 593, 2011.

[2] A. Chakrabortty, & A. Bose, "Smart grid simulations and their supporting implementation methods," *Proceedings of the IEEE*, vol. 105, no. 11, pp. 2220-2243, Nov. 2017

[3] P. Kundur et al., "Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions," *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1387–1401, Aug. 2004.

[4] http://mydocs.epri.com/docs/PublicMeetingMaterials/1202/MVNQQ2YVLLT/Lei_Wang.pdf

[5] J. T. G. Weckesser, H. Jóhannsson, and J. Østergaard, "On-line Dynamic Security Assessment in Power Systems," 2014.

[6] Z. Huang, S. Jin, and R. Diao, "Predictive dynamic simulation for large-scale power systems through high-performance computing," in *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, 2012, pp. 347-354.

[7] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-hill New York, 1994.

[8] R. Diao *et al.*, "On Parallelizing Single Dynamic Simulation Using HPC Techniques and APIs of Commercial Software," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 2225-2233, 2017.

[9] J. H. Chow, *Power system coherency and model reduction*. Springer, 2013.

[10] D. Osipov, K. Sun, "Adaptive nonlinear model reduction for fast power system simulation," *IEEE Transactions on Power Systems*, vo. 33, no. 6, pp. 6746-6754, Nov. 2018.

[11] F. Milano, *Power system modelling and scripting*. Springer Science & Business Media, 2010.

[12] F. Milano, "On current and power injection models for angle and voltage stability analysis of power systems," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2503–2504, May 2016.

[13] S. Jin, Z. Huang, R. Diao, D. Wu, and Y. Chen, "Comparative implementation of high performance computing for power system dynamic simulations," *IEEE Transactions on Smart Grid,* vol. 8, no. 3, pp. 1387-1395, 2017.

[14] H. A. Pulgar-Painemal, "Wind farm model for power system stability analysis," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Illinois Urbana-Champaign, Champaign, IL, USA, 2010.

[15] J. Morren, S. W. H. de Haan, W. L. Kling, and J. A. Ferreira, "Wind turbines emulating inertia and supporting primary frequency control," *IEEE Trans. Power Syst*., vol. 21, no. 1, pp. 433–434, Feb. 2006

[16] S. Wang, & K. Tomsovic, "A novel active power control framework for wind turbine generators to improve frequency response," *IEEE Trans. Power Syst.,* vol. 33, no. 6, pp. 6579-6589, Nov. 2018

[17] J. de Almeida and R. G. Lopes, "Participation of doubly fed induction wind generators in system frequency regulation," *IEEE Trans. Power Syst.,* vol. 22, no. 3, pp. 944–950, Aug. 2007

[18] Y. Zhang, K. Tomsovic, S. M. Djouadi, and H. Pulgar-Painemal, "Hybrid Controller for Wind Turbine Generators to Ensure Adequate Frequency Response in Power Networks," *IEEE J. Emerg. Sel. Top. Circuits Syst*., vol. 7, no. 3, pp. 359–370, Sep. 2017.

[19] Y. Zhang, M. E. Raoufat, K. Tomsovic, and S. M. Djouadi, "Set theory-based safety supervisory control for wind turbines to ensure adequate frequency response," *IEEE Trans. Power Syst*., vol. 34, no. 1, pp. 680-692, Jan. 2019.

[20] A. Arif, Z. Wang, J. Wang, B. Mather, H. Bashualdo, and D. Zhao. "Load Modeling—A Review." *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5986-5999, Nov. 2018.

[21] G. Gurrala, et al, "Parareal in time for fast power system dynamic simulations," *IEEE Trans. Power Syst*., vol. 31, no. 3, pp. 1820-1830, Jul. 2015.

[22] N. Duan, et al, "Applying reduced generator models in the coarse solver of parareal in time parallel power system simulation," in Proc. *IEEE PES Innovative Smart Grid Technologies Conference Europe*, 2016, pp. 1-5.

[23] S. Zadkhast, J. Jatskevich, E. Vaahedi, "A multi-decomposition approach for accelerated time-domain simulation of transient stability problems," *IEEE Trans. on Power Syst*., vol. 30, no. 5, pp. 2301-2311, Sep. 2015.

[24] P. Aristidou, et al, "Power system dynamic simulations using a parallel two-level Schur-complement decomposition," *IEEE Trans. Power Syst*., vol. 31, no. 5, pp. 3984-3995, Sep. 2016.

[25] P. Aristidou, et al, "Dynamic Simulation of Large-Scale Power Systems Using a Parallel Schur-Complement-Based Decomposition Method," *IEEE Trans. Parallel & Distributed Syst.*, vol. 25, no. 10, pp. 2561-2570, 2014.

[26] Y. Liu, Q. Jiang, "Two-stage parallel waveform relaxation method for large-scale power system transient stability simulation," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 153-162, Jan. 2016.

[27] V. Jalili-Marandi, V. Dinavahi, "Instantaneous relaxation-based real-time transient stability simulation," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1327-1336, 2009.

[28] V. Jalili-Marandi, et al, "Large-Scale Transient Stability Simulation of Electrical Power Systems on Parallel GPUs," *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1255-1266, 2012.

[29] M. A. Tomim, et al, "Parallel transient stability simulation based on multi-area Thévenin equivalents," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1366-1377, 2017.

[30] M. A. Tomim, J. R. Martí, T. De Rybel, L. Wang, and M. Yao, "MATE network tearing techniques for multiprocessor solution of large power system networks," in *Power and Energy Society General Meeting, 2010 IEEE*, 2010, pp. 1-6: IEEE.

[31] I. Konstantelos, et al, "Implementation of a massively parallel dynamic security assessment platform for large-scale grids," *IEEE Trans. on Smart Grid*, vol. 8, no. 3, pp. 1417-1426, May 2017.

[32] N. Duan, K. Sun, "Power system simulation using the multi-stage Adomian Decomposition Method," *IEEE Trans. on Power Syst*., vol. 32, no. 1, pp 430-441, Jan. 2017.

[33] G. Gurrala, D. L. Dinesha, A. Dimitrovski, P. Sreekanth, S. Simunovic, M. Starke, "Large multi-machine power system simulations using multi-Stage Adomian Decomposition," *IEEE Trans. on Power Syst*., vol. 32, no. 5, pp. 3594-3606, Sep. 2017.

[34] B. Park, K. Sun, A. Dimitrovski, Y. Liu and S. Simunovic, "Examination of Semi-Analytical Solution Methods in the Coarse Operator of Parareal Algorithm for Power System Simulation," in *IEEE Transactions on Power Systems*.

[35] E. Abreut, B. Wang, and K. Sun, "Semi-analytical fault-on trajectory simulation and its application in direct methods," in Proc. *Power and Energy Society General Meeting*, 2017, pp. 1-5.

[36] B. Wang, N. Duan and K. Sun, "A Time–Power Series-Based Semi-Analytical Approach for Power System Simulation," in *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 841-851, March 2019.

[37] C. Liu, B. Wang, K. Sun, "Fast power system simulation using semi analytical solutions based on Pade Approximants," in Proc. *Power and Energy Society General Meeting*, 2017, pp. 1-5.

[38] R. Yao, K. Sun and F. Qiu, "Vectorized Efficient Computation of Padé Approximation for Semi-Analytical Simulation of Large-Scale Power Systems," in *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3957-3959, Sept. 2019.

[39] C. Liu, B. Wang and K. Sun, "Fast Power System Dynamic Simulation Using Continued Fractions," in *IEEE Access*, vol. 6, pp. 62687-62698, 2018.

[40] R. Yao, K. Sun, D. Shi and X. Zhang, "Voltage Stability Analysis of Power Systems With Induction Motors Based on Holomorphic Embedding," in *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1278-1288, March 2019.

[41] R. Yao, Y. Liu, K. Sun, F. Qiu and J. Wang, "Efficient and Robust Dynamic Simulation of Power Systems With Holomorphic Embedding," in *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 938-949, March 2020.

[42] D. L. Dinesha and G. Gurrala, "Application of Multi-Stage Homotopy Analysis Method for Power System Dynamic Simulations," in *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2251-2260, May 2019.

[43] Y. Liu, K. Sun, R. Yao and B. Wang, "Power System Time Domain Simulation Using a Differential Transformation Method," in *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3739-3748, Sept. 2019.

[44] Y. Liu and K. Sun, "Solving Power System Differential Algebraic Equations Using Differential Transformation," in *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 2289-2299, May 2020.

[45] Y. Liu, K. Sun and J. Dong, "A Dynamized Power Flow Method Based on Differential Transformation," in *IEEE Access*, vol. 8, pp. 182441-182450, 2020.

[46] Y. Liu, K. Sun, " Differential Transformation of a Motor Load Model for Time-Domain Simulation," *arXiv preprint arXiv: 1908.09801*, 2019.

[47] Y. Liu, K. Sun, "Differential Transformation of Nonlinear Power Flow Equations," *arXiv preprint arXiv:2004.08017*, 2020

[48] Y. Liu, Y. Zhang, K. Sun, X. Zhao, " Adaptive Switching Control of Wind Turbine Generators for Necessary Frequency Response," *arXiv preprint arXiv: 2005.14562*, 2020

[49] E. Pukhov, G. Georgii, "Differential transforms and circuit theory," *International Journal of Circuit Theory and Applications*, vol. 10, no. 3, pp. 265-276, Jun. 1982.

[50] I. H. A. Hassan, "Application to differential transformation method for solving systems of differential equations," *Applied Mathematical Modelling*, vol. 32, no. 12, pp. 2552-2559, Oct. 2007.

[51] M. Ghafarian, A. Ariaei, "Free vibration analysis of a system of elastically interconnected rotating tapered Timoshenko beams using differential transform method," *International Journal of Mechanical Sciences*, vol. 107, pp. 93-109, 2016.

[52] M. Matinfar, S. R. Bahar, M. Ghasemi, "Solving the Lienard equation by differential transform method," *World Journal of Modelling and Simulation*, vol. 8, no. 2, pp. 142-146, 2012.

[53] L. Xie, C. Zhou, S. Xu, "An effective numerical method to solve a class of nonlinear singular boundary value problems using improved differential transform method," *SpringerPlus*, vol. 5, no. 1, pp. 1066, Jul. 2016.

[54] F. Kenmogne, "Generalizing of differential transform method for solving nonlinear differential equations," *Applied & Computational Mathematics*, vol. 4, no. 1, pp. 1000196, Jan. 2015.

[55] V. S. Erturk, Z. M. Odibat, S. Momani, "The multi-step differential transform method and its application to determine the solutions of non-linear oscillators," *Advances in Applied Mathematics and Mechanics*, vol. 4, no. 4, pp. 422-438, Aug. 2012.

[56] C. W. Berta, H. Zeng, "Analysis of axial vibration of compound bars by differential transformation method," *Journal of Sound and Vibration*, vol. 275, no. 3, pp. 641-647, Aug. 2004.

[57] Z. Odibat, S. Momani, "A generalized differential transform method for linear partial differential equations of fractional order," *Applied Mathematics Letters*, vol. 21, no. 2, pp. 194-199, Feb. 2008.

[58] S. Ghosh, A. Roy, D. Roy, "An adaptation of adomian decomposition for numericanalytic integration of strongly nonlinear and chaotic oscillators," *Comput. Methods Appl. Mech. Eng*., vol. 196, pp. 1133–1153, 2007.

[59] G. Adomian, "A review of the decomposition method and some recent results for nonlinear equations," *Comput. Math. Appl*., vol. 21, no. 5, pp. 101–127, 1991.

[60] R. D. Zimmerman, et al, "Matpower: Steady-state operations, planning and analysis tools for power systems research and education," *IEEE Tran. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2011.

[61] B. Wang, K. Sun, W. Kang, "Nonlinear modal decoupling of multi-oscillator systems with applications to power systems," *IEEE Access*, vol. 6, pp. 9201-9217, Dec. 2017

[62] V. Ajjarapu, C. Christy, "The continuation power flow: a tool for steady state voltage stability analysis," *IEEE Tran. Power Syst.*, vol. 7, no. 1, pp. 416-423, Feb. 1992.

[63] H. Song, S. Kim, B. Lee, S. H. Kwon, V. Ajjarapu, "Determination of interface flow margin using the modified continuation power flow in voltage stability analysis," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 148, no. 2, pp. 128-132, 2001.

[64] A. Dukpa, et al, "Application of continuation power flow method in radial distribution systems," *Electric Power Systems Research*, vol. 79, no. 11, pp. 1503-1510, 2009.

[65] X.P. Zhang, P. Ju, E. Handschin, "Continuation three-phase power flow: A tool for voltage stability analysis of unbalanced three-phase power systems," *IEEE Tran. Power Syst.*, vol. 20, no. 3, pp. 1320-1329, 2005.

[66] H.D. Chiang, et al, "CPFLOW: A practical tool for tracing power system steady-state stationary behavior due to load and generation variations," *IEEE Tran. Power Syst.*, vol. 10, no. 2, pp. 623-634, 1995.

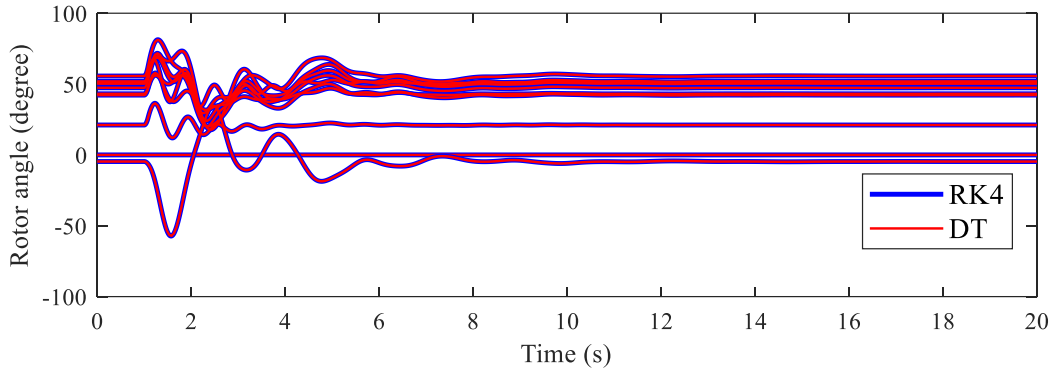
[67] M. J. Gander, S. Vandewalle, “Analysis of the parareal time-parallel time-integration method,” SIAM Journal on Scientific Computing, vol. 29, no. 2, pp. 556-578, 2007.

[68] W. Kang and Q. Gong, "Neural Network Approximations of Compositional Functions With Applications to Dynamical Systems," arXiv preprint arXiv:2012.01698, 2020.

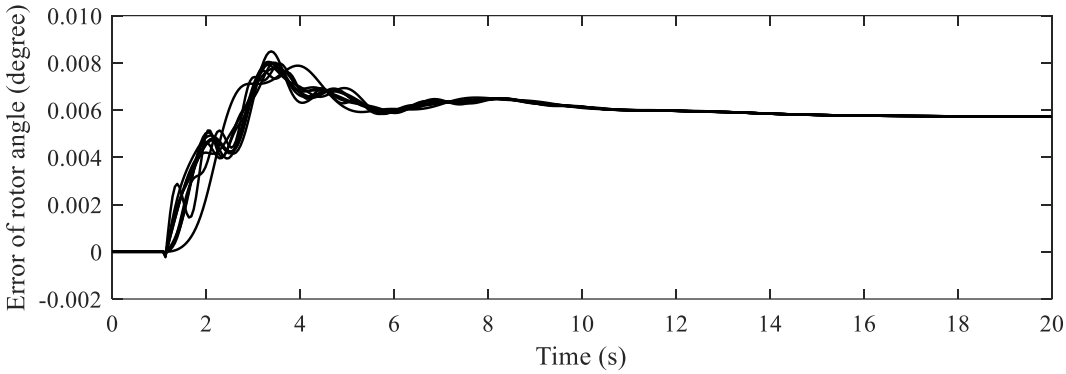
[69] J. Kienitz and N. Nowaczyk, "Affine Recursion Problem and a General Framework for Adjoint Methods for Calculating Sensitivities for Financial Instruments," SSRN 1957082, Nov. 2011.

# APPENDICES

# APPENDIX A – Figures

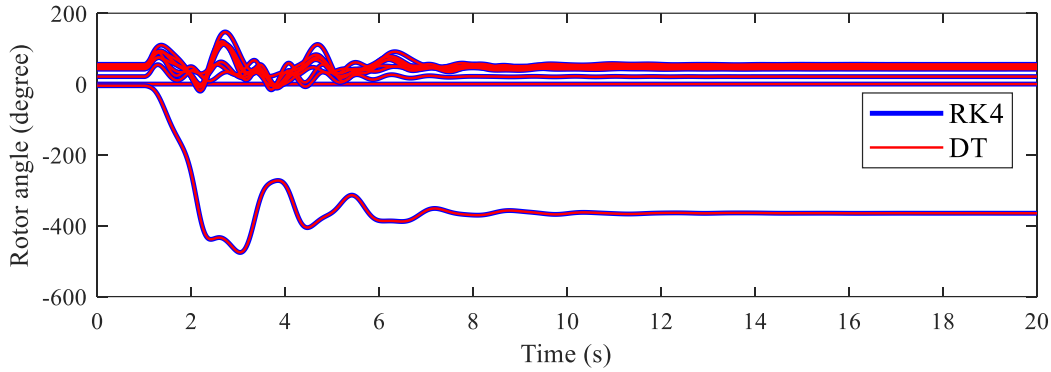



(a) rotor angles

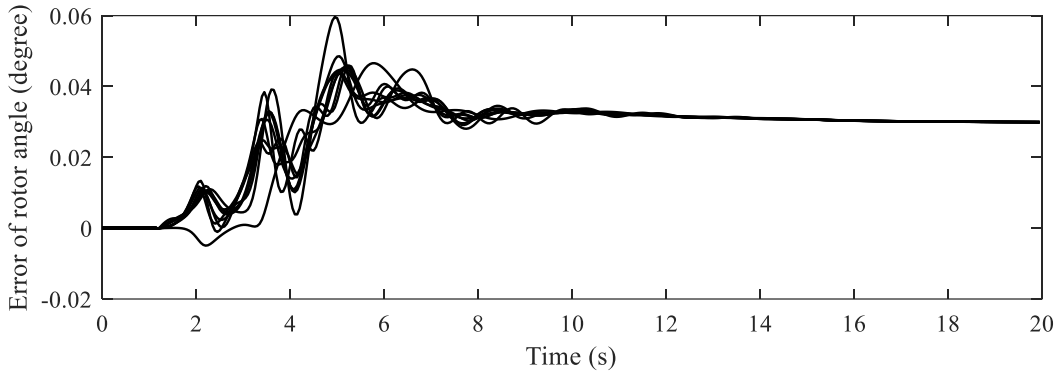



(b) error of rotor angels

Figure 1. 39-bus system under a stable contingency

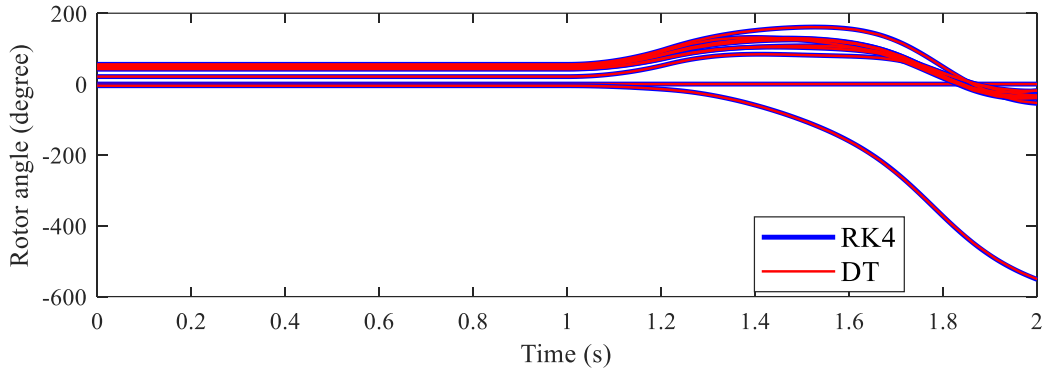



(a) rotor angles

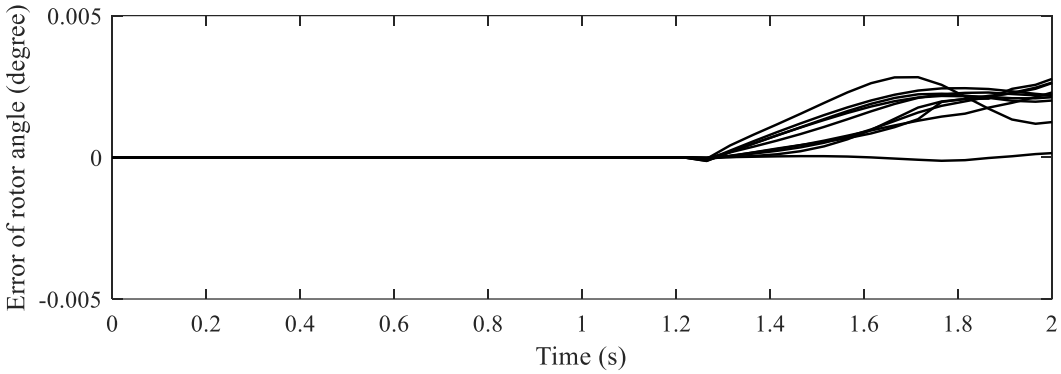



(b) errors of rotor angles

Figure 2. 39-bus system under a marginal stable contingency.

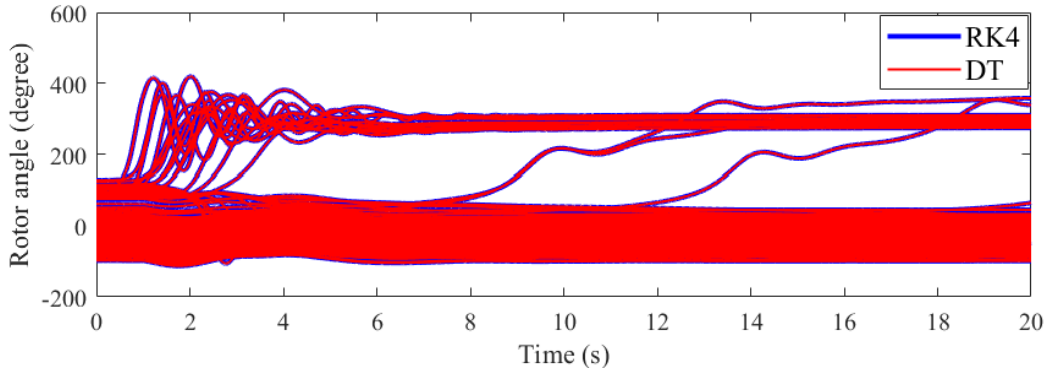



(a) rotor angles

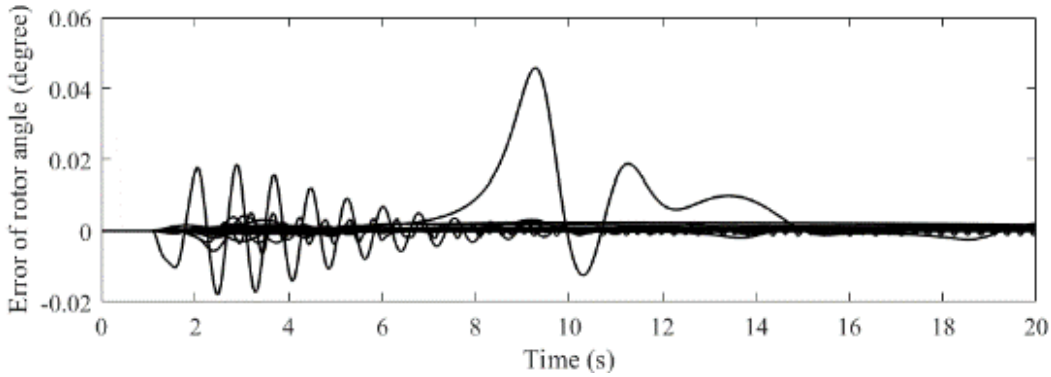



(b) errors of rotor angles

Figure 3. 39-bus system under marginal unstable contingency.

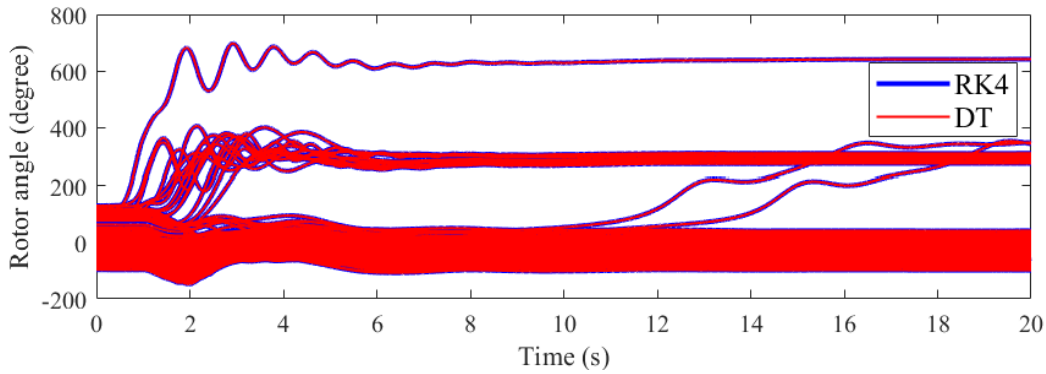



(a) rotor angles

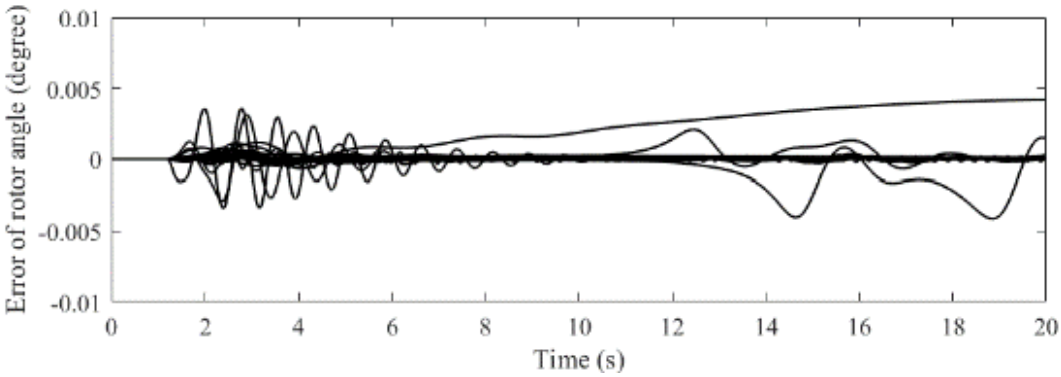



(b) errors of rotor angles

Figure 4. 2383-bus system under a stable contingency.

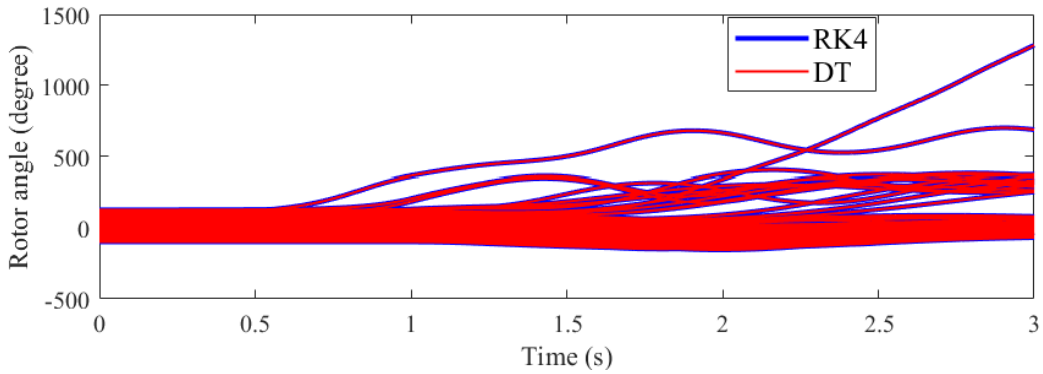



(a) rotor angles

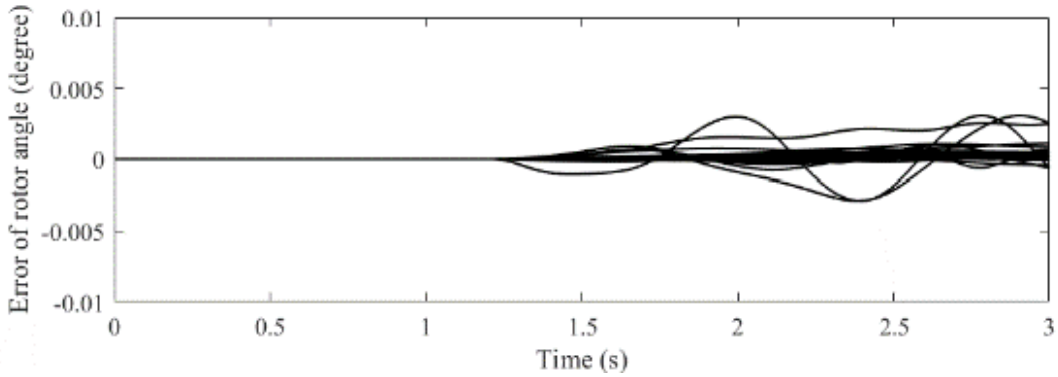



(b) errors of rotor angles

Figure 5. 2383-bus system under a marginally stable contingency.




(a) rotors




(b) errors of rotor angles

Figure 6. 2383-bus system under a marginally unstable contingency.

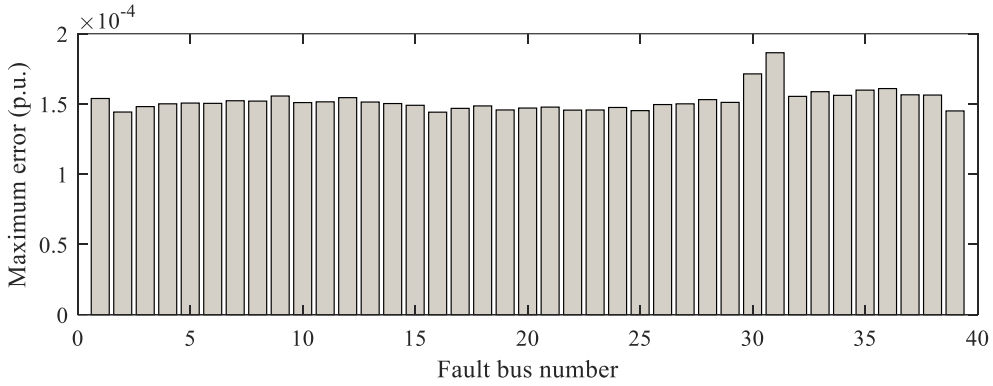



Figure 7. Accuracy of the DT method when faults occur at all buses.

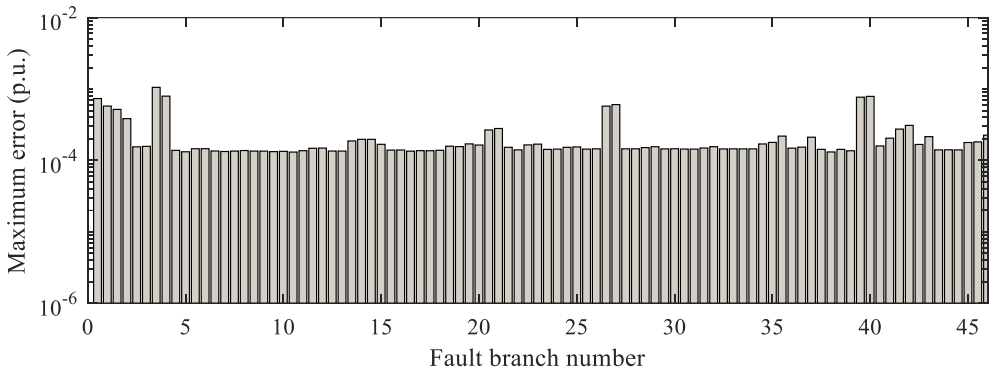



Figure 8. Accuracy of the DT method when faults occur at all branches.

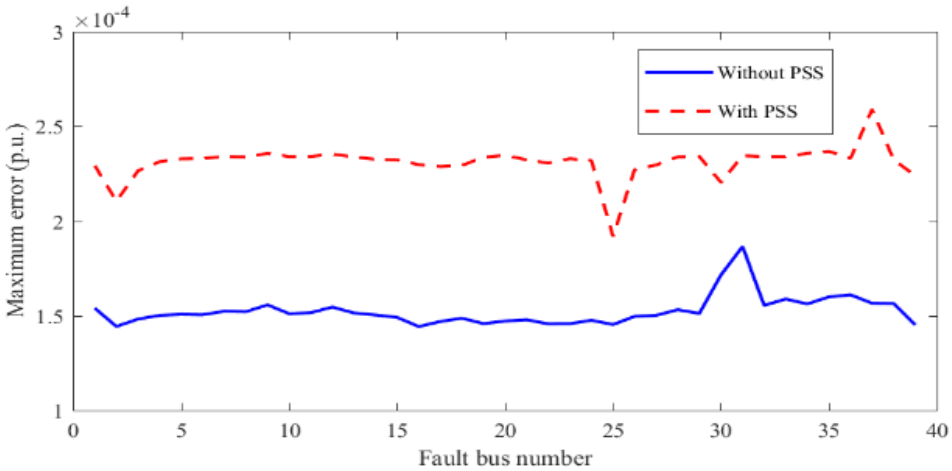



Figure 9. Accuracy of the DT method with and without PSS.

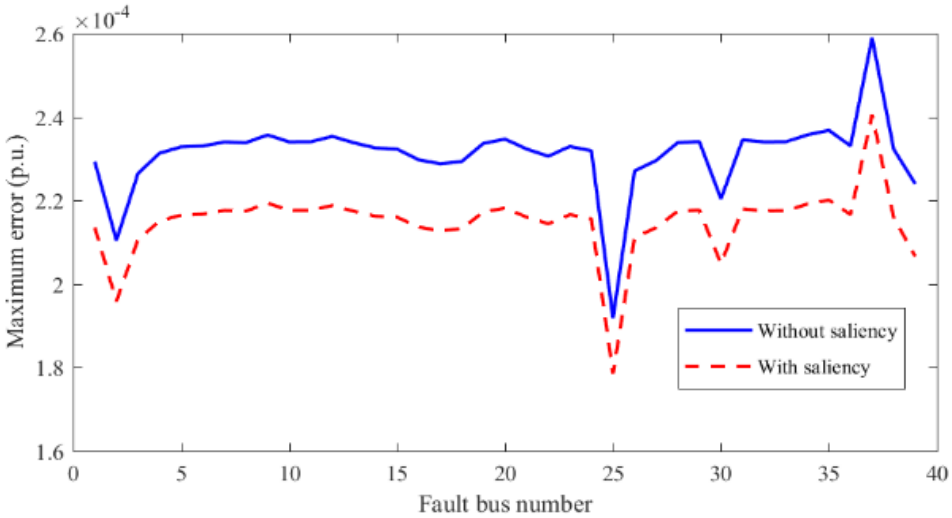
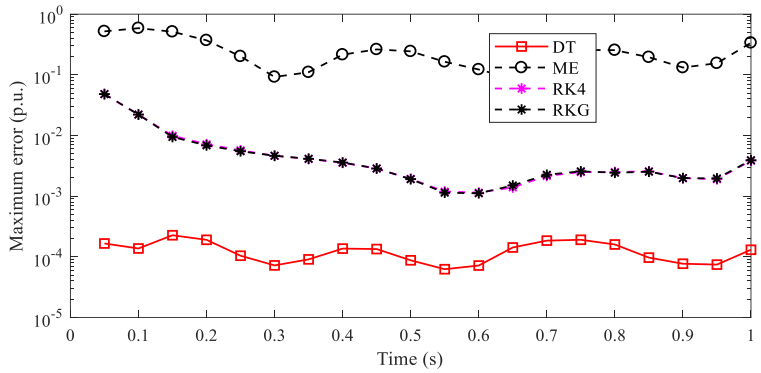
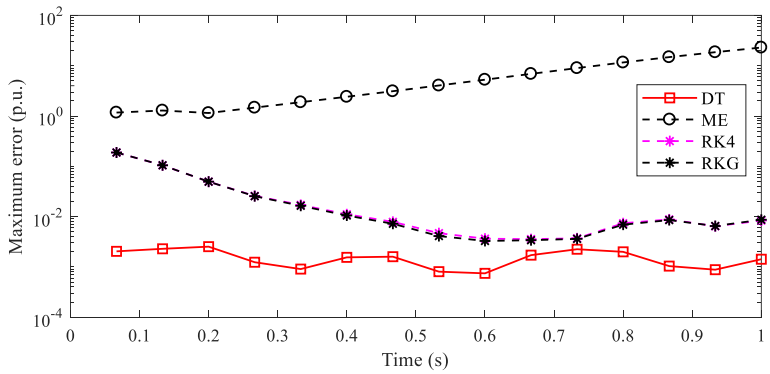


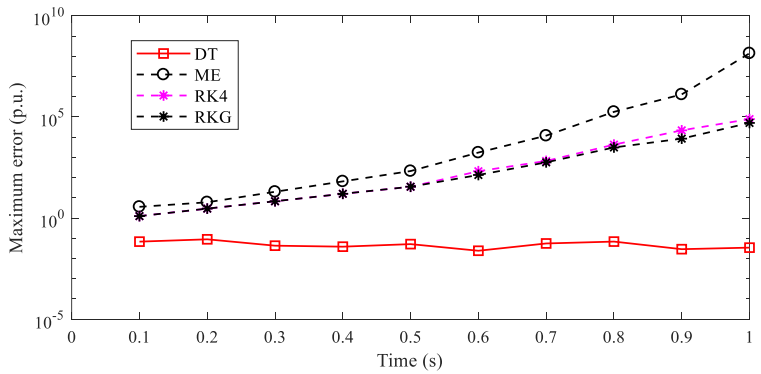

Figure 10. Accuracy of the DT method with and without saliency.




(a) time step = 0.050 s




(b) time step =0.067 s




(c) time step = 0.100s

Figure 11. Error propagation for the three scenarios with different time steps

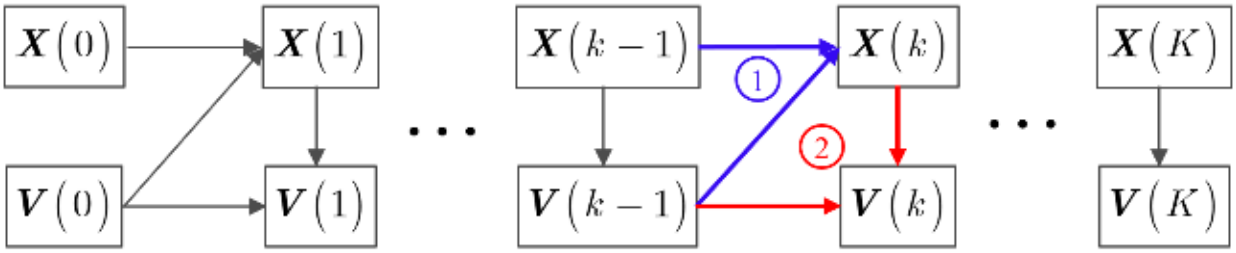
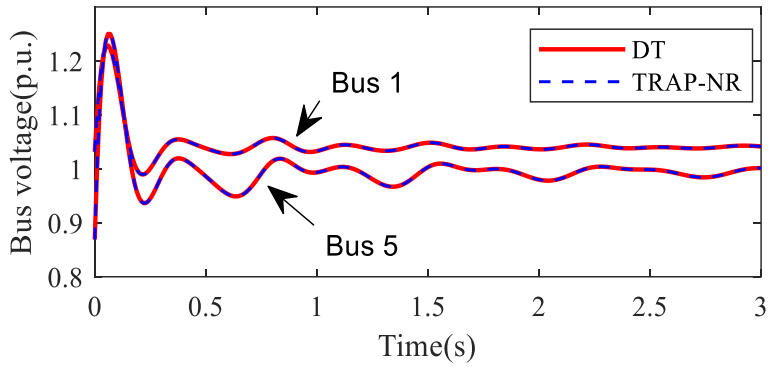
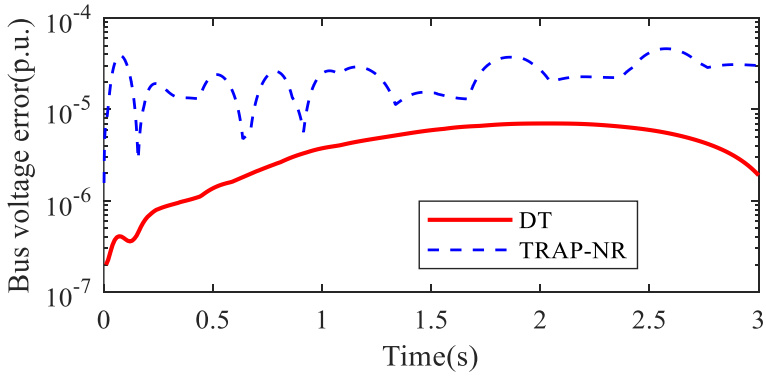



Figure 12. Recursive process to solve power series coefficients

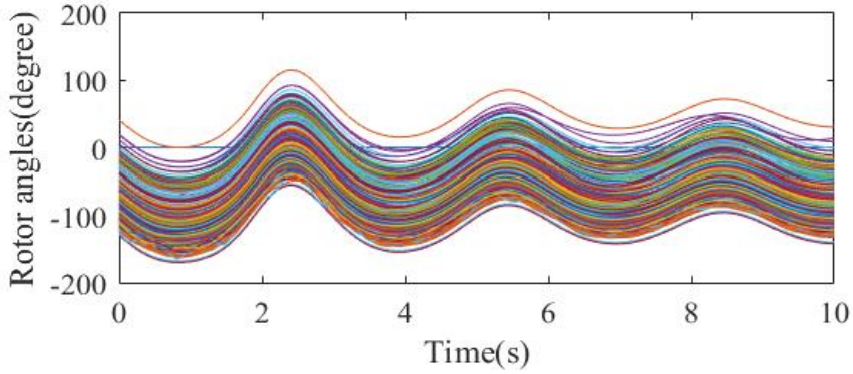



(a)

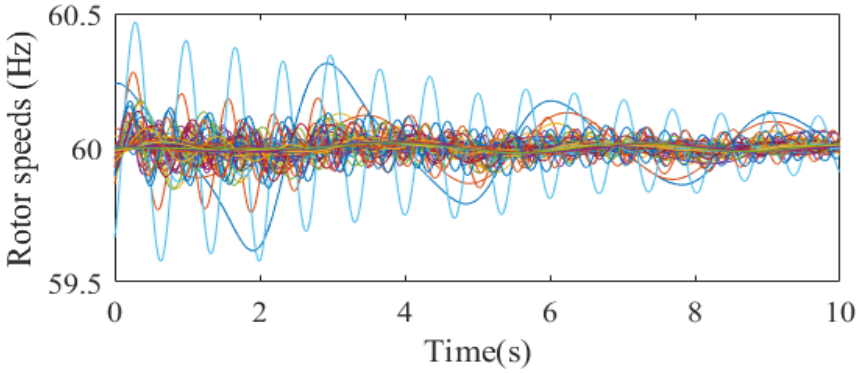



(b)

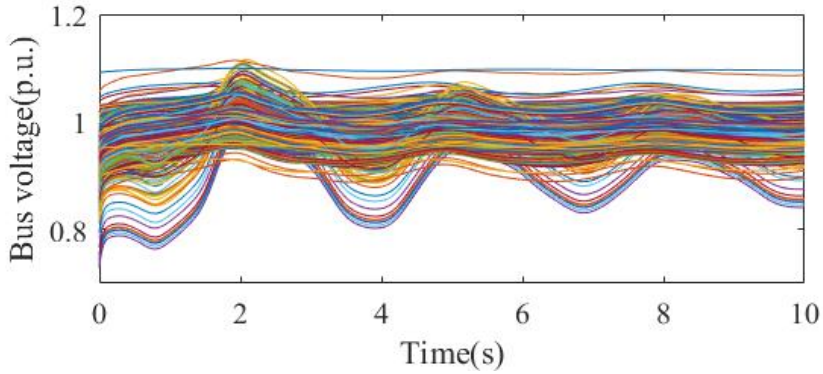
Figure 13. Trajectories and errors of bus voltages for the 9-bus system (a) Voltage trajectories at bus 1 and bus 5 (b) Maximum voltage error of all 9 buses




(a) Rotor angles

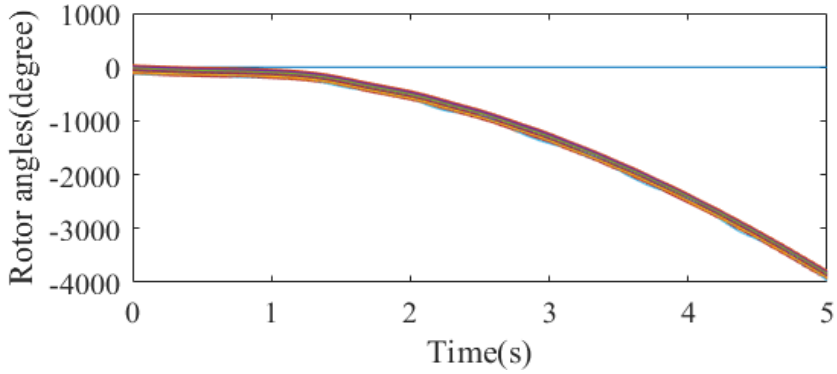



(b) Rotor speeds

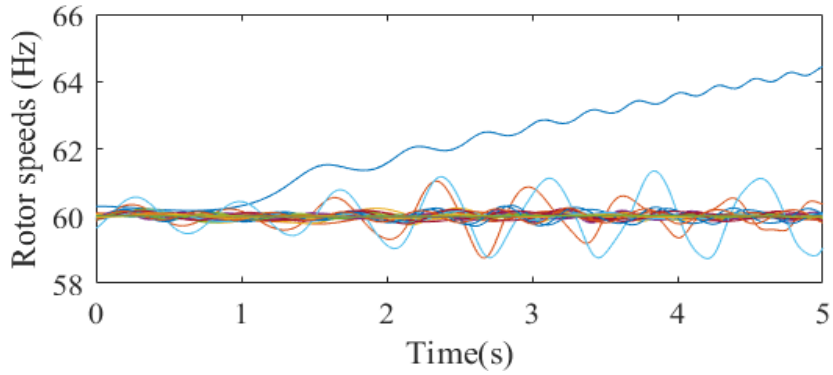



(c) Bus voltages

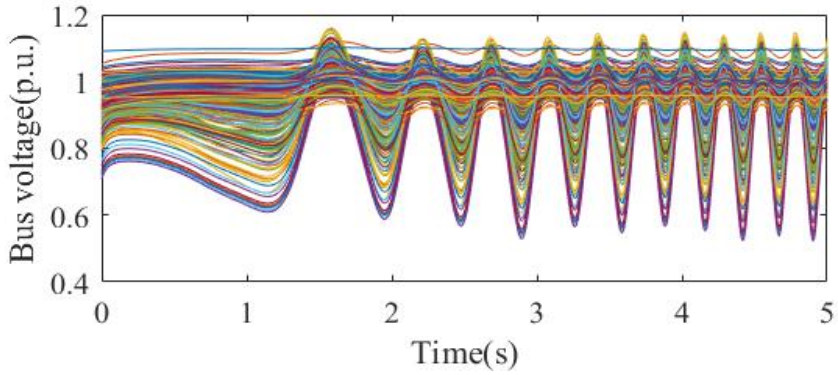
Figure 14. Trajectories of the stable scenario for the 2383-bus system




(a) Rotor angles




(b) Rotor speeds




(c) Bus voltages

Figure 15. Trajectories of the unstable scenario for the 2383-bus system

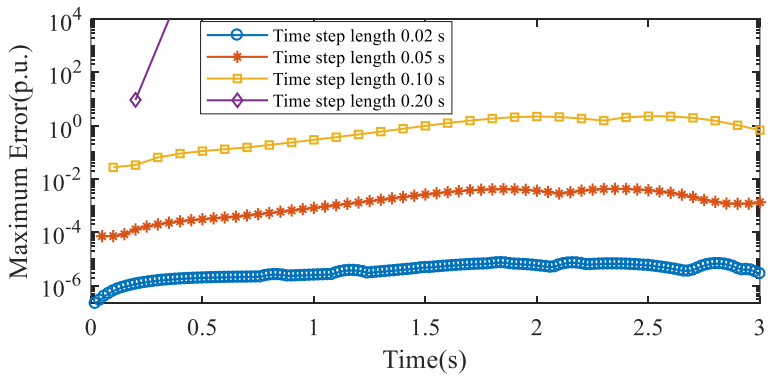



Figure 16. Error propagation under different time step lengths

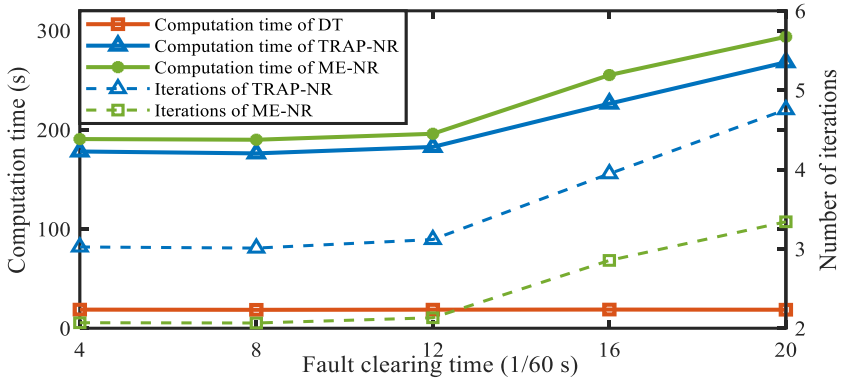



Figure 17. Robustness against different disturbances

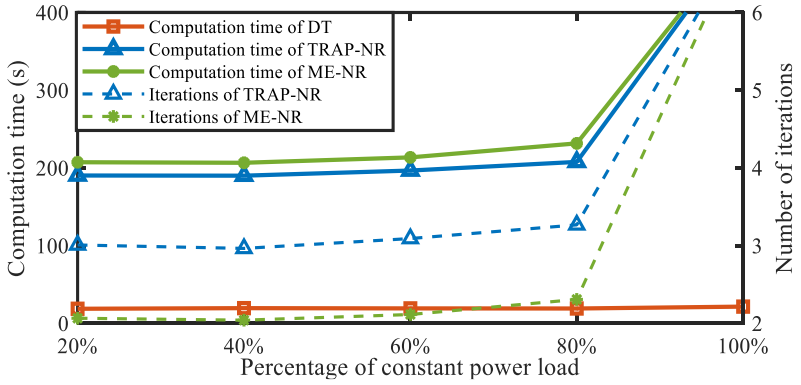



Figure 18. Robustness against different percentages of constant power load

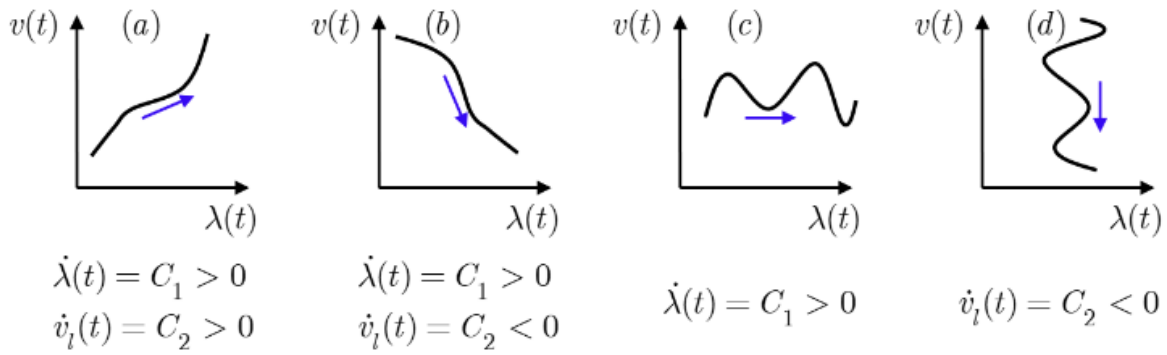



Figure 19. Illustration of the two dynamized formulations for tracing curve segments of power flow equations

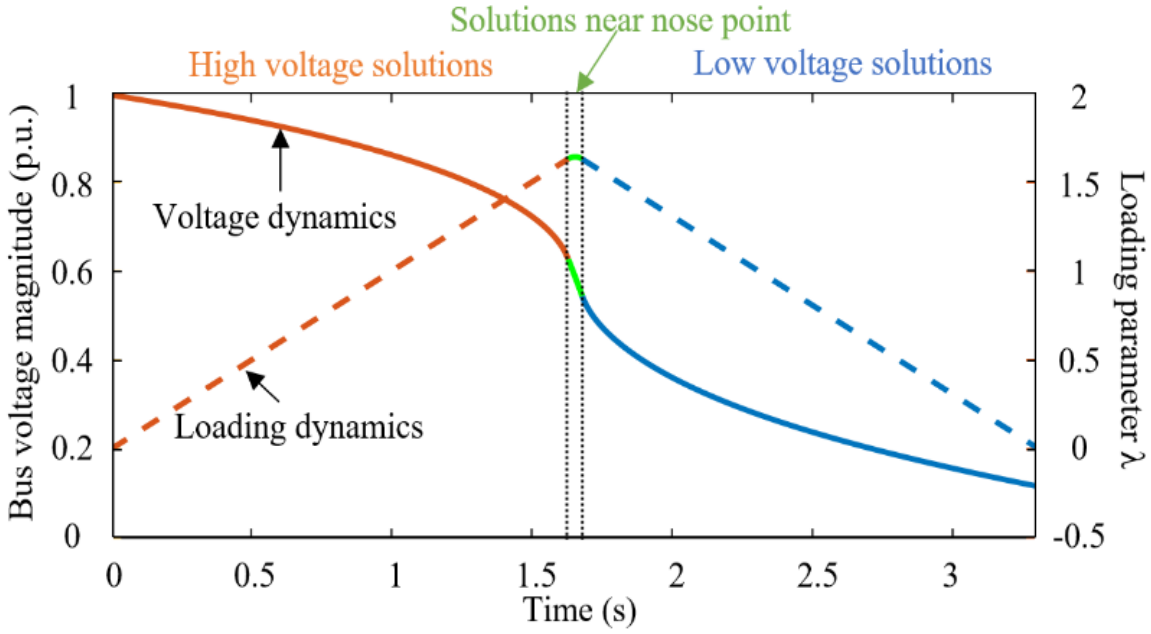



Figure 20. Time domain trajectory of the dynamized power flow model

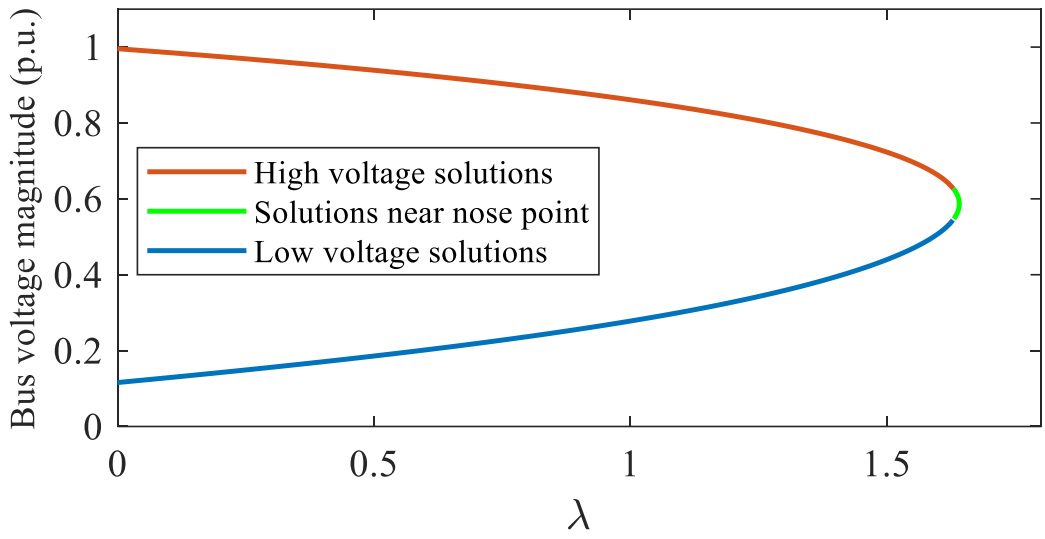
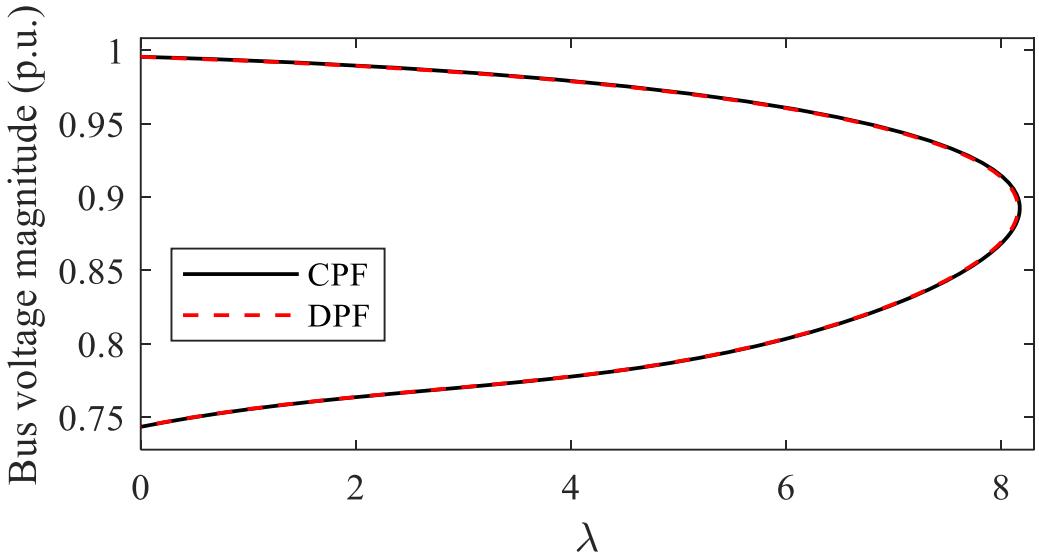
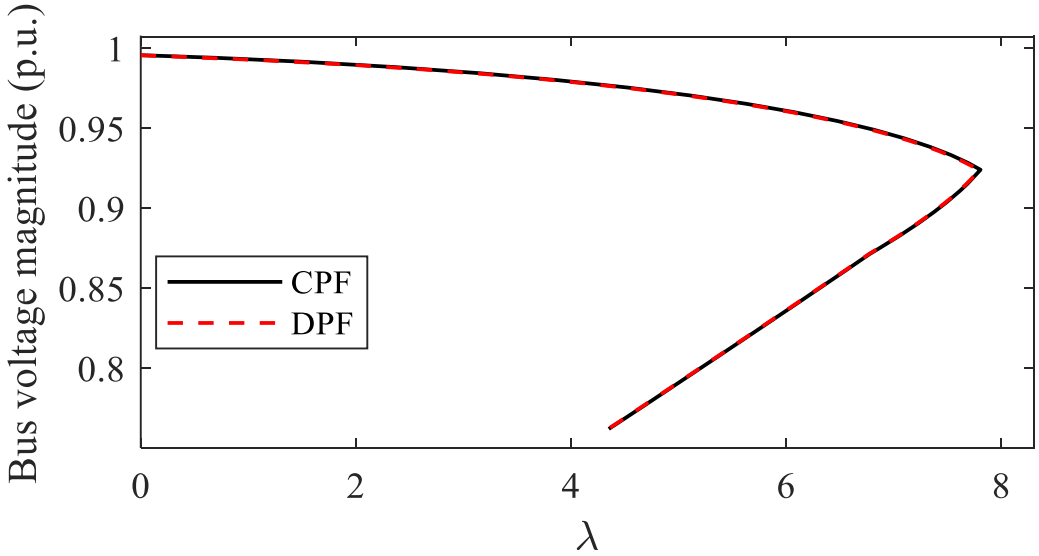



Figure 21. Solution curve of load bus 9 on 9-bus system




(a)




(b)

Figure 22. Solution curves under non-uniform load change direction, a) not consider reactive power limit, b) consider reactive power limit

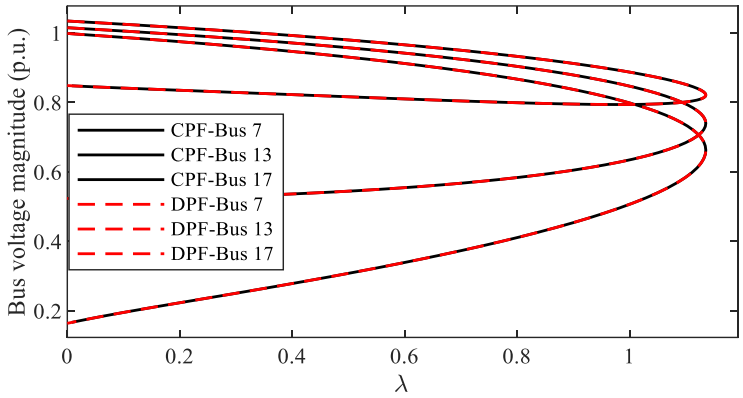



Figure 23. Solution curves on 39-bus system

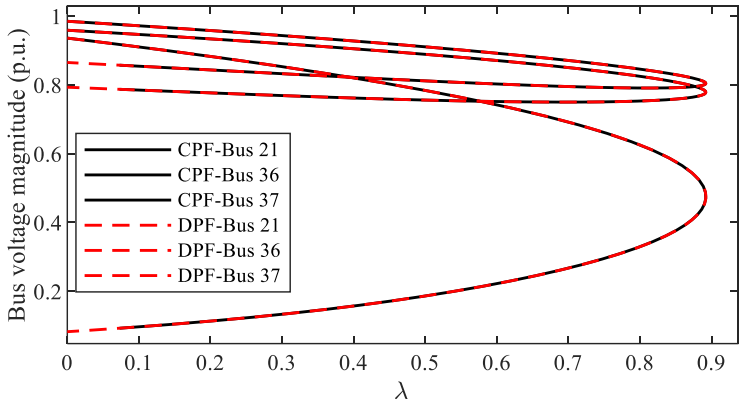



Figure 24. Solution curves on 57-bus system

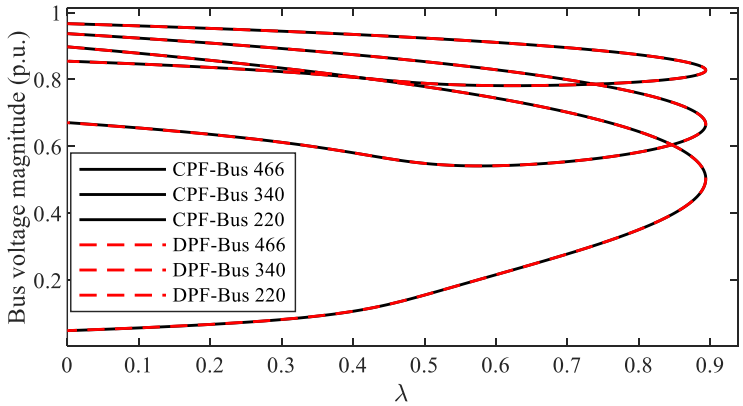



Figure 25. Solution curves on 2383-bus system

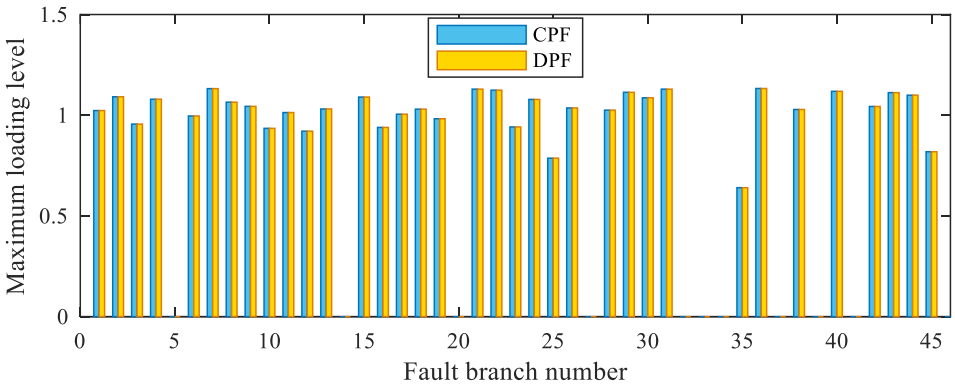



Figure 26. Maximum loading conditions by CPF and DPF

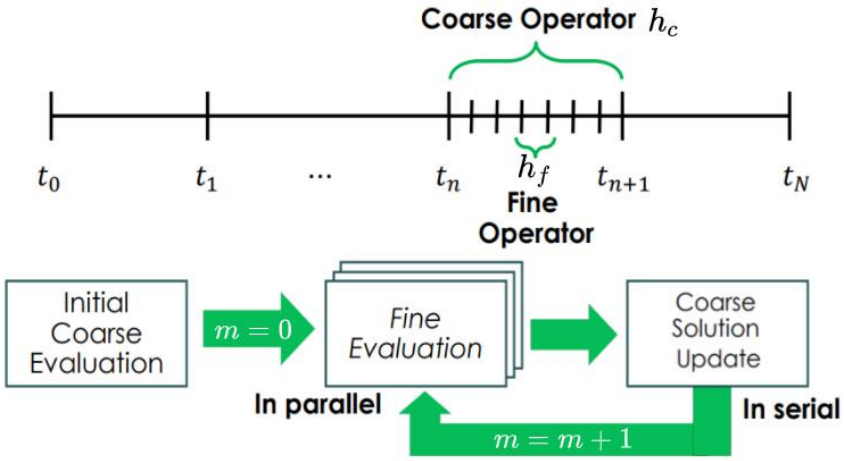



Figure 27. Illustration of the Parareal method

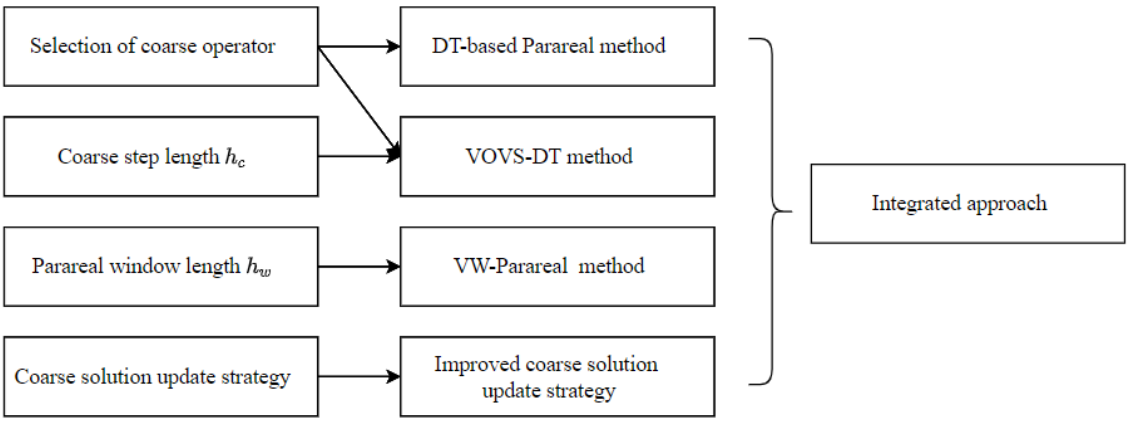



Figure 28. Structure of the proposed method

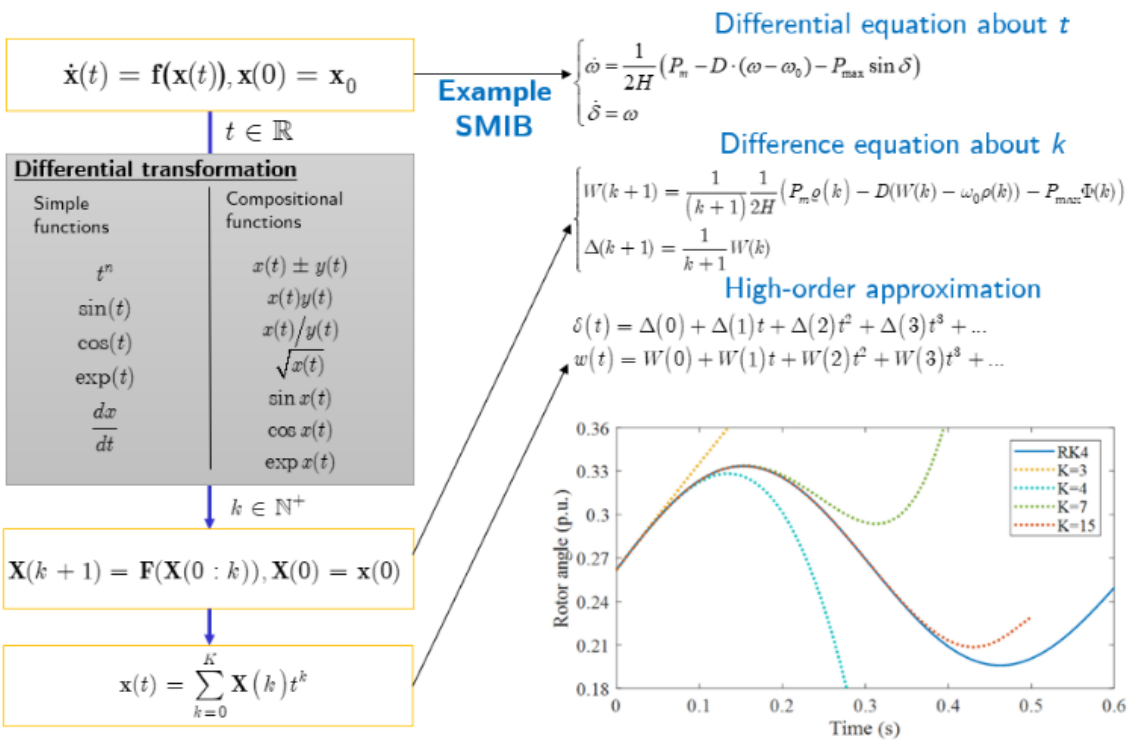



Figure 29. Illustration of the DT method

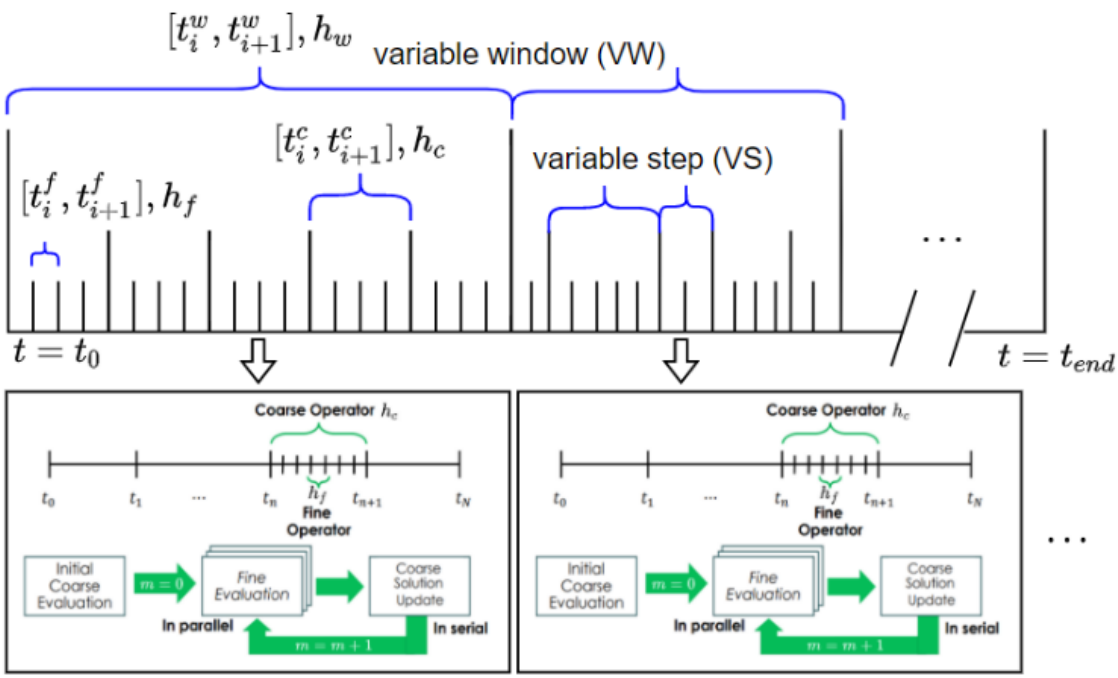



Figure 30. Illustration of the VW-Parareal method

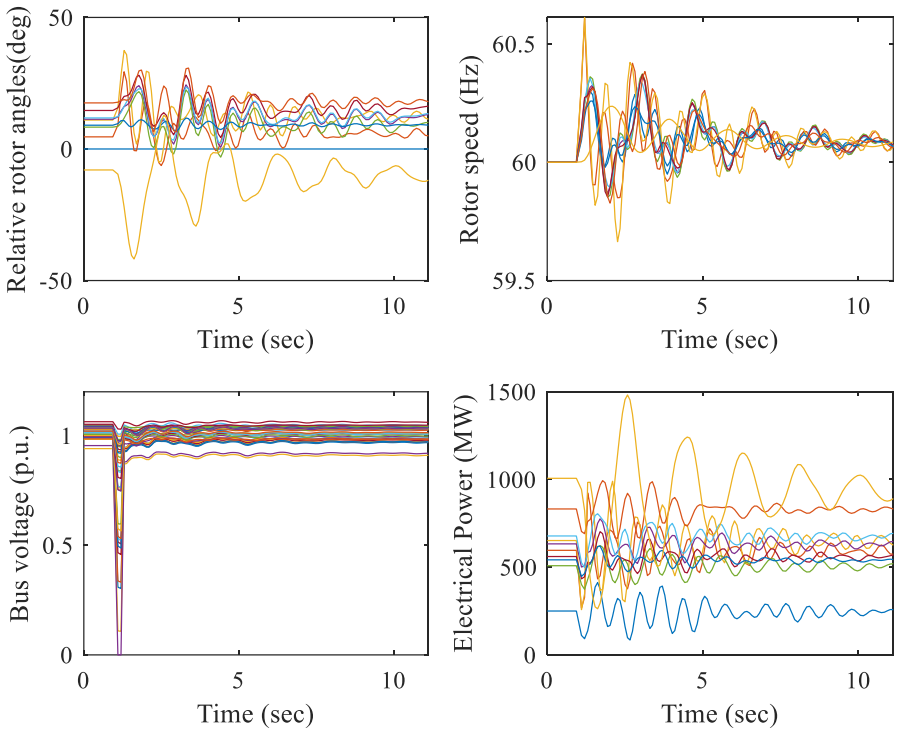



Figure 31. Trajectories of rotor angles, rotor speeds, bus voltages and electrical power in scenario 2

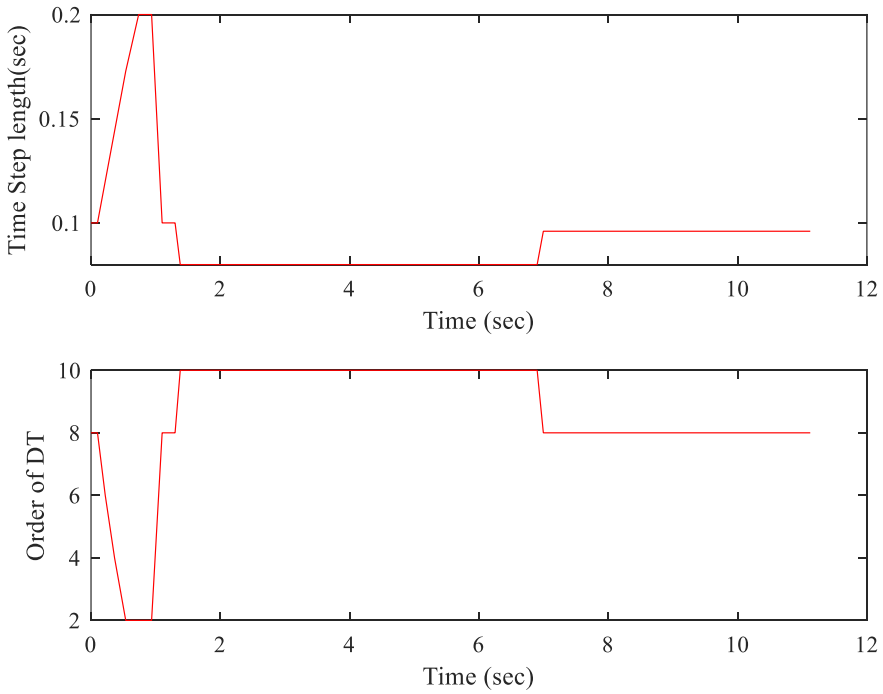



Figure 32. Evolution of time-step length and order of DT over the simulation

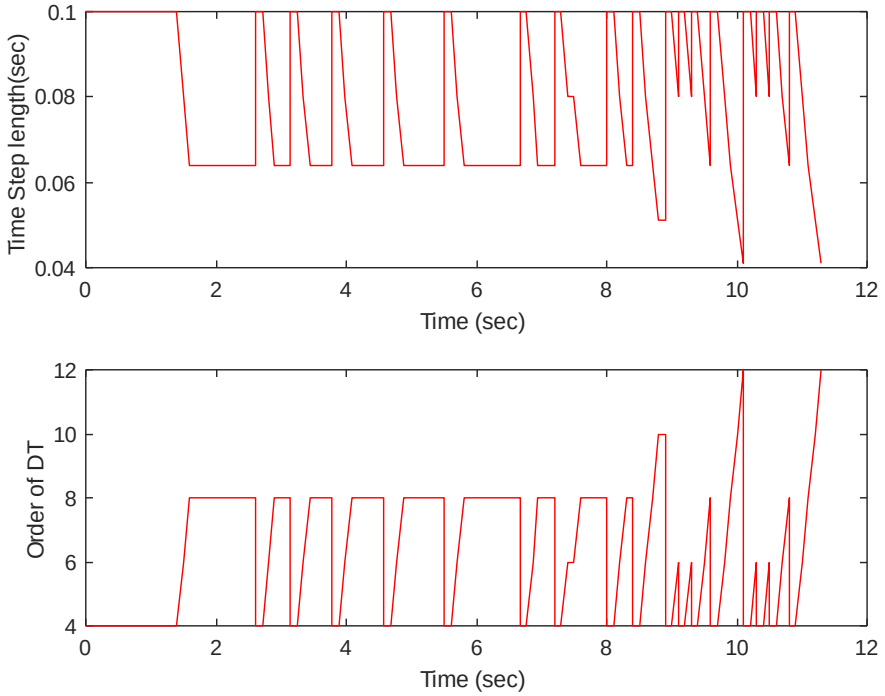



Figure 33. Evolution of time-step length and order of DT over the simulation

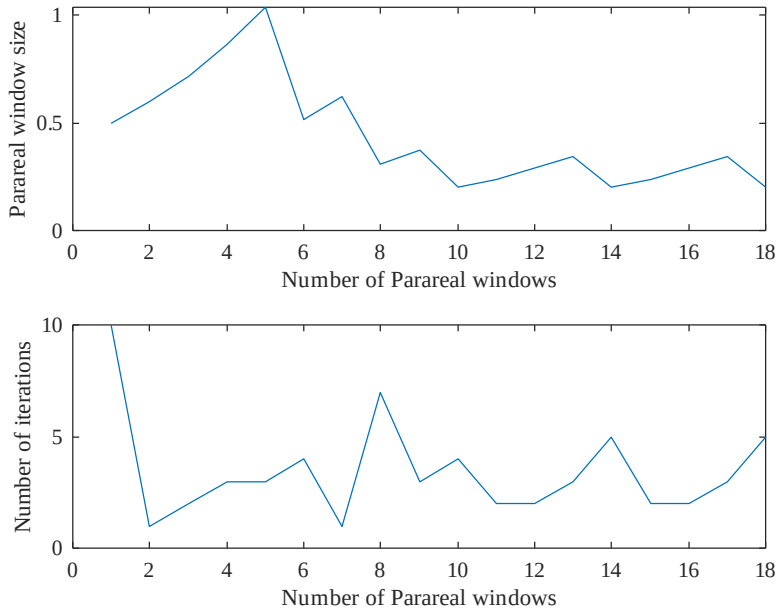



Figure 34. Evolution of time-step length and order of DT over the simulation

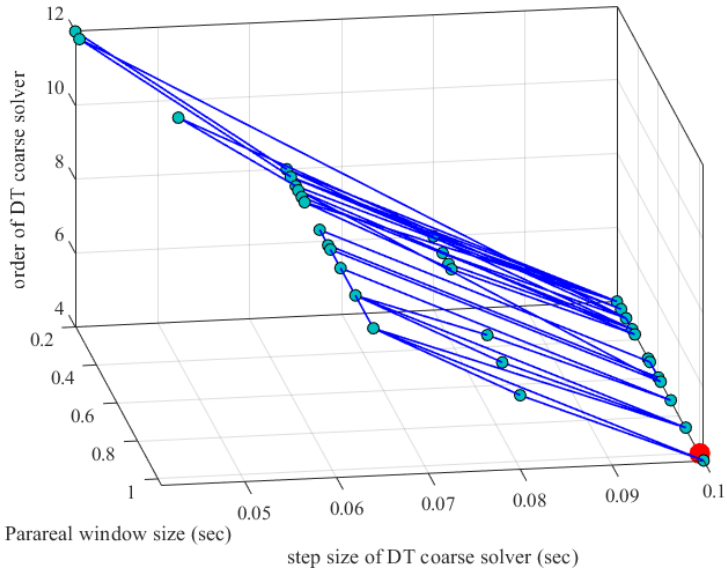



Figure 35. Evolution of time-step length and order of DT over the simulation

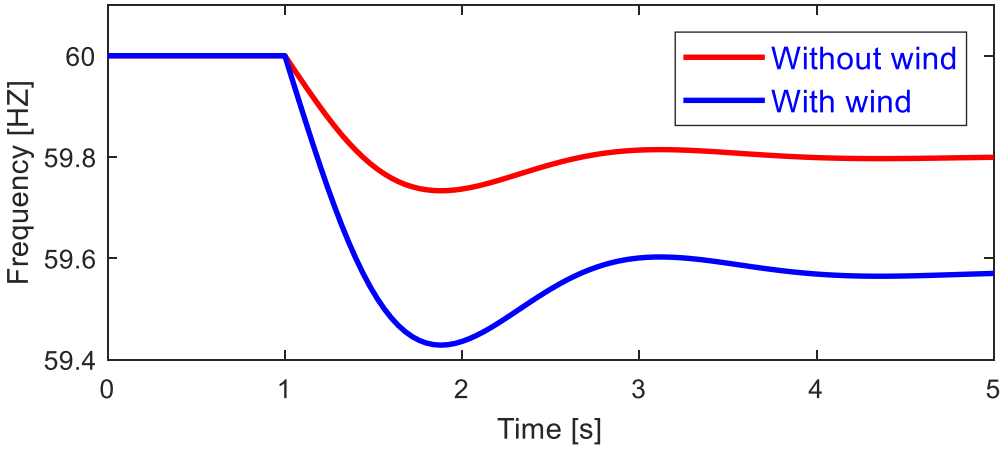



Figure 36. System frequency response with and without wind power

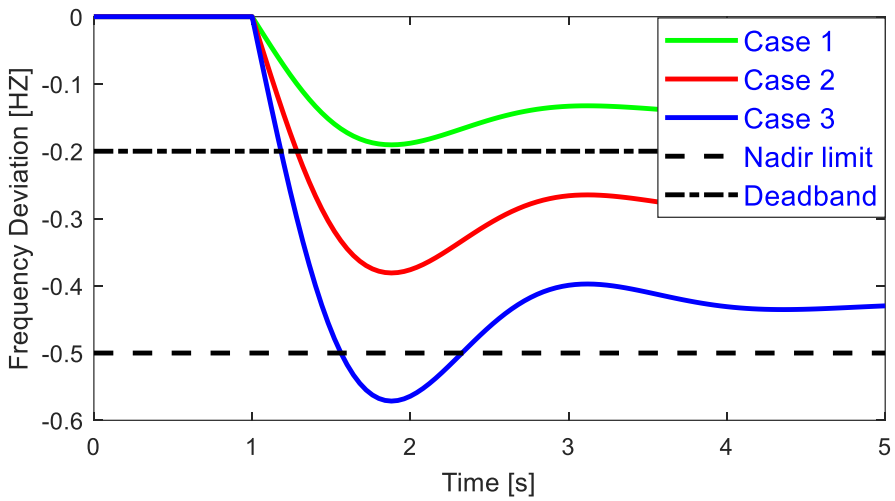



Figure 37. Frequency response under various disturbances

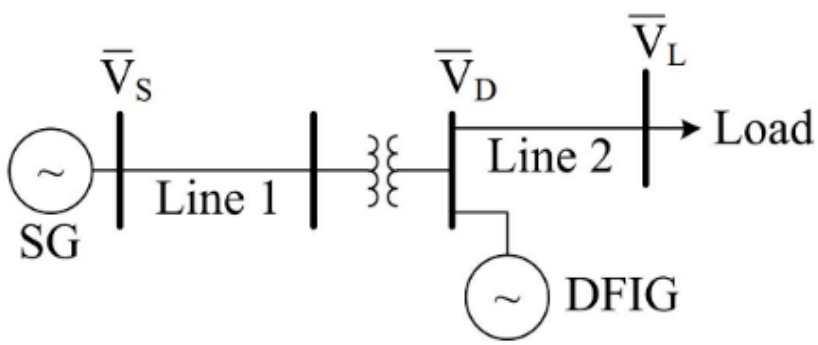
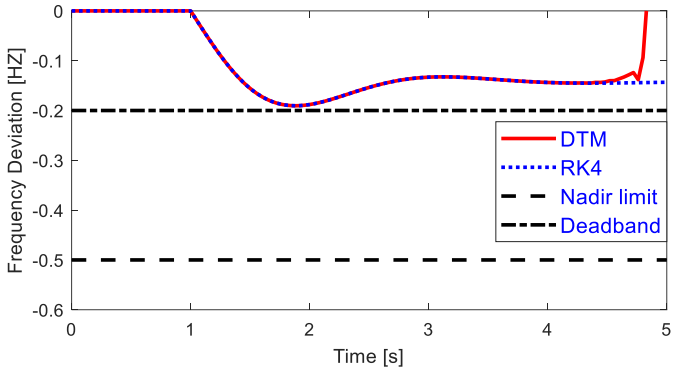
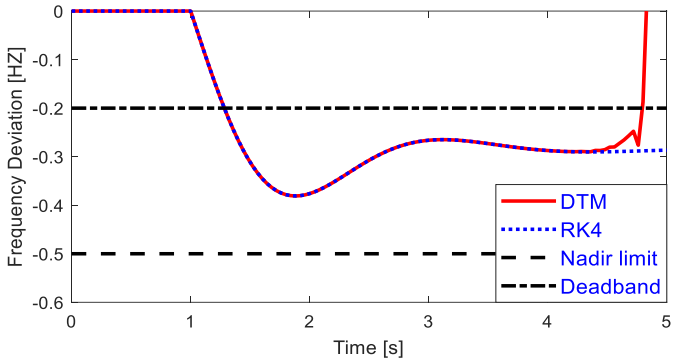


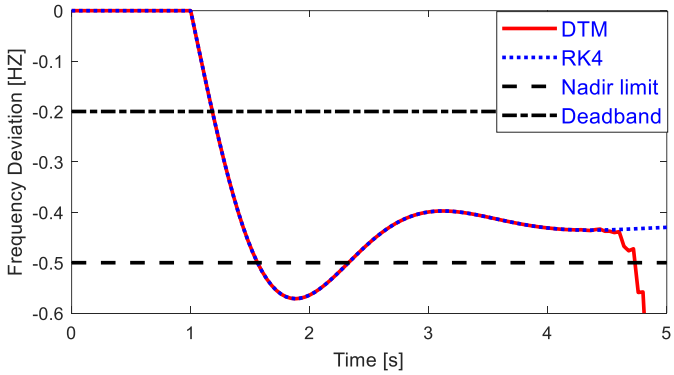

Figure 38. Diagram of four-bus test system




a) Scenario 1: small disturbance




b) Scenario 2: medium disturbance




c) Scenario 3: large disturbance

Figure 39. Comparisons of DTM and RK4

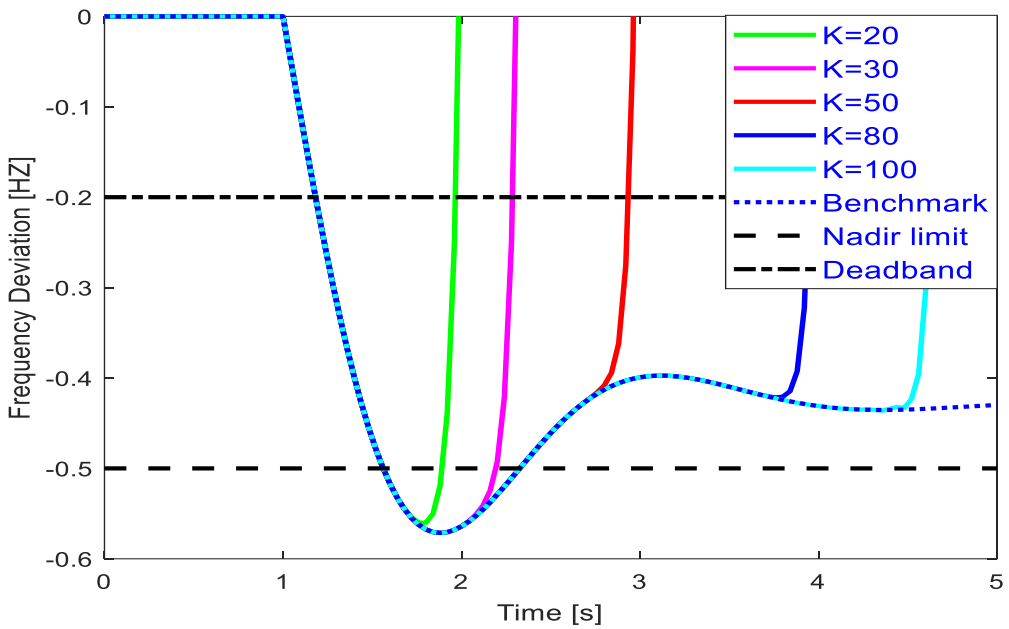



Figure 40. Frequency responses using DTM-based semi-analytical solutions with different orders

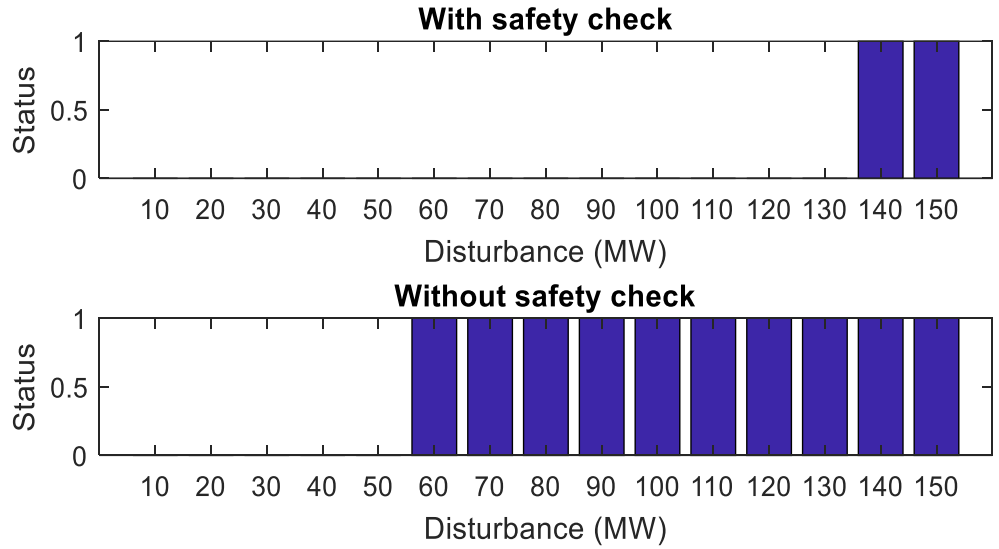



Figure 41. Switching of the frequency support mode with and without safety check( 1 – switched; 0 – unswitched )

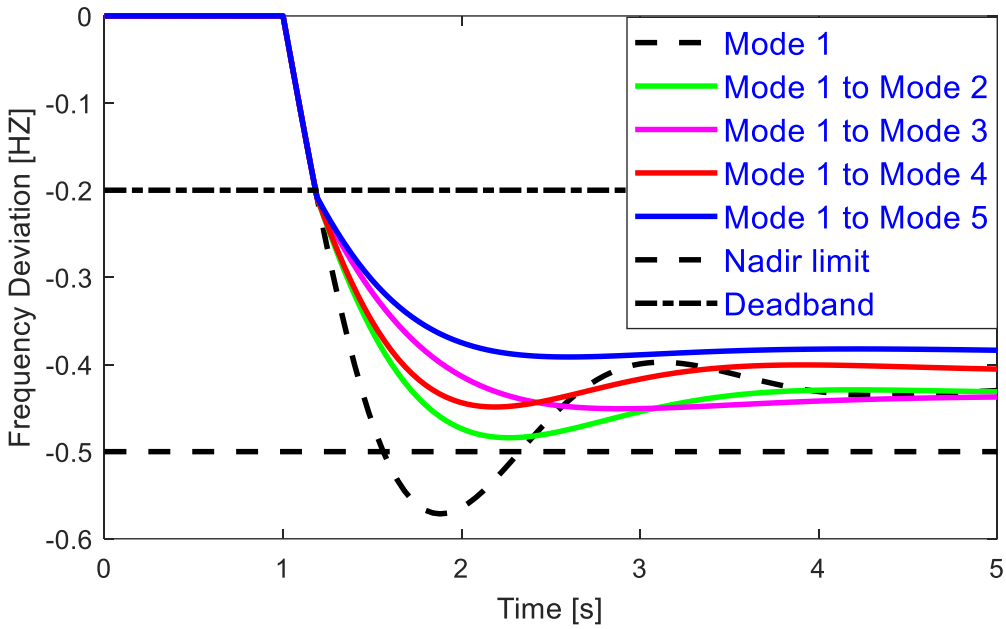
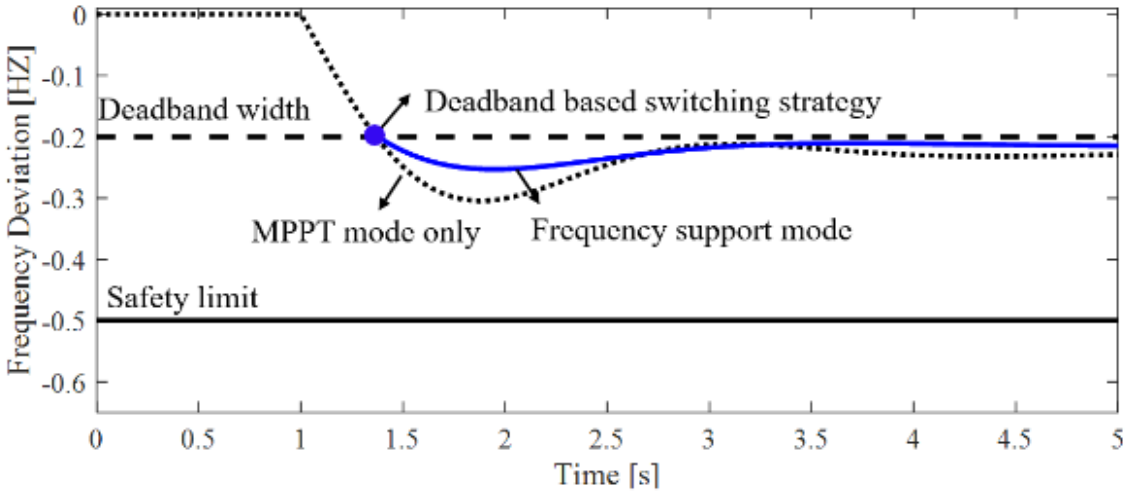
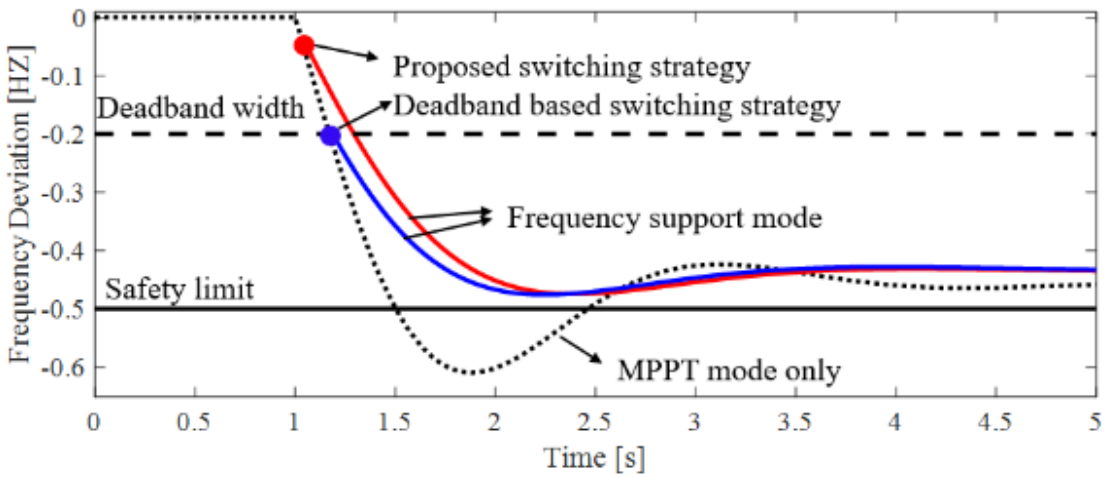



Figure 42. Frequency response when WTGs are switched to frequency support modes at

$$Dw = 0.2 \text{ HZ}$$

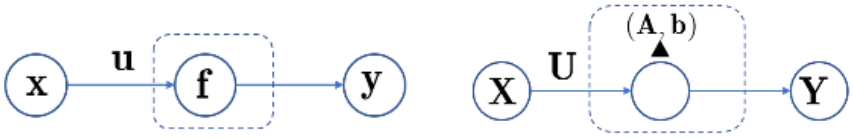



(a)



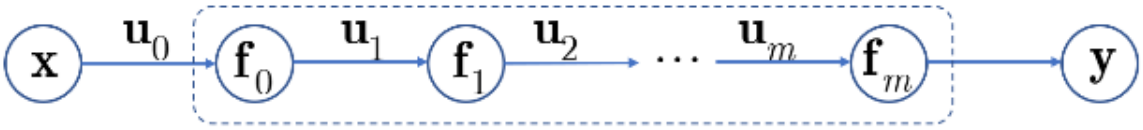

(b)

Figure 43. Switching strategy for the safe and unsafe frequency response

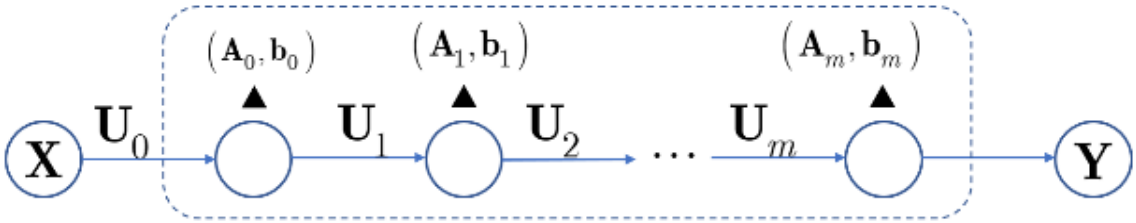



(a) (b)

Figure 44. a) DAG representation of one-input one-output one-layer composition function b) DAG representation of the transformed function equipped with a linear structure.

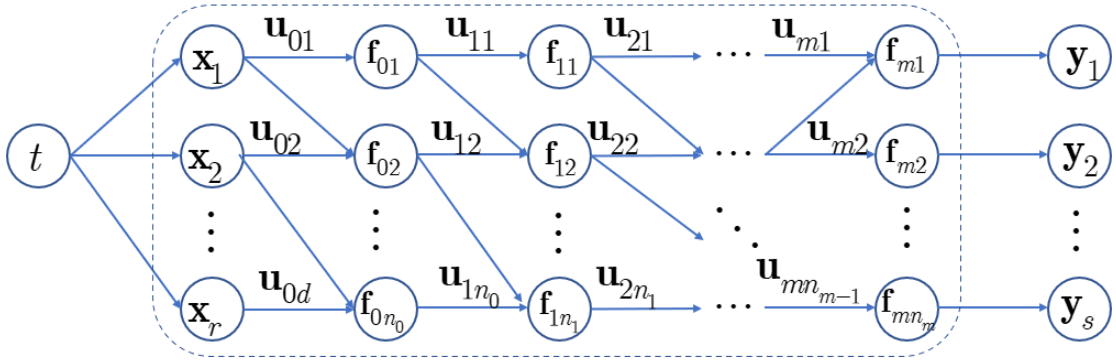



(a)

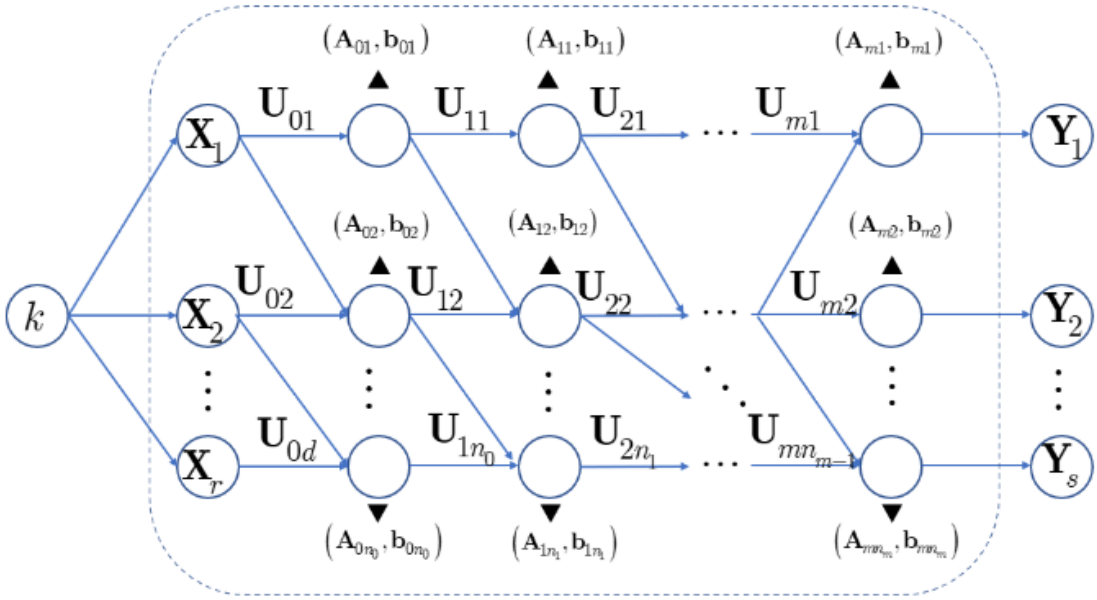



(b)

Figure 45. a) DAG representation of one-input one-output multi-layer composition function b) DAG representation of the transformed function equipped with a linear structure.

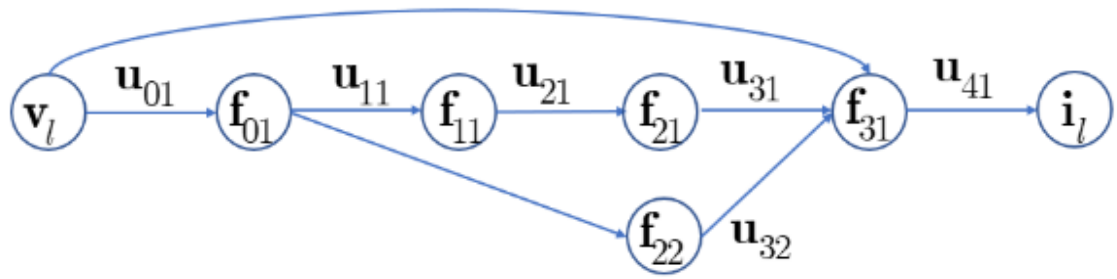



(a)

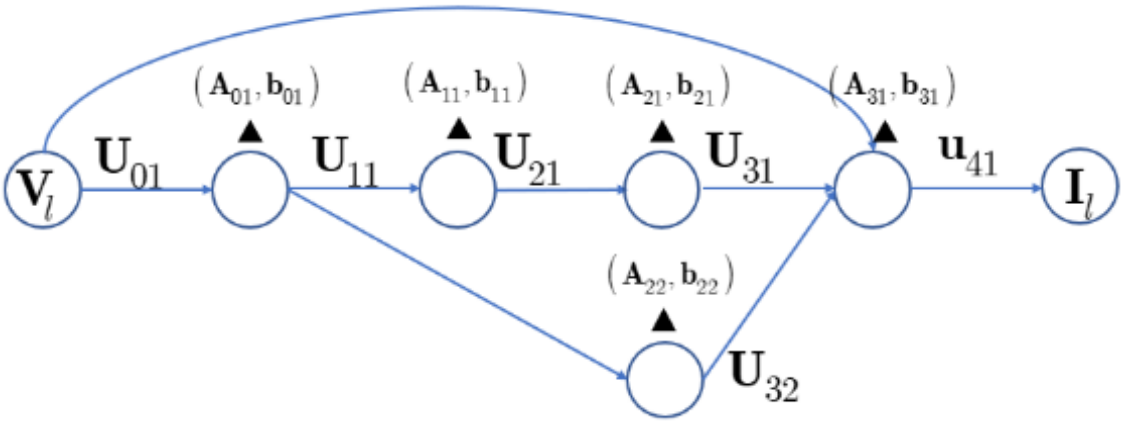



(b)

Figure 46. a) DAG representation of multi-input multi-output multi-layer composition function b) DAG representation of the transformed function equipped with a linear structure

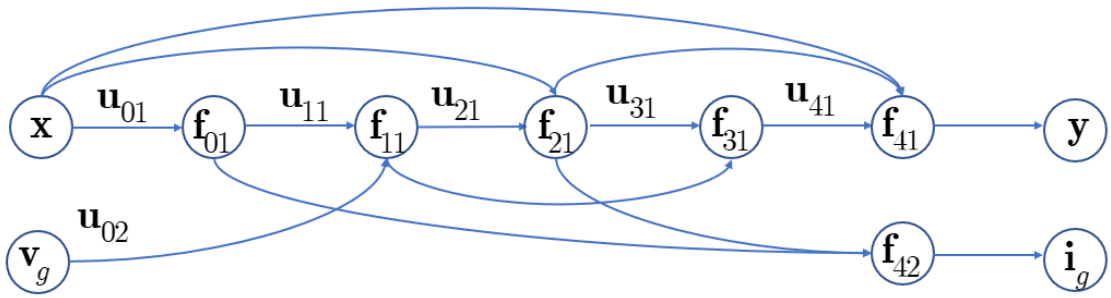



(a)

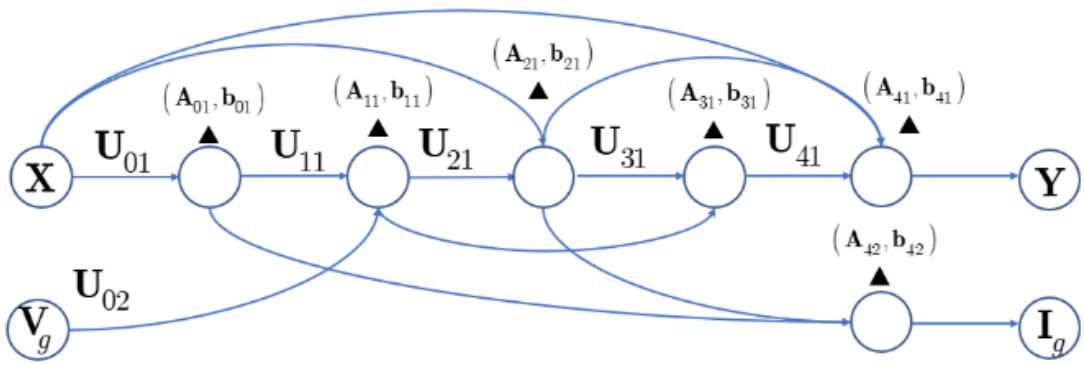



(b)

Figure 47. a) DAG representation of the compositional function in load model b) DAG representation of the transformed function.




(a)




(b)

Figure 48. a) DAG representation of the compositional function in generator model b) DAG representation of the transformed function.

# APPENDIX B – Tables

Table 1. Comparison of Numerical Stability under Three Scenarios

| Time step length (s) | ME | RK4 | RKG | DT |
|---|---|---|---|---|
| 0.050 | √ | √ | √ | √ |
| 0.067 | × | √ | √ | √ |
| 0.100 | × | × | × | √ |

Table 2. Comparison of Maximum Time Step Length to Maintain the Numerical Stability

(Unit: s)

| Test systems | ME | RK4 | RKG | DT |
|---|---|---|---|---|
| 39-bus system | 0.059 | 0.077 | 0.077 | 0.125 |
| 2383-bus system | 0.007 | 0.010 | 0.010 | 0.017 |

Table 3. Comparison of Time Step Length Under Different Error Tolerances (Unit: s)

| Test systems | Error (p.u.) | ME | RK4 | RKG | DT | Trapezoidal | Gear |
|---|---|---|---|---|---|---|---|
| 39-bus system | $10^{-2}$ | 0.012 | 0.045 | 0.045 | 0.090 | 0.020 | -- |
| | $10^{-3}$ | 0.003 | 0.026 | 0.026 | 0.070 | -- | 0.041 |
| | $10^{-4}$ | 0.001 | 0.011 | 0.011 | 0.050 | -- | -- |
| 2383-bus system | $10^{-2}$ | 0.003 | 0.010 | 0.010 | 0.017 | 0.004 | -- |
| | $10^{-3}$ | 0.001 | 0.007 | 0.007 | 0.016 | -- | 0.009 |
| | $10^{-4}$ | 0.0005 | 0.004 | 0.004 | 0.012 | -- | -- |

Table 4. Comparison of Computation Time Under Different Error Tolerances (Unit: s)

| Test systems | Error (p.u.) | ME | RK4 | RKG | DT | Trapezoidal | Gear |
|---|---|---|---|---|---|---|---|
| 39-bus system | $10^{-2}$ | 0.42 | 0.33 | 0.33 | 0.28 | 0.72 | -- |
| | $10^{-3}$ | 1.10 | 0.45 | 0.46 | 0.37 | -- | 0.42 |
| | $10^{-4}$ | 3.71 | 0.86 | 0.87 | 0.49 | -- | -- |
| 2383-bus system | $10^{-2}$ | 4.67 | 3.33 | 3.29 | 3.02 | 1320 | -- |
| | $10^{-3}$ | 14.10 | 4.04 | 4.06 | 3.67 | -- | 376 |
| | $10^{-4}$ | 42.67 | 8.34 | 8.29 | 5.10 | -- | -- |

Table 5. Comparison of Accuracy and Time Performance

| Scenarios | Methods | Error of state variables (p.u.) | Error of bus voltages (p.u.) | Computation time (s) |
|---|---|---|---|---|
| Stable | DT | $2.69\times10^{-6}$ | $3.33\times10^{-7}$ | 18.76 |
| | TRAR | $1.30\times10^{-4}$ | $1.10\times10^{-6}$ | 176.43 |
| | ME-NR | $2.63\times10^{-4}$ | $2.26\times10^{-6}$ | 191.40 |
| Unstable | DT | $1.89\times10^{-6}$ | $2.78\times10^{-7}$ | 18.85 |
| | TRAP | $1.41\times10^{-4}$ | $1.61\times10^{-6}$ | 182.76 |
| | ME-NR | $2.79\times10^{-4}$ | $2.93\times10^{-6}$ | 196.02 |

Table 6. Accuracy and Time Performance for Different Values of *K*

| *K* | Error of state variables (p.u.) | Error of bus voltages (p.u.) | Computation time (s) |
|---|---|---|---|
| 2 | $2.70\times10^{-2}$ | $4.91\times10^{-4}$ | 8.78 |
| 3 | $8.51\times10^{-4}$ | $2.10\times10^{-5}$ | 10.31 |
| 4 | $3.33\times10^{-5}$ | $1.82\times10^{-6}$ | 11.82 |
| 8 | $2.69\times10^{-6}$ | $3.33\times10^{-7}$ | 18.76 |
| 12 | $2.69\times10^{-6}$ | $3.33\times10^{-7}$ | 27.91 |

Table 7. Comparison of Number of LU Factorization

| Methods | $n_{LU}$ | $M$ | $N_{LU}=n_{LU}\times M$ |
|---|---|---|---|
| DT | 1 | 100 | 100 |
| TRAP-NR | 3.004 | 1000 | 3004 |
| ME-NR | 2.060 | 1000 | 2006 |

Table 8. Numbers of Times of Solving Linear Equations

| Test Systems | CPF | DPF |
|---|---|---|
| 39-bus system | 174 | 11 |
| 57-bus system | 108 | 10 |
| 2383-bus system | 424 | 18 |

Table 9. Comparison of Time Performance (Unit: second)

| Test Systems | CPF | DPF |
|---|---|---|
| 39-bus system | 0.26 | 0.03 |
| 57-bus system | 0.50 | 0.04 |
| 2383-bus system | 24.45 | 10.13 |

Table 10. Average Number of Iterations Using the Conventional Parareal Algorithm

| -- | | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|---|
| $h_c = 0.100$ | $h_w = 0.50$ | 1 | 1 | 5 |
| | $h_w = 0.75$ | 7 | 7 | 7 |
| | $h_w = 1.00$ | Divergent | Divergent | Divergent |
| | $h_w = 1.25$ | Divergent | Divergent | Divergent |
| $h_c = 0.075$ | $h_w = 0.50$ | 1 | 1 | 1 |
| | $h_w = 0.75$ | 1 | 1 | Divergent |
| | $h_w = 1.00$ | 1 | 1 | Divergent |
| | $h_w = 1.25$ | 1 | Divergent | Divergent |
| $h_c = 0.050$ | $h_w = 0.50$ | 1 | 1 | 1 |
| | $h_w = 0.75$ | 1 | 1 | 1 |
| | $h_w = 1.00$ | 1 | 1 | 1 |
| | $h_w = 1.25$ | 1 | 1 | 1 |

Table 11. Average Number of Iterations Using the DT-based Parareal Method

| -- | | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|---|
| $h_c = 0.100$ | $h_w = 0.50$ | 1 | 1 | 1 |
| | $h_w = 0.75$ | 1 | 1 | 1 |
| | $h_w = 1.00$ | 1 | 1 | 1 |
| | $h_w = 1.25$ | 1 | 1 | 1 |
| $h_c = 0.075$ | $h_w = 0.50$ | 1 | 1 | 1 |
| | $h_w = 0.75$ | 1 | 1 | 1 |
| | $h_w = 1.00$ | 1 | 1 | 1 |
| | $h_w = 1.25$ | 1 | 1 | 1 |
| $h_c = 0.050$ | $h_w = 0.50$ | 1 | 1 | 1 |
| | $h_w = 0.75$ | 1 | 1 | 1 |
| | $h_w = 1.00$ | 1 | 1 | 1 |
| | $h_w = 1.25$ | 1 | 1 | 1 |

Table 12. Average Number of Iterations using the VW-Parareal Method

| -- | | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|---|
| $h_c = 0.100$ | $h_w = 0.50$ | 1.91 | 1.42 | 1.86 |
| | $h_w = 0.75$ | 1.83 | 2.00 | 1.95 |
| | $h_w = 1.00$ | 3.00 | 2.97 | 2.72 |
| | $h_w = 1.25$ | 3.10 | 3.10 | 3.06 |
| $h_c = 0.075$ | $h_w = 0.50$ | 1.00 | 1.00 | 2.25 |
| | $h_w = 0.75$ | 1.00 | 1.00 | 2.16 |
| | $h_w = 1.00$ | 1.00 | 1.00 | 2.32 |
| | $h_w = 1.25$ | 3.42 | 2.41 | 2.48 |
| $h_c = 0.050$ | $h_w = 0.50$ | 1.00 | 1.00 | 1.00 |
| | $h_w = 0.75$ | 1.00 | 1.00 | 1.00 |
| | $h_w = 1.00$ | 1.00 | 1.00 | 1.00 |
| | $h_w = 1.25$ | 1.00 | 1.00 | 1.00 |

Table 13. Average Window Length Using the VW-Parareal Method

| -- | | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|---|
| $h_c = 0.100$ | $h_w = 0.50$ | 0.49 | 0.85 | 0.48 |
| | $h_w = 0.75$ | 0.61 | 0.39 | 0.49 |
| | $h_w = 1.00$ | 0.32 | 0.32 | 0.32 |
| | $h_w = 1.25$ | 0.33 | 0.33 | 0.33 |
| $h_c = 0.075$ | $h_w = 0.50$ | 1.31 | 1.31 | 0.51 |
| | $h_w = 0.75$ | 1.53 | 1.53 | 0.34 |
| | $h_w = 1.00$ | 1.73 | 1.73 | 0.34 |
| | $h_w = 1.25$ | 0.33 | 0.32 | 0.32 |
| $h_c = 0.050$ | $h_w = 0.50$ | 1.31 | 1.22 | 1.22 |
| | $h_w = 0.75$ | 1.53 | 1.53 | 1.53 |
| | $h_w = 1.00$ | 1.73 | 1.73 | 1.73 |
| | $h_w = 1.25$ | 1.86 | 1.86 | 1.86 |

Table 14. Average Number of Iterations Using the VW-Parareal Method with Improved Coarse Update Strategy

| -- | | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|---|
| $h_c = 0.100$ | $h_w = 0.50$ | 1.00 | 1.00 | 1.33 |
| | $h_w = 0.75$ | 1.00 | 1.14 | 1.86 |
| | $h_w = 1.00$ | 1.33 | 1.67 | 2.17 |
| | $h_w = 1.25$ | 1.60 | 1.80 | 2.40 |
| $h_c = 0.075$ | $h_w = 0.50$ | 1.00 | 1.00 | 1.00 |
| | $h_w = 0.75$ | 1.00 | 1.00 | 1.00 |
| | $h_w = 1.00$ | 1.00 | 1.00 | 1.33 |
| | $h_w = 1.25$ | 1.00 | 1.00 | 1.80 |
| $h_c = 0.050$ | $h_w = 0.50$ | 1.00 | 1.00 | 1.00 |
| | $h_w = 0.75$ | 1.00 | 1.00 | 1.00 |
| | $h_w = 1.00$ | 1.00 | 1.00 | 1.00 |
| | $h_w = 1.25$ | 1.00 | 1.00 | 1.00 |

Table 15. Average Window Length Using the VW-Parareal Method with Improved Coarse Update Strategy

| -- | | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|---|
| $h_c = 0.100$ | $h_w = 0.50$ | 1.31 | 1.31 | 1.31 |
| | $h_w = 0.75$ | 1.53 | 1.53 | 1.53 |
| | $h_w = 1.00$ | 1.73 | 1.73 | 1.73 |
| | $h_w = 1.25$ | 1.86 | 1.86 | 1.86 |
| $h_c = 0.075$ | $h_w = 0.50$ | 1.31 | 1.31 | 1.31 |
| | $h_w = 0.75$ | 1.53 | 1.53 | 1.53 |
| | $h_w = 1.00$ | 1.73 | 1.73 | 1.73 |
| | $h_w = 1.25$ | 1.86 | 1.86 | 1.86 |
| $h_c = 0.050$ | $h_w = 0.50$ | 1.31 | 1.22 | 1.22 |
| | $h_w = 0.75$ | 1.53 | 1.53 | 1.53 |
| | $h_w = 1.00$ | 1.73 | 1.73 | 1.73 |
| | $h_w = 1.25$ | 1.86 | 1.86 | 1.86 |

Table 16. Average Time Step Length of 39-bus System (Seconds)

| -- | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|
| DT | 0.100 | 0.100 | 0.100 |
| VO-DT | 0.100 | 0.100 | 0.100 |
| VS-DT | 0.103 | 0.084 | 0.069 |
| VOVS-DT | 0.103 | 0.090 | 0.088 |

Table 17. Average Order of 39-bus System

| -- | Scenario 1 | Scenario 3 | Scenario 3 |
|---|---|---|---|
| DT | 8.00 | 8.00 | 8.00 |
| VO-DT | 7.58 | 9.33 | 9.31 |
| VS-DT | 8.00 | 8.00 | 8.00 |
| VOVS-DT | 7.78 | 8.94 | 9.15 |

Table 18. Average Order of 2383-bus System

| -- | Method 1 | Method 2 | Method 3 |
|---|---|---|---|
| Average number of iterations | 9.10 | 3.24 | 3.44 |
| Average window length | 1.00 | 0.46 | 0.44 |
| Average order of DT | -- | 4 | 6.50 |
| Average time step length of DT | -- | 0.1 | 0.08 |
| CPU time | 3021 | 2346 | 2198 |

Table 19. Operation Status of Frequency Support Mode

| Situations | Disturbance | Actual status of frequency support | Ideal status of frequency support |
|---|---|---|---|
| Case 1 | Small | Deactivated | Deactivated |
| Case 2 | Medium | **Activated** | **Deactivated** |
| Case 3 | Large | Activated | Activated |

Table 20. Settings of the Three Scenarios

| Situations | Disturbance | $\mathrm{D}p_d$ (MW) |
|---|---|---|
| Scenario 1 | Small | 50 |
| Scenario 2 | Medium | 100 |
| Scenario 3 | Large | 150 |

# VITA

Yang Liu received the B.S. degree in energy and power engineering from Xi'an Jiaotong University, China, in 2013, and the M.S. degree in power engineering from Tsinghua University, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, The University of Tennessee, Knoxville, USA. His research interests include power system simulation, nonlinear dynamics, stability, and control.