

A mixed-signal computer architecture and its application to power system problems

THÈSE N° 6613 (2015)

PRÉSENTÉE LE 4 SEPTEMBRE 2015

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

GROUPE KAYAL

PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Theodoros KYRIAKIDIS

acceptée sur proposition du jury:

Dr J.-M. Sallese, président du jury
Prof. M. Kayal, Dr S.-R. Cherkaoui, directeurs de thèse
Prof. T. Van Cutsem, rapporteur
Prof. A. Hatzopoulos, rapporteur
Prof. M. Paolone, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

Ἐν ἀρχῇ ἦν ὁ Λόγος.
— Ἰω. 1

To my family.

Acknowledgements

First of all I would like to thank my supervisors Maher Kayal and Rachid Cherkaoui. They gave me the opportunity to conduct my research “the old way”, in the sense of providing me with suggestions and tips along the way rather constraining me to a fixed target. It is this kind of freedom that allows the investigation of “crazy ideas” and it is due to them that this work has been enabled.

I would like to equally thank the professor Thierry Van Cutsem of the University of Liège and his group for their exemplary hospitality during my spell with them in Liège. Particular mention goes to dear friend and colleague Petros Aristidou for all our fruitful discussions and his invaluable help.

A big part of what you read in this manuscript is done in collaboration with colleagues of my lab and elsewhere. Sometimes collaboration was so tight that it is really difficult to say who has really contributed what. I would like to specially mention Guillaume Lanz, Denis Sallin and Georgios Lilis. Each one with his deep dedication to his field, and each one always willing to share and help.

Apart from the above, there is a number of people that have contributed their bit in the scientific part of this work. It being a discussion over a coffee, a review of a paper, a coding idea, or some useful small-talk over a pint of beer, to you all people, I am grateful.

I would also like to thank all the people who have made my days here in Switzerland happy and cheerful: my office-, flat-, and team-mates, and all my amazing Greek and Swiss friends.

I would finally like to thank my family, for their constant support and for the happiness they bring to my life.

Lausanne, 04 Juin 2015

Θ. K.

Abstract

Radical changes are taking place in the landscape of modern power systems. This massive shift in the way the system is designed and operated has been termed the advent of the “smart grid”. One of its implications is a strong market pull for faster power system analysis computing. This work concerns in particular transient simulation, which is one of the most demanding power system analyses. This refers to the imitation of the operation of the real-world system over time, for time scales that cover the majority of slow electromechanical transient phenomena. The general mathematical formulation of the simulation problem includes a set of non-linear differential algebraic equations (DAEs).

In the algebraic part of this set, heavy linear algebra computations are included, which are related to the admittance matrix of the topology. These computations are a critical factor to the overall performance of a transient simulator.

This work proposes the use of analog electronic computing as a means of exceeding the performance barriers of conventional digital computers for the linear algebra operations. Analog computing is integrated in the frame of a power system transient simulator yielding significant computational performance benefits to the latter. Two hybrid, analog and digital computers are presented.

The first prototype has been implemented using reconfigurable hardware. In its core, analog computing is used for linear algebra operations, while pipelined digital resources on a field programmable gate array (FPGA) handle all remaining computations. The properties of the analog hardware are thoroughly examined, with special attention to accuracy and timing. The application of the platform to the transient analysis of power system dynamics showed a speedup of two orders of magnitude against conventional software solutions.

The second prototype is proposed as a future conceptual architecture that would overcome the limitations of the already implemented hardware, while retaining its virtues. The design space of this future architecture has been thoroughly explored, with the help of a software emulator. For one possible suggested implementation, speedups of two orders of magnitude against software solvers have been observed for the linear algebra operations.

Key words: [**Computing**]: analog computing, pipeline processing, field programmable gate arrays, reconfigurable computing, computer accelerator architectures, high performance computing; [**Power systems**]: power system simulations, power system analysis computing, power system dynamics; [**Mathematics**]: linear algebra - *linear systems*, numerical analysis - *numerical simulation, numerical linear algebra*;

Résumé

Des changements radicaux prennent place dans les systèmes électriques modernes. Ces changements massifs dans la façon dont le système est conçu et exploité correspondent à l'avènement du "smart grid". L'une des conséquences est le besoin d'outils d'analyse du système plus rapides.

Ce travail concerne en particulier une analyse très exigeante, la simulation transitoire. Ceci se rapporte à l'imitation de l'opération du système réel, pour des échelles de temps qui couvrent la majorité des phénomènes transitoires électromécaniques lents. La formulation mathématique générale du problème comprend un ensemble d'équations algébriques différentielles non-linéaires (DAE).

Dans la partie algébrique de cet ensemble, de lourds calculs d'algèbre linéaire prennent place, qui sont liés à la matrice d'admittance de la topologie. Ces calculs sont un facteur critique de la performance globale d'un simulateur transitoire.

Ce travail propose l'utilisation des calculs analogiques comme un moyen de dépasser les barrières de performance des ordinateurs numériques classiques pour les opérations d'algèbre linéaire. Le calcul analogique est intégré dans le cadre d'un simulateur transitoire et apporte d'importants avantages en termes de performances. Deux ordinateurs hybrides, analogique et numérique, sont présentés.

Le premier prototype a été mis en œuvre en utilisant ressources reconfigurables. Dans son cœur, le calcul analogique est utilisé pour des opérations d'algèbre linéaire, tandis que les ressources numériques sur un Field Programmable Gate Array (FPGA) gèrent tous les calculs restants. Les propriétés du matériel analogique sont examinées. Une attention particulière est portée à la précision et au timing. L'application de la plate-forme à l'analyse transitoire de la dynamique du réseau électrique a montré une accélération de deux ordres de grandeur par rapport à des solutions logicielles classiques.

Le deuxième prototype est proposé comme une future architecture conceptuelle qui permettrait de surmonter les limitations du matériel déjà mis en œuvre, tout en conservant ses vertus. L'espace de conception de cette future architecture a été explorée à fond, avec l'aide d'un émulateur en logiciel. Pour une mise en œuvre suggérée, accélérations de deux ordres de grandeur contre logiciels classiques ont été observées pour les opérations d'algèbre linéaire.

Mots clefs : **[Informatique]** : calcul analogique, traitement en pipeline, circuit logique programmable, computing reconfigurable, architectures d'ordinateur de l'accélérateur, calcul haute performance ; **[Réseau électrique]** : simulations du réseau électrique, informatique

Acknowledgements

de l'analyse du réseau électrique, dynamique du réseau électrique ; **[Mathématiques]** : algèbre linéaire - *systèmes linéaires*, analyse numérique - *simulation numérique*, algèbre linéaire numérique;

Περίληψη

Ριζικές αλλαγές λαμβάνουν χώρα στον τομέα των σύγχρονων συστημάτων ηλεκτρικής ενέργειας (ΣΗΕ). Αυτή η μετάβαση στον τρόπο με τον οποίο το σύστημα σχεδιάζεται και λειτουργείται αποκαλείται η έλευση του 'έξυπνου δικτύου'. Ένα από τα επακόλουθα της είναι η ανάγκη για ταχύτερους υπολογιστές ανάλυσης ΣΗΕ.

Αυτή η εργασία επικεντρώνεται σε μια ιδιαίτερη απαιτητική ανάλυση, την προσομοίωση ΣΗΕ. Αυτό αναφέρεται στην πιστή αναπαράσταση της λειτουργίας του συστήματος για σταθερές χρόνου που καλύπτουν την πλειονότητα των ηλεκτρομηχανικών μεταβατικών φαινομένων στο σύστημα. Η γενική μαθηματική διατύπωση του προβλήματος της προσομοίωσης περιλαμβάνει ένα σύνολο μη γραμμικών διαφορικών-αλγεβρικών εξισώσεων (ΔΑΕ).

Στο αλγεβρικό μέρος αυτού του συνόλου περιλαμβάνονται πολλοί και απαιτητικοί υπολογισμοί γραμμικής άλγεβρας. Αυτοί σχετίζονται με τον πίνακα αγωγιμότητας της τοπολογίας και είναι κρίσιμοι για την συνολική απόδοση ενός προσομοιωτή μεταβατικών φαινομένων ΣΗΕ.

Σε αυτή την εργασία προτείνεται η χρήση αναλογικών αρχιτεκτονικών ως μέσο υπέρβασης των περιορισμών των συμβατικών ψηφιακών υπολογιστών όσον αφορά ιδιαίτερα τους υπολογισμούς γραμμικής άλγεβρας. Οι αναλογικοί υπολογισμοί ενσωματώνονται στην ροή του προσομοιωτή μεταβατικών φαινομένων αποφέροντας σημαντικά ωφέλη στην απόδοση του. Δύο υβριδικοί υπολογιστές παρουσιάζονται.

Ο πρώτος αφορά ένα πρωτότυπο το οποίο έχει κατασκευαστεί σε επαναδιαμορφώσιμο υλικό. Στον πυρήνα του αναλογικοί υπολογισμοί χρησιμοποιούνται για την γραμμική άλγεβρα, ενώ σωληνωμένοι επεξεργαστές έχουν προγραμματιστεί σε μια συστοιχία επιτόπια προγραμματιζομένων πυλών (ΦΠΓΑ) για τους υπόλοιπους υπολογισμούς. Οι ιδιότητες του αναλογικού τμήματος εξετάζονται διεξοδικά, με ιδιαίτερη έμφαση στην ακρίβεια και τον χρονισμό. Η εφαρμογή της πλατφορμας στην προσομοίωση ΣΗΕ έδειξε μια επιτάχυνση δύο τάξεων μεγέθους σε σχέση με συμβατικό ψηφιακό λογισμικό.

Ένα δεύτερο πρωτότυπο προτείνεται ως μελλοντική αρχιτεκτονική η οποία θα μπορέσει να διατηρήσει τις αρετές του υπάρχοντος πρωτοτύπου, αλλά και ταυτόχρονα να ξεπεράσει τους περιορισμούς του. Ένας εξομοιωτής υλοποιήθηκε σε λογισμικό με σκοπό την ανάλυση και την βελτιστοποίηση του σχεδιασμού. Για την τελικά προτεινόμενη διαμόρφωση, επιτάχυνση δύο τάξεων μεγέθους επιτεύχθηκε για τους υπολογισμούς γραμμικής άλγεβρας σε σχέση με συμβατικό ψηφιακό λογισμικό.

Λέξεις κλειδιά: Πληροφορική: αναλογική πληροφορική, σωληνωμένη επεξεργασία, συστοιχία

Acknowledgements

επιτόπια προγραμματιζόμενων πυλών, επαναδιαμορφώσιμοι υπολογιστές, αρχιτεκτονικές επιταχυντών υλικού υπολογιστές υψηλών επιδόσεων. Συστήματα ηλεκτρικής ενέργειας (ΣΗΕ): προσομοίωση ΣΗΕ, υπολογιστές ανάλυσης ΣΗΕ, δυναμική ΣΗΕ. Μαθηματικά: γραμμική άλγεβρα - γραμμικά συστήματα, αριθμητική ανάλυση - αριθμητική ολοκλήρωση, αριθμητική γραμμική άλγεβρα.

Contents

Acknowledgements	i
Abstract (en/fr/el)	iii
List of figures	xiii
List of tables	xvii
Introduction	1
1 Modern power system landscape	5
1.1 Current architecture	6
1.1.1 The need for change	13
1.2 Future architecture: the Smart Grid	14
1.2.1 Distributed energy resources	16
1.2.2 Aggregations	17
1.3 The effect of the smart grid on analysis tools	18
2 Power system simulation, linear algebra and computing platforms	21
2.1 Simulation	22
2.1.1 Mathematical formulation of simulation problems	23
2.2 Linear algebra in power systems	24
2.2.1 Algorithms and implementations	24
2.2.2 Coherence between the algorithm and the platform	26
2.3 Dedicated platforms	28
2.3.1 SIMD	28
2.3.2 Multi-core	28
2.3.3 Heterogeneous computing	30
2.3.4 GPU	30
2.3.5 FPGA	31
2.3.6 DSP	32
2.3.7 ASIC and VLSI	32
2.3.8 Conventional computing and its limitations	33
2.3.9 Unconventional computing	33
2.4 Analog electronic computers	34
	ix

Contents

2.4.1	In linear algebra	35
2.4.2	In power systems computing	36
2.4.3	Evaluation criteria	37
2.5	Outlook	38
3	Realized dedicated mixed signal solver	41
3.1	Hardware	42
3.1.1	Analog part	45
3.1.2	Digital part	55
3.1.3	Timing	61
3.2	Software	62
3.2.1	Backend	62
3.2.2	Frontend	65
3.3	Inaccuracy	66
3.3.1	Analog inaccuracy	66
3.3.2	Digital inaccuracy	80
3.3.3	Effect on the mathematical operation	82
3.3.4	Calibration	84
3.4	Results	86
3.4.1	Linear system solving	86
3.4.2	Sample radial and meshed topologies	91
3.4.3	Transient simulation	95
3.4.4	Dynamic stability analysis	101
3.4.5	Effect of the integration algorithm on the results	103
3.4.6	Effect of time step versus waiting time	107
3.5	Conclusions	108
3.5.1	Comparison with related work	108
3.5.2	Limitations	111
4	Concept future solver	113
4.1	RAMSES overview	115
4.2	Design methodology	119
4.2.1	Power system components and matrix building	120
4.2.2	Electronic equivalents	124
4.2.3	Value range profiling	129
4.3	MSC architecture	131
4.3.1	Local cells	131
4.3.2	Global architecture	132
4.3.3	Topological mapping and value mapping	135
4.3.4	Mathematical operations	135
4.3.5	Inaccuracies and effect in the linear operations	136
4.3.6	Interface between the MSC and the RAMSES flow	138
4.3.7	Timing	139

4.4	Numerical results	140
4.4.1	Selection of mapping ratios	140
4.4.2	Linear system solving	141
4.4.3	Integration into the RAMSES flow	151
4.5	Conclusions	160
5	Conclusions	161
5.1	Future work	162
5.1.1	On the existing solver	162
5.1.2	On the conceptual future solver	163
5.2	Final discussion	165
A	Test power system topologies	167
B	Common modeling assumptions in power systems	171
B.1	Phasor representation	171
B.2	Balance in electrical quantities	174
B.3	Symmetry in the network	177
B.4	Power considerations	179
B.5	Per-unit representation	182
C	Numerical integration	189
C.1	Explicit and implicit methods	189
C.2	Numerical properties of methods	190
C.2.1	Convergence	191
C.2.2	Order	191
C.2.3	Stability	191
C.3	Truncation error	192
	Bibliography	220
	Curriculum Vitae	221

List of Figures

1.1	A schematic of the traditional structure of a power system [1]	6
1.2	The power system as a black box	8
1.3	The operation cycle of a stakeholder	8
1.4	The effect of a faster operation cycle for a stakeholder	9
1.5	Building blocks of an Energy Management System (EMS) of a utility operator .	9
1.6	The concept of DER aggregation	18
2.1	Power system dynamic simulation domains for different time scales - adapted from [2, 3]	22
3.1	Overview of the multi-platform system	42
3.2	Levels of the multi-platform system	42
3.3	Photo of the multi-platform system	43
3.4	Overview of the existing mixed-signal computer [4].	44
3.5	Partitioned solution scheme for power system simulation equations	44
3.6	Generalized π model of a generic power system branch	46
3.7	The complex two-port network for a branch connecting buses f and t and the effect it has on the building of the Y matrix	48
3.8	The complex one-port network for a shunt element on bus s and the effect it has on the building of the Y matrix	49
3.9	Power system topological mapping into an electronic resistor network equivalent.	50
3.10	Schematic of an electrical branch of the existing FPPNS	51
3.11	Detail of the implementation of a slice of the FPPNS using discrete electronics	52
3.12	A stack of four FPPNS slices	52
3.13	Synoptical schematic of a node of the FPPNS	53
3.14	Configuration of the digital part of the existing computer prototype (Altera Cyclone III FPGA).	56
3.15	Stability region of the Forward Euler method and the 2-step Adams-Bashforth method	57
3.16	Pipelined versions of the FE and the AB2 algorithms	58
3.17	Datapath of the synthesized pipeline for generators that are modeled with the classical generator model of (3.30) using the FE integration scheme of (3.28) . .	60
3.18	The USB controller and the shared RAM that is interfaced to the FPGA	60

List of Figures

3.19 Schematic diagram of the timing break up of the operations of the FPPNS for one simulation step	61
3.20 Diagram of the timing break up of a computing pipeline	62
3.21 Architecture overview of <code>elab-tsaot</code>	63
3.22 Interface design pattern for the SS and TD engines	63
3.23 Hardware Abstraction Layer (HAL) of the dedicated hardware	64
3.24 The model-view-controller (MVC) software architectural pattern used to implement the frontend of <code>elab-tsaot</code>	66
3.25 Physical and schematic representation of parasitics for an electrical branch that contains a potentiometer	70
3.26 Schematic representation of the resulting electrical circuit taking into account the node parasitic capacitances	77
3.27 An RC circuit in which R and C are in series.	79
3.28 Calibration procedure	84
3.29 Sample topology with 5 electrical branches and 5 electrical nodes	86
3.30 Sample radial topology	91
3.31 Sample meshed topology	93
3.32 Maximum absolute voltage error for radial and meshed topologies of increasing size	94
3.33 Rotor angle oscillations of generator #3 of the 18-bus topology using different simulators	96
3.34 Rotor angle oscillations of generator #11 of the 59-bus topology using different simulators	97
3.35 Rotor angle oscillations of generator #54 of the 59-bus topology using different simulators	97
3.36 Reduced schematic of the 59-bus test case	98
3.37 Timing breakup of computation time for transient simulation of the 18-bus . .	100
3.38 Screenshot of the Analysis editor of the <code>elab-tsaot</code> visualizing results for a n-1 branch contingency analysis on an 18-bus system	102
3.39 Minimal time-step FPPNS error compared to PC software simulation reference for the 18-bus test system	103
3.40 FPPNS error due to digital imprecision for different time steps for the 18-bus test system	104
3.41 FE instability while AB2 succeeds in retaining the stability of the numerical solution	104
3.42 CCT for branches #4, #10, #30 of the 18-bus system with varying timesteps using FE & AB2	106
3.43 CCT for branch #30 of the 18-bus system with varying timesteps using FE & AB2, in a calibrated and an non-calibrated FPPNS environment	106
3.44 Internal angle of generator #4 after a transient event in the 18-bus case, for different ADC waiting times.	107

4.1	A schematic overview of the proposed mixed-signal computer (MSC)	114
4.2	The flow of the RAMSES transient simulator with the linear algebra operation identified in red	118
4.3	The two-port, four-pole network (2 poles per port) defined for a branch by (4.8) and the effect it has on the building of the \mathcal{Y}	120
4.4	Empirical pie chart of occurrences of branch types for several typical power systems with sizes ranging between 3-15k buses	121
4.5	The one-port, two-pole network (2 poles per port) defined for a shunt element by (4.9) and the effect it has on the building of the \mathcal{Y}	122
4.6	The 2×2 diagonal correction that arises from the dynamic behavior of an injector and the effect it has in the augmenting of \mathcal{Y} to $\tilde{\mathcal{Y}}$	123
4.7	An equivalent of a power system bus that follows the real decomposition, the ordering and the sign convention for the MSC	123
4.8	An outline of an electrical two-port network used to map a power system branch	124
4.9	Examples of the effect of the TPN building blocks (potentiometers and VCCS) to the g-parameters matrix of the TPN	125
4.10	Schematic of the digital part of a TPN	127
4.11	An outline of an electrical one-port network used to map a power system shunt element or a diagonal correction	128
4.12	Schematic of the digital part of an OPN	129
4.13	Synoptical diagram of an injection and measurement node of the MSC	130
4.14	Schematic of the digital part of an IMN	130
4.15	Schematic of a local cell (LC) of the MSC	133
4.16	Connectivity detail and simplified schematic of a TPN connection multiplexer (MUX)	133
4.17	Global view of the digital architecture of the MSC	134
4.18	Relative ∞ -norm errors that are introduced in the electronic equivalent of the Γ matrix	142
4.19	Relative ∞ -norm errors that are introduced in the electronic equivalent of the Γ^{-1} matrix	142
4.20	Relative ∞ -norm errors that are introduced in the electronic equivalent of the \mathcal{I} vector	143
4.21	Relative ∞ -norm errors that are introduced in the electronic equivalent of the \mathcal{I} vector; dynamic scaling of the current mapping ratio ρ_I is followed as per (4.36)	144
4.22	Relative ∞ -norm errors that are introduced to the voltage solution \tilde{V} due to conductance and current inaccuracies of the MSC	145
4.23	Relative ∞ -norm errors introduced to the voltage solution \tilde{V}' due to the quantization of the voltage ADC of the IMNs	146
4.24	The schematic of the electronic equivalent of an 11-bus system created in OrCAD Capture	147
4.25	Thumb-rule diagram for the order of magnitude of parasitic capacitances on different electronic design paradigms	148

List of Figures

4.26 Transients of the node of voltages of the MSC-mapping of the 11-bus system using a current source slew rate of $t_{itr} = 10 \text{ ns}$	149
4.27 Transients of the node of voltages of the MSC-mapping of the 11-bus system using a current source slew rate of $t_{itr} = 1 \text{ ns}$	150
4.28 Transients of the node of voltages of the MSC-mapping of the 11-bus system assuming uniform parasitic capacitances of $C = 10 \text{ fF}$	150
4.29 Voltage of bus #7 of the 11-bus system after a 60 ms 3- ϕ fault on the same bus .	152
4.30 MSC invocations for each time instant for the transient scenario on the 11-bus system	153
4.31 Average penalty on the number of Newton (internal) iterations for one time (external) iteration	154
4.32 Voltage of bus #4072 of the 77-bus system after a 60 ms 3- ϕ fault on the same bus	156
4.33 MSC invocations for each time instant for the transient scenario on the 77-bus system	157
5.1 Most generic sparsity pattern of the A_i matrix of injectors	164

List of Tables

1.1	Operations in power systems	10
1.2	Smart grid technologies	15
1.3	Benefits of distributed generation	17
2.1	Abstraction layers of power system computing	23
2.2	Linear algebra libraries in power system research and applications	26
2.3	Analogies between flow networks, power systems, and analog electronic networks	36
3.1	Time domain and Laplace domain	77
3.2	Stages of inaccuracy of linear operations performed by the mixed-signal computer	82
3.3	59-bus test case maximum absolute rotor angle deviation for TSA	98
3.4	Speed comparison between different engines for the TD simulation of the 18-bus system	100
3.5	Timing break-up between PC, USB communication and the FPPNS for a transient stability operation	101
3.6	Timing results summary for the n-1 branch contingency analysis	101
3.7	Timing results summary for the CCT analysis	102
3.8	n-1 branch contingency results for different integration algorithms and time steps	105
3.9	Characterization of the existing platform	109
4.1	Comparison of the roles of different platforms and domains in the FPPNS and the MSC	115
4.2	Different models for different parts of the branch model of Fig. 3.6	122
4.3	TPN modifications and their ability to represent branch types	125
4.4	TPN modifications, their complexity and their primary usage for branch type representation	128
4.5	Interface between the RAMSES software and the MSC platform	138
4.6	Empirical settling times for different parasitic capacitances and current sources slew rates	149
4.7	Free design parameters of the MSC and their effect in the operation	151
4.8	Test cases to validate the design guidelines of table 4.7	153
4.9	Effect of analog inaccuracy to the average penalty on Newton iterations with respect to the bit resolution of the current injections	155

List of Tables

4.10	Effect of analog inaccuracy to the average penalty on Newton iterations with respect to the bit resolution of the voltage measurements	155
4.11	Effect of analog inaccuracy to the average penalty on Newton iterations with respect to the bit resolution the reconfigurable conductances	155
A.1	General overview of systems to be examined	168
A.2	General linear algebraic properties of the admittance matrix Y of power system test cases	169

Introduction

The purpose of this work is to create *a computing platform, dedicated to power system simulation*. First of all it is important to understand, *why* such a platform is necessary. In order to understand this motivation, an overview of modern power system and its stakeholders is given in chapter 1. The operations that are performed by the stakeholders are described and their time intensity is assessed. The environment is shifting however, with radical changes being imminent, in what has been termed as the advent of the “smart grid”. The triggers of this change are presented and the premises of this new future state of the grid are shown. It is partly this change, that calls for faster power system analysis tools.

Equally important is to see *what* is expected by a power system analysis platform. Special focus is given to simulation of the dynamics of the power system, since this is the major focus of this work. Through modeling of the power system components, each analysis is associated with a mathematical formulation. The latter is handled by algorithms dedicated to it. The algorithms are concerned with *how* the problem is solved. They are the ones to be executed on the hardware computing platforms, i.e. *where* the problem is solved. This logical chain and the coherence that is necessary between its components is the focus of the first part of chapter 2.

In this course it will become evident, that linear algebra has a central position in power system computations. This is related to the interconnected nature of the system under study, which results in large matrix-vector relations between the quantities of interest, e.g. voltage, currents, and admittances. Linear algebra qualifies as one of the most challenging scientific computing domains for conventional digital computing. The state of the art of conventional digital platforms with favorable computing characteristics for linear algebra & power system applications is reviewed in the second part chapter 2. As an alternative to conventional digital computing the analog electronic paradigm is presented, which is the main focus of the next chapters of this work. Both analog electronic networks and power system topologies are *flow networks*, and this underlying affinity makes analog electronic computing particularly pertinent to power system problems.

Chapter 3 presents a hardware platform that is dedicated to the transient simulation of power systems. The platform, termed Field Programmable Power System Network (FPPNS), has an analog and a digital part. The analog part handles the linear algebra operations, and

the digital one the solution of the differential algebraic equations that describe the dynamic behavior of the components of the system. A USB connection is used to connect the FPPNS to a conventional PC. On the PC side a co-designed software application has been created. It acts as a user interface as well as a wrapper of the functionality of the dedicated hardware. The mathematical properties of the FPPNS are thoroughly examined in the chapter. The inaccuracies of the hardware are examined and their effect on the final results provided by the FPPNS is shown. The system is evaluated across nine different criteria, and a comparison with related state-of the art is conducted.

Despite the favorable performance characteristics of the realized platform, certain limitations have been identified. Hence, chapter 4 introduces a concept design for a future generation of a mixed-signal computer that overcomes most of the limitations of the current prototype. The new design is again mixed signal, analog and digital and is termed Mixed Signal Computer (MSC). An a priori design objective was to integrate the MSC into the flow of the transient simulator RAMSES [5, 6]. A software emulator of the proposed hardware platform has been created. With the help of the emulator a thorough exploration of the design space of MSC parameters has been conducted. The effect of individual parameters to the overall accuracy of the results has been examined and related design guidelines have been proposed. One suggested implementation has been successfully utilized in the RAMSES flow to simulate small and medium sized systems. For the linear algebra operations, speedups of four orders of magnitude have been achieved. This translates to respectable overall speedups of the simulation. Finally, conclusions and future perspectives of this work are given in chapter 5.

This work lies in the cross-field of three distinct and vast domains: power systems, electronics, and computing. An exhaustive presentation of every encountered topic is not realistically feasible and is out of the scope of this manuscript. When required, the reader is referred to related literature. Where additional insight on peripheral topics has been deemed useful, dedicated appendices have been included.

Contributions of this work

Hereunder the major original contributions of this work are listed.

- A detailed **list of operations and analyses in the modern power system environment** has been compiled.
- An overview of the emerging **future of the power system**, termed the “smart grid” has been presented.
- The critical **role of linear algebra operations in power system computations** has been identified and related literature has been reviewed.
- The importance of the **coherence between the underlying platform and the algorithm**

that is executed has been highlighted. In view of this observation, a literature review on hardware platforms that are dedicated to linear algebra operations has been compiled.

- The **limitations of conventional digital electronic hardware** are noted and analog computing is proposed as an alternative. The state of analog computing is given, again with a particular focus on power system applications.
- A set of plausible **evaluation criteria for analog computers** is drafted. These criteria are used to evaluate the main results of this work.
- A **hardware prototype** dedicated to the analysis of power systems has been created. The system spans along a conventional host PC and a dedicated hardware platform.
 - On the PC side, **dedicated software** has been written. It acts as a user interface and it exposes the functionality of the dedicated hardware to any programming language through an API.
 - The dedicated hardware consists of an analog and a digital part which is synthesized on reconfigurable hardware. The *author did not take part in the actual implementation* of the platform. He participated in the **design of the system and the determination of its specifications**, especially on the digital and the communication interface side.
 - The **mathematical properties of the analog part** of the platform have been detailed. The accuracy of the actual implementation has been thoroughly studied.
 - An investigation has been conducted on the **effect of numerical integration** on the quality of the final results of the dedicated hardware.
 - The **timing properties of the analog part** of the hardware have been investigated and an aggregate simplified model has been proposed.
 - The **limitations** of the proposed design are exposed.
- A **conceptual future solver** that overcomes the limitations of the existing prototype has been proposed, termed Mixed Signal Computer (MSC). It is again hybrid (analog-digital) and dedicated to power system simulation. A major design goal has been to integrate the proposed hardware in the flow of the RAMSES [5, 6] power system transient simulator software.
 - The mathematical properties of the MSC are presented. An **analysis of the accuracy and the timing** of the operations is in the same vein as in the case of the existing realized prototype.
 - A fully parameterizable software emulator of the MSC has been created. It has been used to get results on the application of the MSC in atomic linear algebra operations, as well as on its integration into the RAMSES flow. Based on these results an **exploration of the design space** has been conducted and design guidelines are proposed.

1 Modern power system landscape

In this chapter an overview of the architecture of the power system is given. Its traditional structure is presented alongside a brief description of the stakeholders that participate in its design, planning and everyday operation. A list of operations of the stakeholders is compiled and characterized for their time intensity.

The traditional way to operate the grid is changing. Reasons that call for that change will be highlighted. The prospective evolution of the grid into the future is what is often called the “smart-grid”. Premises of it will be postulated and the side-effect it has to power system analysis tools will be shown.

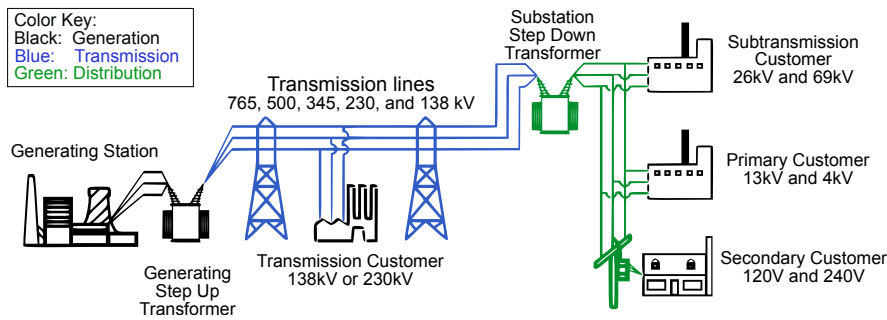


Figure 1.1 – A schematic of the traditional structure of a power system [1]

1.1 Current architecture

The power system has been named the “largest man-made machine ever”. It can be defined as a complex socio-economic-technical system, featuring strong interrelation between societal needs and expectations, market operations and technical infrastructure. Electricity is conceivably the most multipurpose energy carrier in our modern global economy, and it is therefore primarily linked to human and economic development. Electricity growth has overtaken that of any other fuel, leading to ever-increasing shares in the overall mix. This trend is expected to continue throughout the following decades, with large parts of the world population in developing countries appealing to be connected to power grids [7].

Electrical power systems have been traditionally designed and operated taking energy from high-voltage levels, and distributing it to lower voltage level networks. There are large generation units connected to transmission networks. In these networks there is a bulk transport of electricity, with central coordination of control. Demands are passive and uncontrollable, connected to distribution networks. Distribution systems are also passive and, in the lower levels of voltage, radial in operation. They are designed to accept power from transmission systems and distribute to customers, generally with unidirectional flows [8]. Such a system is depicted in Fig. 1.1.

It is made up of many geographically dispersed components and it can exhibit global change almost instantaneously as a result of local actions. Actions are exerted on the components of the system by the *stakeholders*. As a stakeholder we consider anyone that is in interaction with the power system, i.e. that is affected by it or that can affect the system through his actions. It is out of the scope of this work to present a thorough stakeholder analysis of the power system. Instead the following sample entities are defined indicatively for the facilitation of the analysis that follows. These entities have arisen after the liberalization and the unbundling of the top-down monopolies in power system operators. They are the conceptual players in the power system world that have an interest/utility in fulfilling their objectives. These objectives may be financial profit, or simply the utility of serving their need for electricity.

ISO refers to the Independent System Operator. ISOs coordinate, control and monitor the

operation of the system in a given region. In some regions around the world the ISO also has market authorities.

GENCO stands for Generation Company, i.e. a company engaged solely in producing electricity, normally by owning or maintaining energy production facilities (e.g. generators).

TRANSCO stands for Transmission Company, as understood in the market sense, i.e. a company which owns or maintains energy transmission facilities and the business object of which is the transmission of energy.

DISCO stands for Distribution Company, i.e. a theoretical (or real) company which owns or maintains energy distribution facilities and the business object of which is the distribution of energy from the transmission level of the grid to the *End users*

End users are the final consumers of electricity, residential, commercial or industrial. As it will be explained in sections that follow, there is a recent trend that the end users also feed electricity back into the grid.

The above definitions of the ISO, the GENCO, the TRANSCO and the DISCO are only indicative and always depend on the local power system governance and regulatory scheme. Often the boundaries between their responsibilities are blurred.

The aim of the stakeholders is to drive the system in a way that generates the most utility to them. Utility can be monetary or anything relevant, e.g. proper functioning of the grid within its technical limits for some ISOs. An illustrating abstraction can be introduced that is drawn from general system theory. For each stakeholder, the power system can be viewed as a black box \mathcal{F} that has two kind of inputs: the *uncontrollable* inputs that the stakeholder cannot control u_u , and the ones that can be controlled, termed *controllable* u_c and *semi-controllable* inputs u_{sc} . The former are termed controllable because the stakeholder has a large degree of freedom on them $[u_{cmin} - u_{cmax}]$, and the latter are termed semi-controllable because the degree of freedom on them $[u_{scmin} - u_{scmax}]$ is smaller. The goal of each stakeholder is to manipulate the controllable and the semi-controllable inputs so as to keep some system “state” variable x within some “good operation limits” $[x_{min} - x_{max}]$. These state variables (e.g. voltage magnitudes, angles, power flows, prices) are different for each stakeholder and they are all subject to equality and inequality constraints, to ensure proper operation of the system. This concept is illustrated in Fig. 1.2.

The way each stakeholder acts on the system is as follows.

Acquisition is the phase where the stakeholder collects data on the system, i.e. “measures” variables of interest to him.

Analysis is the phase where the stakeholder runs his “computation routines” in order to determine his actions.

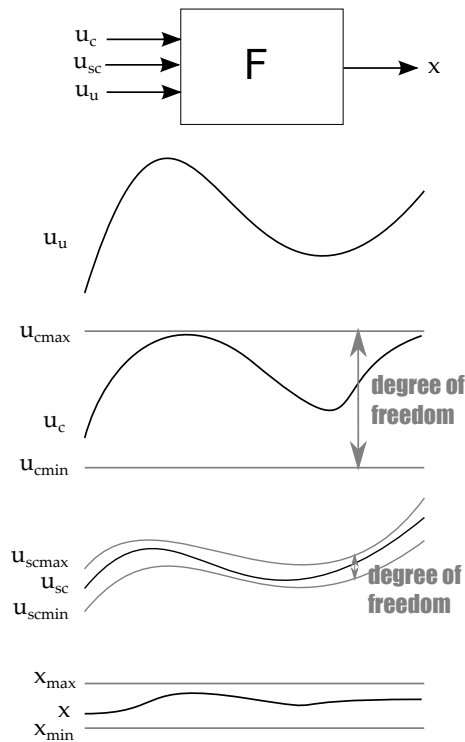


Figure 1.2 – The power system as a black box

Actuation is the feedback of the stakeholder back to the grid, in terms of values for the controllable and the semi-controllable variables.

The above cycle is executed repeatedly, in what is here called the *stakeholder operation cycle*, shown in Fig. 1.3.

The faster the stakeholder cycle, the better the ideal x curve is approximated as shown in Fig. 1.4. In the ideal case where all acquisition, analysis and actuation phases were instantaneous, the x curve would be as close to the ideal one as possible. It is therefore to the interest of the stakeholder to improve the speed of every element of the cycle of his operation cycle.

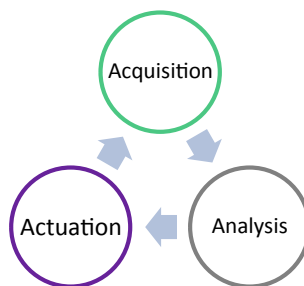


Figure 1.3 – The operation cycle of a stakeholder

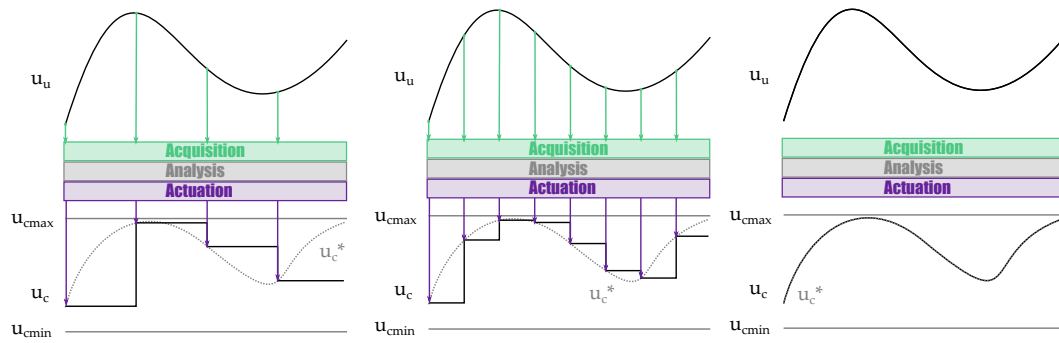


Figure 1.4 – The effect of a faster operation cycle for a stakeholder

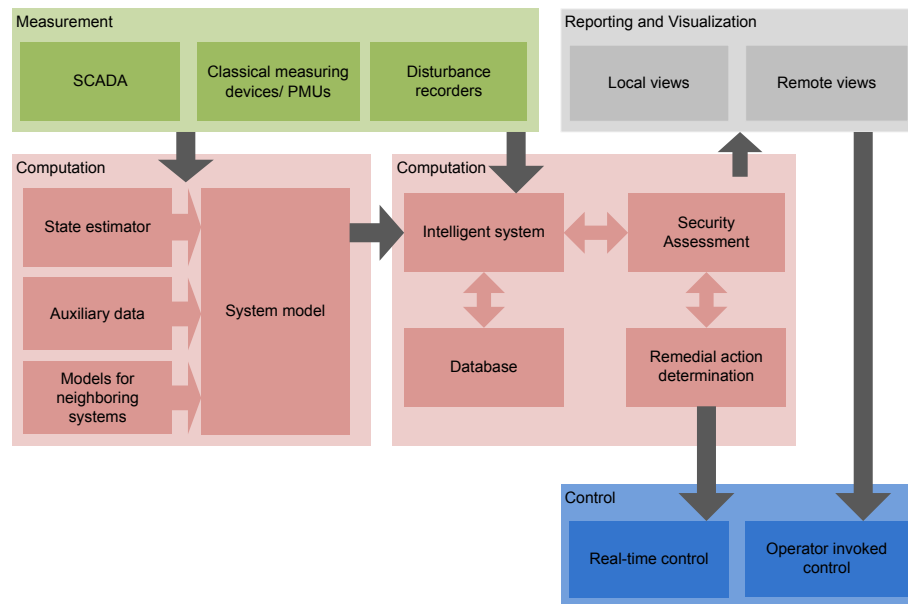


Figure 1.5 – Building blocks of an Energy Management System (EMS) of a utility operator

For utility operators, these operations are performed in computer-aided environments called energy management systems (EMS). EMSs have been present since the era of vertical integration in power systems. EMSs are a point of monitoring, control and optimization for a part of a grid. For that to be enabled, an EMS employs a wide range of supervisory control and data acquisition (SCADA) functionality. Such a system is shown schematically in Fig. 1.5; color coding from Fig. 1.3 has been retained.

Table 1.1 provides a non-exhaustive list of operations commonly executed in the frame of a modern power system. Column “Level” refers to the level of the power system, as in table 1.2.

Table 1.1 – Operations in power systems

Operation	Algorithms	Software & Equipment involved	Level	Stakeholders	Time intensity
Economical feasibility studies (equipment sizing)	Operational research		GTD	ISO, GENCO, TRANSCO, DISCO	1
Transmission expansion, Optimal compensation device/capacitor placement	Prediction scenarios/methods, (Available Transfer Capability (ATC) methods), VAR optimization methods		T	TRANSCO	1
Power flow analysis	Power flow algorithms, network reduction algorithms		*	ISO	2 .. 3
Static Security Assessment (SSA), contingency analysis, emergency switching plans	N-1/N-P contingency analysis		*	ISO, GENCO, TRANSCO, DISCO	2
Protective device coordination/sizing, Distance relaying configuration, Reclosure synchronization	Short circuit analysis (ANSI C37/IEC 60909)	Distance/protection relays	*	ISO, GENCO, TRANSCO, DISCO, 1-2	2
Voltage stability analysis	Continuation power flow (CPF) methods, Bifurcation analysis		*	ISO	2
Power quality/Harmonic (distortion) analysis	Fourier analysis		TD	ISO, TRANSCO, DISCO	2
Small-signal Stability analysis, Power System Stabilized (PSS) configuration	Eigenvalue algorithms, sensitivity analysis		T	ISO, TRANSCO, DISCO	1-2
Dynamic Security Assessment (DSA)	Time domain (TD) simulation (ms scale), Symmetrical components theory		TD	ISO, TRANSCO, DISCO	2
Electro-Magnetic Transients analysis (EMT), Lightning protection	Time domain (TD) simulation (sub-ms scale)	EMTP	TD	ISO, TRANSCO, DISCO	2
Reliability assessment	Probabilistic methods, Index based quantification		*	ISO, GENCO, TRANSCO, DISCO	2

Optimization, Unit Commitment	Unit commitment algorithms, Optimal Power (OPF) algorithms, emissions/security constrained Market simulation	*	ISO, GENCO, TRANSCO, DISCO	3
Market decision making/bidding	Market agents	*	ISO, GENCO, End users	4
Load forecasting	Statistical methods, Intelligent systems (ANN, etc.)	C	(TRANSCO), DISCO, End users	3-4
Price/Load forecasting	Artificial intelligence, neural networks, times series analysis	*	Microgrid controllers, market agents	3-4
State Estimation, Monitoring/Situational awareness (engineering diagnostics, billings)	WLS, Kalman filters	TDC	SCADA, Wide Area Measurement Systems (WAMS)	3-5
Active power flow and frequency control, Power electronic interface control	AGC & FACTS configuration	TD	Prime movers (speed governing)	5
Smart metering		C	Smart meters	4
Intelligent load shedding		DC	Breakers, substations	5
Home/Building automation		C	Distributed electronics (sensors, actuators, comm. infrastructure)	4-5
Transmission/distribution line design	Line parameter estimation	TD	TRANSCO, DISCO	1
Reactive power flow and voltage control	Reactive compensation techniques, AVR configuration	TD	Tap changing/ULTC transformers, shunt/series capacitor banks/reactors, static var systems	4-5
Sub-synchronous resonance analysis	Prime movers and related controls	GT	GENCO, ISO	1-2
Controlled islanding schemes, Black start planning, System restoration		GT	ISO	1

Insulation coordination	Time-domain simulation (sub-ms scale)	TD	TRANSCO, DISCO	1
Power factor correction		Compensation device / - capacitor banks	End users	1

The column “Time intensity” refers to how tight are the time margins for the operation to be finished. Increasing number denote tighter requirements.

1: planning/design operations These analyses are often performed only once, in a planning/design phase, hence, they have no practical time constraints.

2: periodic offline operations These are analyses that are executed once per day/week. They are performed more than once, but their periodicity is so infrequent that poses almost no practical limits to their execution.

3: periodic online operations¹ These are analyses that executed once per a tighter time frame than the above, e.g. in the execution frame of a “measurement - data acquisition - state estimation - analysis” cycle of an Energy Management System in a system operator (15 minutes). Or in the bidding-clearing time frame of an energy exchange (5 minutes).

4: real-time operations¹ These operations refer to operations that are performed continuously, i.e. that are subject to some real-time constraints.

5: hard real-time operations In these case of real-time operations the constraints are “hard”, in the sense that if they are not met, some part of the expected functionality in the system will fail. These task are the most demanding tasks from a computing point of view.

1.1.1 The need for change

It is now widely accepted that changes to the aforementioned traditional architecture are necessary. Among others, factors contributing to this reshaping are the following [10].

- Different governance frame: liberalization, shift from vertical monopolies to deregulated markets.
- Consumer consciousness, power “ethics”, e.g. consider post-Fukushima declining support for nuclear.
- Aging infrastructure: early post-WWII.
- Increase in demand quantity beyond expectation, e.g. demand doubled since '60s.
- Changes in demand nature: changing peak hours, load patterns, increased quality-of-service demands, differentiated products.

¹It is useful to note a distinction raised in [9], between the terms *online* and *real-time* when used in the power system domain. Online refers to the results of the analysis tool being available to the SCADA/EMS of the operator, while real-time refers to the results of the analysis being ready within a time frame that is deemed “real-time” for the specific application.

- Advances in technology: storage, distributed generation, smart metering.
- Climate change and related political/public opinion pressures.

As very significant investments will be required to simply renew this infrastructure (\$6.9 trillion capital investment spanning 15 to 25 years reported in [11]), the most efficient way forward is to incorporate innovative technologies and solutions when planning and executing this renewal [12].

1.2 Future architecture: the Smart Grid

The term *smart grid* is used interchangeably to denote a plethora of emerging technologies in the field of power systems. It would be more precise to refer to the smart grid as *the grid of the future*, but for conventionality the use of the term is retained throughout this work. Two underlying conceptual points of the smart grid are its *de-centralization* and the existence of *localized intelligence*.

History and organizational analysis suggest that a centralized point of control comes in the way of rapid scaling of value creating activities, often becoming the bottleneck limiting evolution of ideas and novelty of executions [13]. Furthermore, a bottom-line premise of the smart grid is to be self-healing. This implies some sort of underlying distributed control of the system with individual components as independent intelligent agents. These agents would compete and cooperate to achieve global optimization [14].

A complex socio-technical system featuring similar requirements is the internet. And as Robert Metcalfe urges, *it is of utmost importance to mine the history of the internet, to learn from that experience, and to find the correct lenses to look at the smart grid problem* [15]. The genius in the internet is in its flexibility and versatility. For the smart grid to follow the success story of the internet, it needs to endorse this versatile design and encourage innovative participation from all stakeholders.

For that the following premises are set [13]: The smart grid is the future evolution of the power system that:

- will meet all our energy demands (not just current electricity demand),
- will be powered almost totally by renewable resources,
- will operate all its components at their maximum allowable capacity,
- command and control intelligence will be pervasively distributed and sub-second responsive, and
- will be able to self heal.

Table 1.2 – Smart grid technologies

Tech	Level ¹
Substation automation	TD
Distribution automation	TD
Phasor measurement	GTDC
Smart metering	C
Smart appliances	C
Home automation	C
Demand response	C
Electric vehicles	GC
Energy storage	GC
Distributed generation	G
Renewable sources	G
Communication infrastructure	GTDC
DER aggregation schemes	GTDC

¹ G: Generation; T: Transmission; D: Distribution; C: Consumption

The definition of smart grid can also depend on local conditions; different countries can have very different starting points for the progress towards smart grid. Deployment of smart grid technologies will occur over a long period of time, adding successive layers of functionality and capability onto existing equipments and systems.

Technology is the key consideration and it can be defined by certain technical characteristics: e.g. predictive, integrated, interactive, optimized, flexible, accessible, reliable, economic, and secure. Broadly speaking, three major technological domains of the smart grid are distributed intelligence, communication technologies, and automated control systems [16]. So far, research has been focusing on all three of them. Among others, impressive work has been carried out in technologies such as: substation automation, distribution automation, phasor measurement (PMU), smart metering, smart appliances, home automation, demand response, electric vehicles (PHEV), energy storage, distributed generation, renewable sources, communication infrastructure, DER aggregation schemes, etc. Scrutiny of the aforementioned is out of the scope of this paper. Table 1.2 summarizes them according to which level of the power system they pertain to: Generation, Transmission, Distribution or Consumption.

The next section presents recent advances in the field of distributed energy resources (DER) that seem to be greatly contributing to shifting the power system paradigm away from the prevalent centralized production scheme. The vehicle through which DER are to impact the structure of the power system is DER aggregations, reviewed in section 1.2.2.

1.2.1 Distributed energy resources

Perhaps the most heavily researched field in smart-grid related technologies is distributed energy resources (DER). DER include distributed generation (DG), responsive loads and storage systems alike. Smart grid advances regarding the loads mainly focus on their responsiveness (see table 1.2). Storage systems would add to the flexibility of the system, providing the ability to shift energy consumption/production in time. Distributed generation is the sector that has received the most attention of the three and seems to have the greatest potential.

There is an ongoing debate on the definition of the distributed generation. To the author's opinion, the most apt definition has been given in [17]: DG is defined as an electric power source connected directly to the distribution network or on the customer side of the meter. Recent developments in DG include the following [18].

- Improvement of the efficiency of solar cells up to 20-24%.
- Increase in the capacity of wind generators from few kW up to several MW.
- Development of micro-, bio- and multi-fuel-CHPs (combined heat and power) plants to replace the conventional ones.
- Development of fuel cell technologies.
- Increase of efficiency and capacity of storage devices.
- Introduction of new renewable energy resources as tidal generators, small hydro generators, etc.

Reference [19] provides a comprehensive insight on DG technologies and their respective status quo.

The projected benefits of increased DG penetration seem to be in accord with the concept of sustainable development. This becomes clear once the benefits of DG are categorized in four domains as shown in table 1.3 [20, 8].

For completeness sake, it should be noted that high-degree penetration of DG into the bulk grid can also have adverse effects. These mostly relate to the technical operation of the system [8], including inversion in the energy flow, difficulties in voltage control and in management of reactive power. Also, despite the professed positive impact on reliability, it has been argued that there can also be a negative impact due to the non-dispatchable nature of some types of generation, such as intermittent solar panels, wind turbines, tidal generators, etc. Furthermore, the unpredictability implied by dependance on weather conditions for those generation units would require additional conventional standby capacity.

Up to now, DER have been used to displace energy from conventional generating plants but not to displace their capacity as they are not visible to system operators [23]. However, it is

Domain	Benefit
Technical	increase overall system efficiency e.g. CHP, bypass 'congestion' in existing transmission grids, avoid transmission losses, provide network support or ancillary services, increase flexibility, continuity and reliability of supply, positive effect on transient stability [21]
Social [22]	enable power equity, diversification of energy sources to enhance energy security, support for competition policy, improved utilization of local resources, local job creation
Environmental	increase overall system efficiency e.g. CHP, bypass 'congestion' in existing transmission grids, avoid transmission losses, provide network support or ancillary services, increase flexibility, continuity and reliability of supply
Economic	uncertainty in electricity markets favours small generation schemes, cost-effective improvement of power quality and reliability, new market opportunities, facilitate investment (easier real estate, easy acquisition of equipment)

Table 1.3 – Benefits of distributed generation

now clearly understood that DG needs not only to displace the energy produced by central generation but also to provide all the associated flexibility and manageability. So there is a need to move away from the *fit and forget* approach that has characterized DG installation so far [20]. As a means of comprehensive integration of DER into power systems various aggregation schemes have been suggested.

1.2.2 Aggregations

Aggregation is the process of linking small groups of industrial, commercial, or residential customers into a larger power unit to make them visible to the electric system (see figure 1.6). Aggregations usually involve loads, local storage and DG. Thus, load or generation profiles of individual consumers and/or small generators appear as a single unit to the system. Building up a large and flexible portfolio enables aggregators to operate DER and to provide services to the power system, e.g. system balancing. It is by combining these features (more flexibility, lower operating costs) that aggregations will reduce the gap to profitability of DER units [16]. As mature representative examples of such aggregation schemes one can mention the microgrid (uGrid)[24, 25] and the virtual power plant (VPP) [26].

A *microgrid* (see figure 1.6) usually consists of an aggregation of localized small-scale generation and demand resources connected at a single point of common coupling (PCC), usually at a low voltage level in the grid. The main idea is grouping these resources and presenting

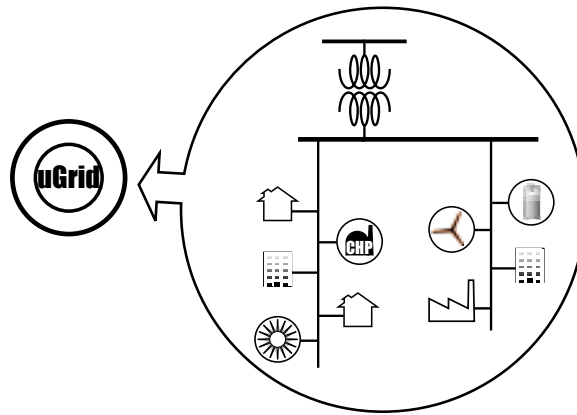


Figure 1.6 – The concept of DER aggregation

them as a single controllable entity to the rest of the grid. In this way, DG becomes visible and given attractive remuneration it can provide services for the grid, e.g it can act as a source of power, or it can provide ancillary services [27]. An interesting feature of a microgrid is that it can operate in two modes: both connected to the grid and in an autonomous islanded mode, increasing the flexibility of the host power system. To the utility the microgrid can be thought of as a controlled cell of the power system. To the customer it can be designed to meet his special needs and provide additional benefits. An elaborate presentation of the technological status quo of microgrids, their management and related projects around the world can be found in [28, 29].

The *virtual power plant* is a similar concept. It consists of a flexible representation of a portfolio of smaller generation and demand resources, in a way that a single operation profile is created. This profile consists of the composite parameters characterising each participating element of the aggregation. As a result, the VPP is characterized as a whole by a set of parameters usually associated with a traditional transmission connected generator, such as scheduled output, ramp rates, voltage regulation capability and reserves [30, 23]. The VPP concept was thoroughly investigated by the European project FENIX ¹.

1.3 The effect of the smart grid on analysis tools

One very important side effect of the advent of the smart grid is the requirement of faster analysis tools for the grid.

- Given the shorter time constants of the inertia-poor, sub-(milli-)second responsive smart grid environment it is a technical requirement that the “real-time” frame is shorter.
- It is to the interest of all stakeholders to perform their analyses faster since this would give them an edge against competition in the open market/operations environment of

¹<http://www.fenix-project.org/>

the smart grid.

- There is smaller experience on the operating mechanisms of the smart grid from a technical point of view (e.g. voltage, transient stability etc.), hence more analyses are required to ensure the safe operation of the grid.

Faster tools mean a shorter time for the “Analysis” part of the stakeholder cycle of Fig. 1.3, which leads to better attainment of the goals of the stakeholders, as per Fig. 1.4. Should substantial speed increase be in hand, new opportunities would arise in the Energy Management Centers (EMC) of the power utilities. Current analyses would be completed within smaller time windows and larger types of analysis (e.g. for larger, more detailed networks) would be possible in the same amount of time. More importantly there would be a phenomenon of *technology push*, in the sense that new types of applications and services could be explored, currently hindered by the unavailability of potent enough tools [31].

2 Power system simulation, linear algebra and computing platforms

This work is particularly concerned with transient simulation. In the latter heavy linear algebra operations with the admittance matrix of the grid are involved. Given the computing intensity, linear algebra operations are a domain where significant performance benefits can be harnessed. Designers of power system applications have been long aware of the fact and have tried to optimize the related software and hardware parts. Different architectures have been employed, a review of which is given in this chapter.

The vast majority of related work concerns conventional digital computing. Certain characteristics of linear algebra operations make conventional computing less appropriate for it. As an alternative analog computing is proposed. Both analog electronic networks and power system topologies are flow networks, and this underlying affinity makes analog electronic computing particularly pertinent to power system problems. The state of the art of analog computing is given in the end of the chapter, with particular emphasis on power system applications.

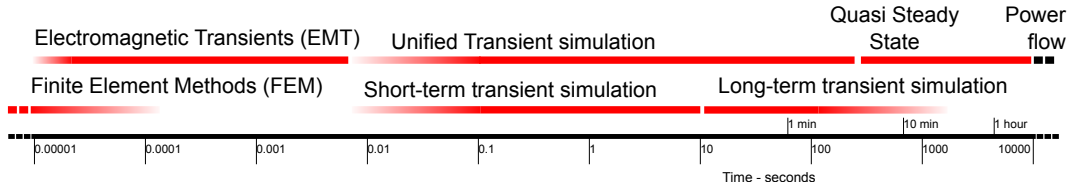


Figure 2.1 – Power system dynamic simulation domains for different time scales - adapted from [2, 3]

2.1 Simulation

Out of the different operations of table 1.1 the focus of this work are the ones primarily concerned with the simulation of the power system. Simulation and its related concept of modelling allow us to get information about how the system will behave without testing it in real life. Simulation is the imitation of the operation of the real-world system over time. Power system simulation may refer to different time scales, as shown in Fig. 2.1.

This is particularly pertinent in the field of power systems, for a range of reasons.

- Due to the massiveness of the system performing real life tests would be too expensive.
- Actually it would not be possible, since the system is in continuous use, i.e. it cannot be stopped for scenario testing.
- The (slow) time constants involved prohibit the realization of all the analyses that would be needed. Therefore it is quite common to perform faster-than-real-time simulations so as to speedup the analysis in question.

A power system simulation application consists of three abstraction layers (1-3), the power system problem, the underlying mathematical problem, and the computing implementation of the latter. These levels and transitions between them are illustrated in table 2.1.

The *power system problem* is the one to be solved. It comes directly from the power system world, e.g. the power flow analysis. The translation of the power system problem to a mathematical problem is done through the *modeling* of the power system. Model selection is a fine process which is determined by the (a) purpose, (b) the time scale, and (c) the detail that is required in the application in question. At level 2, the problem consists of mathematical predicates which can be handled by relevant algorithms. These algorithms are amenable to a *computing implementation*, i.e. a machine code that can be executed directly by a computing platform. The *hardware* computing platform on which the implementation is executing is shown below the dashed line in the table.

It is important that the above is generic, and can be more or less applied to a wide range of analyses. It is equally important to highlight that after model selection is complete, at level 2,

Table 2.1 – Abstraction layers of power system computing

Layer	
3	Power system problem
	↓
2	Mathematical problem
	↓
1	Computing implementation
	↓
0	Platform

modeling

algorithms

runs on

the problem has been completely translated outside the power system domain. From then on, it can be decoupled from power system knowledge and expertise, and can be handled by mathematicians and computer scientists.

As an example, most of what is thought of as “power system algorithms” are actually not in the power system domain at all. When W. F. Tinney introduced the Newton-Raphson (NR) power flow algorithm in 1967 [32], he actually only applied an already established mathematical algorithm, i.e. the Newton-Raphson method, to a typical mathematical problem, i.e. system of non-linear equations. The latter is actually the formulation that results from the modeling of the power flow problem in the power systems domain.

2.1.1 Mathematical formulation of simulation problems

A general formulation of any power system simulation problem includes a set of non-linear differential algebraic equations (DAEs).

$$x = \begin{bmatrix} x_d \\ x_a \end{bmatrix}, \quad \begin{bmatrix} \dot{x}_d \\ 0 \end{bmatrix} = \begin{bmatrix} f_d(x_d, x_a) \\ f_a(x_d, x_a) \end{bmatrix} \quad (2.1)$$

The evolution of state variables x_d and algebraic variables x_a along time is determined by numerically integrating the differential part of the equations f_d . If an initial value of $x_d(0) = x_{d0}$ is supposed for (2.1), then the perfect analytical solution of it along time is $x(t) = \phi_t(x_{d0}, t_0)$. The point of numerical integration algorithms is to approximate this perfect analytical trajectory by a sequence of points $x^{(k)}$ that correspond to respective time instants $t^{(k)}$. For a given point in time t^k a numerical integration algorithm produces a point of the trajectory for time instant $t^{(k+1)} = t^{(k)} + h$, where h is a timestep [33]. The solution of the above is highly dependent on the time scales of interest for the study as per Fig. 2.1. In the extreme case of steady-state studies (e.g. power flow on the far right of the figure), all the variables are algebraic and the equation set is purely algebraic $0 = f_a(x_a)$.

2.2 Linear algebra in power systems

One class of mathematical problems (level 2) that are ubiquitous in power system computing applications are linear algebra operations. The exact amount of time spent in linear algebra operations in a power system algorithm depend on its exact flow.

For example:

- In [34], the linear algebra solver takes 30.3-62.9% of the total execution time for the power flow calculations on an artificial system of 800k buses, and 28.6-80.8% for the power flow calculations on the WECC system.
- In the transient simulator detailed in [5], linear algebra related operations (mostly linear system solving and sparse matrix-vector multiplication) account for 21-58.1% of the total execution time of the program.
- The assembling and factorization of the Jacobian at every step of a quasi-Newton power flow method in [35] takes up 85% of the computing time.

Given the computing intensity of linear algebra operations, the latter are identified as having a significant potential for performance benefits in power system calculations. The operations that are particularly pertinent are (sparse) matrix-vector multiplication (2.2a) and (sparse) linear system solving (2.2b).

$$\text{solve for } y \text{ (} A, x \text{ known)} : \mathbf{y} = A \cdot \mathbf{x} \quad (2.2a)$$

$$\text{solve for } x \text{ (} A, y \text{ known)} : \mathbf{y} = A \cdot \mathbf{x} \quad (2.2b)$$

The matrices involved in these operations are usually exactly or closely related to the *admittance matrix* Y of the topology, which links bus voltages to current injections. It is square, sparse, complex, often symmetric and it arises from the interconnected nature of the grid and formalizes the nodal voltage equations [36, 37, 38]. The size of Y is equal to the number of buses in the system, and in cases of practical interest it is in the order of (tens of) thousands [39, 40, 41].

2.2.1 Algorithms and implementations

The algorithm to solve the matrix-vector multiplication problem of (2.2a) is trivial from a mathematical point of view. However, this basic algorithm can be implemented in a variety of ways: in the ordering of the loops, the storage scheme of the elements, vectorization/grouping of the operands, parallelization techniques etc.

The solution of the linear system problem of (2.2b) is less trivial but well-established direct (e.g. LU decomposition, Cholesky decomposition, QR decomposition, etc.) and iterative techniques (Jacobi, Gauss-Seidel, CG, BiCG, GMRES) exist to solve such systems [42, 43, 44]. Direct methods are called so, because they attempt to solve (2.2b) for an exact x in “one-shot”, meaning in a set sequence of a finite number of algorithmic steps. This is in contrast to iterative methods, which start from an initial guess and try to generate a sequence of improving solution approximations for (2.2b). These approximations converge to the precise solution only to the limit ($iter \rightarrow \infty$).

The realization of these theoretical problem-solving algorithms on computing *platforms* is the focus of *numerical linear algebra*. Numerical linear algebra implementations are usually based on low-level building blocks of elementary operations. These operations are traditionally grouped in three levels of Basic Linear Algebra Subprograms (BLAS) functionality [45, 46, 47]. Usually the performance is heavily dependent on how well the BLAS software realization matches/exploits the specificities of the host platform. This explains the wealth of vendor/architecture-specific/optimized BLAS implementations that are available, e.g. [48, 49].

Higher level linear algebra operations can be constructed by invoking cores of one of the three basic BLAS levels. They usually provide a broader range of linear algebra routines, such as matrix factorizations, linear system solving and eigen-analysis. Given that in most of the cases these libraries are based on some lower level BLAS implementation, their performance depends highly on the performance of the latter. A survey on linear algebra libraries is given in [50].

The major trend on how to include linear algebra algorithms in the flow of a power system application is the utilization of off-the-shelf libraries [51, 52, 53, 54]. Efforts to develop implementations tailored to the specific very needs of the problem also exist [55, 56]. Table 2.2 summarizes linear algebra library utilization in power system research.

The sparse matrix-vector multiplication problem of (2.2a) is part of Basic Linear Algebra Subprograms (BLAS) level 2 functionality under the naming convention $xyyMV$. A performance edge may be gained if special properties of the matrix/vector operands are exploited, such as real/complex entries, (block) triangular/banded/Hermitian/symmetrical structure, sparsity, diagonal dominance, positive definiteness, etc. [56].

For the linear system solving problem of (2.2b), the majority of power system related work uses predominantly direct methods, such as pivoted or not $L(D)U$ decomposition [36, 57, 40, 5, 3], as well as multifrontal Cholesky and QR decompositions [52]. For these implementations, superior results are reported for KLU in [40, 94, 69, 54, 95].

Preconditioned *iterative methods*, sometimes called inexact, have recently emerged into the power system application landscape [70, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 53, 83]. As said, iterative methods are able to reduce the residual of the linear solution $\mathbf{r} = \mathbf{A} \cdot \mathbf{x} - \mathbf{b}$ down to a predefined desirable tolerance, in this way avoiding the extra computational cost of an

Table 2.2 – Linear algebra libraries in power system research and applications

Library name	Related work
SuperLU	[57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67]
KLU	[68, 51, 40, 69]
UMFPACK	[52, 70, 71, 72, 73, 67, 69, 74, 75]
NMath	[76]
ILNumerics	[77]
ALGLIB	[78]
SPARSKIT	[79, 80, 81]
MUMPS	[70]
PARDISO	[58, 69, 82, 83]
PETSc	[84, 85, 39, 86, 87, 34]
SuiteSparse	[88, 83, 89]
CXSparse	[90, 91]
Hypre	[92]
NICSLU	[92]
HSL[93]	[5, 6]

over-solution. They can be more easily parallelized and they are thus deemed more suitable for multi-core platforms [60]. Also, they are not so critically dependent on the ordering of the matrix Y as direct methods [38]. Computationally, iterative solvers prevail only when the number of buses is rather high ($> 21k$ [53], $> 2k$ [99]), and this advantage becomes quite significant for very high number of buses ($> 300k$ [103]). As in the direct method case, iterative solvers are integrated into power system related works mostly as off-the-shelf linear system libraries.

2.2.2 Coherence between the algorithm and the platform

The coherence between specificities of the algorithm and the platform is essential for high performance. This coherence refers to how well the algorithm is adapted to the particularities of the architecture, e.g. a vectorized version of matrix-vector multiplication will run faster on a machine with SIMD instructions. Vice versa, it also refers to how suitable the platform is to the nature/frequency/type of operations executed in the algorithm, e.g. an architecture with high memory bandwidth is more suitable for an algorithm with many memory access operations.

In linear algebra algorithms in general, and the ones for (2.2a) & (2.2b) in particular, the following general observations can be made. Some of them are more pertinent to dense rather than sparse versions of linear algebra operations.

- a. The vast majority of the operations are non-integer arithmetic operations (in contrast

to integer arithmetic, bitwise, logical, etc. operations).

- b. Operations feature data parallelism, i.e. they are usually vectorized, e.g. matrix-vector multiplication can be decomposed in multiple vector inner products.
- c. The operands change often, e.g. a limited number of operations is applied to a large number of operands and not vice versa, e.g. a single multiplication operation has to be performed on every element of a vector in a vector multiplication by a scalar.
- d. The operands usually change regularly/not-randomly, e.g. the elements of a vector are accessed incrementally, one by one in a vector multiplication by a scalar.
- e. There are many loops, e.g. across columns and rows (in the dense case) for a matrix multiplication by a scalar.

Conventional General Purpose (GP) single-core Central Processing Units (CPUs) are not optimally designed to handle such requirements. Especially troublesome are the vectorial operations in matrix manipulations, and the speed limitations in memory access [106, 107]. The performance penalty cannot be leveraged by increasing the clock frequency beyond a practical limit, mainly because of electronic limitations and power consumption issues [108].

The idea of platforms/architectures dedicated to specific problems is based on the obvious observation that routines run faster on hardware specifically made for them. Various architectures have been proposed and established that overcome the linear algebra performance shortcomings of the traditional CPU. Architectures dedicated to power system linear algebra are the ones that most effectively handle the specificities (a)-(e) identified above.

Point (a) is now trivially handled by the inclusion of hardware that can handle non-integer arithmetic operands, e.g. Floating-Point Units (FPUs) that are ubiquitous in modern day processing units.

Point (b) is handled by vectorizing the datapath of the arithmetic unit of the platform. This can be either by the inclusion of (Single Instruction Multiple Data) SIMD instructions into the Instruction Set (IS) of the machine, or by pipelining, e.g. in linear and/or systolic arrays of sub-PUs.

Point (c) relates to the computer horsepower to memory bandwidth ratio of the platform. Linear algebra in that sense is particularly memory-intensive. The issue can be handled by increasing the overall memory bandwidth or by increasing the memory bandwidth locally, e.g. by the introduction of distributed memory (so that PUs may have faster access to their local memory). Generally, exploiting the memory hierarchy has been a major driving force for the advance of linear algebra libraries.

Point (d) strongly relates to point (b) in that the operands of many linear algebra super-operations are vectors. Thus, their implementation may be sped up using automatic counters so that arithmetic operations are performed to incremental positions of a contiguous memory

space - similar to counters in direct memory access (DMA) units. An 1d-vectorization of this kind is found at the core of many sparse matrix storage formats and computation schemes [109].

Point (e) can be effectively tackled by hardware controlled looping that simplifies overhead for loops. The parallelized nature of linear algebra suggests that such operations run best on analogously parallel platforms. It is of no surprise that most platform modifications aim to some sort of parallelization of the computation datapath, since it is nowadays established that performance increase is to be sought after in the parallelization domain [110, 111].

2.3 Dedicated platforms

Most of the platforms combine more than one of the above, in an effort to maximize benefits. Comparative studies on the performance of different platforms architectures on linear algebra applications can be found in [112, 111, 113, 114, 115, 116]. An overview of platform paradigms is given in the following subsections.

2.3.1 SIMD

A kind of a mildly dedicated platform are processors that implement Single Instruction Multiple Data (SIMD) instructions, i.e. that are able to process multiple data within the same instruction cycle or in a pipelined fashion - see also points (b) and (d). Almost all modern processing units have SIMD capabilities (VIS, MMX, 3DNow!, SSE1-5, XOP, FMA3-4, AVX/-512) which can be exploited by linear algebra implementations, e.g. [48]. Of course the latter should take into account the vectorizing capabilities of the hardware, and therefore follow a different coding philosophy. Generally assembling operands into vectors that are fed into higher-level BLAS routines should be sought after, as it makes better use of the capabilities and the memory hierarchy of the platform [117].

The potential of SIMD architectures for power system oriented, linear algebra applications was identified as early as their introduction as “array” or “vector processors” [31, 118, 119, 120, 121], and has enjoyed widespread popularity ever since [122, 123, 124, 125, 117, 90, 91]. SIMD functionality (data parallelism) is nowadays usually integrated into systems with multiple levels of parallelism (instruction and task parallelism) in multi-core platforms (see next section) [122, 123, 124, 125, 117].

2.3.2 Multi-core

A natural extension of the level of parallelism of the platform are *multi-core architectures*. In this work, for simplification purposes, multi-core is understood as having more than one datapath, no matter how this is physically implemented, e.g. multiple cores on a die, multiple dies in a package, multiple packages in a system, etc. In this way, the term multi-core in this

work encompasses the (normally wider) terms of multi-chip CPUs and multi-CPU systems. For example, a cluster of 5 computers, each of which has 2 quad-core processors, is counted as a 40-core system in this respect. The recent Multiple/Many Integrated Cores (MIC) accelerator, and Single-Chip Cloud Computer (SCC) [126] hardware fall into the wider category of multi-core architectures, as defined hereinabove.

Multi-core architectures may have different connectivity and arbitration patterns (symmetric, asymmetric), memory arrangements (shared, distributed, shared-distributed), and may also implement SIMD instruction sets, but in any case they are expected to be able to execute multiple processes without time-sharing, i.e. fully parallel. Most of linear algebra routines can be naturally parallelized (e.g. matrix multiplication [127]) up to an Amdahl's limit. Another slightly different flavor of parallelization is to cast the algorithms into a block/tile/panel version with blocks shared between different computing cores [128], or the problem can be treated in a domain-decomposed way, with each of the sub-problems being handled to a core (e.g. linear system solving with the Schur decomposition [129], or multi-grid methods [130]).

The vast majority of high-grade linear algebra takes advantage of multi-core architectures and vectorization when possible. MIC-accelerated multi-core systems successfully rival more dedicated architectures (e.g. GPU-accelerated) in terms of performance [131].

In power systems

Power system particularities for the special case of (2.2a) & (2.2b) has been considered in [132, 133, 134]. Generally, the majority of applications in the power system domain exploit multi-core architectures for the linear algebra part. This is usually done through the inclusion of a multi-core enabled linear algebra library. Related work covers many power system analyses, e.g.

- power flow studies [135, 136, 137, 138, 139, 140, 141, 142],
- voltage stability studies [143],
- transient stability/electromechanical transients simulation [144, 137, 145, 146, 147, 148, 138],
- state estimation [149, 60, 140],
- electromagnetic transients simulation [146].

All of parallelized power system analyses assume the execution of the application on a platform with parallel capabilities, i.e. multi-core, often with SIMD support. In any case, multi-core platforms call for *parallel programming*. Software parallelization should always be in accord with hardware capabilities. A detailed presentation of the topic is out of the scope of this work.

2.3.3 Heterogeneous computing

Another class of specialized platforms are the ones that follow the heterogeneous computing paradigm. Heterogeneity refers to the use of processing units (cores) with dissimilar architecture and properties in an effort to get the best of all worlds. The workload is divided between different PUs based on their specialization. Usually, heterogeneous architectures are centered around one (or possibly more) GP PU(s) that handles generic, mostly serial, tasks. The GP PU(s) are interfaced with dedicated PU(s), often referred to as *coprocessors* or *accelerators*, which handle specialized tasks such as vector operations, intense floating point mathematics, signal processing, etc.

Over time, heterogeneous computing has gained significant acceptance for scientific computing [112], especially for linear algebra applications [111]. Connectivity, communication schemes and memory arrangements between the different PUs (processors and coprocessors) vary. The coprocessors that have been credited as having the biggest potential in linear algebra applications are MICs (see previous subsection), Graphical Processing Units (GPUs), Digital Signal Processors (DSPs), Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs), all briefed in next subsections.

2.3.4 GPU

GPUs are highly parallel vector processors with local memory for each of the processing cores. The total amount of memory is usually smaller than traditional multi-core CPUs (across different levels of cache) but the number of processing cores is normally larger [140]. They were initially targeted to graphics operations, but nowadays they are increasingly used as GP PUs, in what is known as General-Purpose Computing on Graphics Processing Units (GPGPU) arrangements [150]. Their characteristics make them ideal for a wide range of operations similar in computational nature [151].

The highly parallelized, wide-width SIMD datapaths of GPUs can tackle many of the points identified as problematic in section 2.2.2. Therefore a great wealth of GPU-enabled linear algebra work exists in literature, e.g. [152, 153]. Most of the time the integration of GPUs is done through linear algebra libraries that support GPU acceleration. Loop restructuring for the parallel operations (vectorizing, chaining, blocking, tiling) is critical in the process [107]. GPUs are often utilized en-masse and/or combined with multi-core GP PUs in hybrid architectures e.g. [154, 155]. FLOPS performance of modern GPUs is often one order of magnitude higher than GP CPUs. Given an optimized implementation, GPU speedups in the range of order(s) of magnitude are reported in most of the above - most conservative estimates of a speedup of 2.5x are given in [156]. One limitation of GPUs is that they have a higher computer horsepower to memory bandwidth ratio than GP (C)PUs, so this might lead to memory bottlenecks for GPU kernels [157]. Concerns have also been raised regarding their energy per operation efficiency [158] but relevant results in the Green500 list show that all 17 first spots are occupied by heterogeneous GPU-accelerated architectures (as of June 2014) [159].

In power systems

Power system applications that have GPU-accelerated linear algebra operations have been presented for many type of analyses:

- power flow [160, 161, 162, 163, 164, 165, 166, 167, 168],
- state estimation [169, 170],
- transient stability/electromechanical transients simulation [171, 172, 173, 174],
- electromagnetic transients simulation [175, 176, 177, 178, 179],
- optimization, e.g. Optimal Power Flow (OPF) [180], Unit Commitment [179].

In all of the cases, the heterogeneous computing concept is used: the CPU assumes main control of the application and the GPU is accelerating the burdensome parts.

2.3.5 FPGA

Field Programmable Gate Arrays (FPGAs) are integrated circuits that can be reconfigured after their manufacturing. This enables the designer of an FPGA platform to tailor it to the specificities of its application. FPGAs have recently seen penetration into the supercomputing domain, and this trend is expected to continue [181]. Because of the reconfigurability of the datapath and the high local distributed memory bandwidth, FPGAs have been deemed particularly suitable for linear algebra kernels, e.g. [182]. In most recent implementations, speedups of one order of magnitude are reported for linear system solving on conventional platforms [183].

A key enabler of FPGA utilization in linear algebra applications has been the addition of “hard” arithmetic cores, particularly for floating point operations [184], see also point a) in section 2.2.2. To target the vector/matrix nature of linear algebra operations, the dataflow arrangements in FPGAs include systolic [185]) and serial/parallel linear arrays [186]. The use of pipelining is also ubiquitous in the above.

In power systems

A lot of recent research has been focused in accelerating typical power system analyses through the use of FPGAs.

- power flow [187, 188, 189, 190, 191, 192, 193, 194]
- electromagnetic transients simulation [195, 196, 197, 198, 199, 200, 201, 202]
- optimization [203, 191]

As for GPUs, in most of the cases the FPGA is used as a dedicated accelerator running in conjunction with a conventional machine that handles the main flow of the application.

2.3.6 DSP

Digital Signal Processors (DSPs) are processing units optimized for the needs of digital signal processing. Digital signal processing operations are related to points b and c of section 2.2.2 and the datapath of DSPs is optimized for them. This high degree of specialization makes the instruction set of DSPs highly irregular, which renders them less suitable for general purpose computations. Streaming (direct memory access - DMA), pipelining (single-cycle instructions), SIMD with very-long-instruction-words (VLIW) techniques and hardware multiply-accumulate (MAC) units are a commonplace in DSP architectures.

The use of DSPs for numerical linear algebra has been readily investigated [204, 205, 206]. It normally uses a array of DSP processors [204] or a combination of DSPs with GP PUs in heterogeneous architectures [207]. Because of the exceptional power efficiency of DSP resources these heterogeneous architecture have achieved some impressive FLOPS/Joule figures [206]. Throughput is found to stabilize above a matrix size of approx 500-1500, and scaling with the number of DSP cores has been shown to be quasi-linear [206, 204]. In [205] a sparse matrix-vector multiplication is realized, similar to the one required by (2.2a). It is there highlighted the potential of DSPs to surpass GPUs in terms of power efficiency if manufactured using modern fabrication techniques (subsequently realized by TI® in the C667x series, as of March 2013).

2.3.7 ASIC and VLSI

The most dedicated platforms that can be constructed are Application Specific Integrated Circuits (ASICs) through Very Large Scale Integration (VLSI) Processes. ASICs are expected to deliver performance that greatly exceeds general purpose PUs, with a cost of significantly reduced flexibility/reconfigurability. For linear algebra, due to the vectorial nature of operations, proposed ASIC solutions have linear [208] or systolic array [209, 210, 211] arrangements.

For completeness sake, it should be mentioned that in-between general purpose and fully application-specific solutions (ASICs), there exist the Application-Specific Instruction Set Processors (ASIPs) [212]. These PU cores provide a tradeoff between flexibility of general-purposeness and dedication [213]. ASIP linear algebra implementations have a fixed instruction set part for GP computing and adapt their reconfigurable part to linear algebra needs, e.g. matrix singular value decomposition (SVD) in [213].

Conceptual designs for “power-system computers” based on custom VLSI ASICs have been proposed [214, 215, 216]. All of related work is conceptual and is based on systolic arrays, in an effort to tackle the nature of linear algebra computations. Nowadays, ASIC concepts can be proven by rapid-prototyping them on FPGAs.

2.3.8 Conventional computing and its limitations

All of the above platforms are based on conventional technology. The term conventional encompasses three main aspects. Conventional computers are

- *digital*, in that data is represented using discrete values (for a finer difference between digital and its contrary, analog, see [217]),
- *binary*, in that the discrete value of the data is encoded, stored and manipulated using their binary representation, i.e. “ones” and “zeros”, and
- *electronic*, in that the carrier of information is electricity, i.e. voltage, current and charge.

Digital binary electronic computers are manufactured using (very) large scale integration (VLSI) techniques on semi-conductor materials, mostly silicon, overlaid with various conducting (metal) or insulating layers. A set of circuits that implement a certain functionality is packed on the same physical piece, the chip, and the result is called an integrated circuit (IC). All of the platforms mentioned in the previous subsections, e.g. CPUs, GPUs, FPGAs, etc. come in the form of ICs. More complex designs are built on special boards, the Printed Circuit Boards (PCBs), with metallic connections that connect many ICs together.

The Moore's law defines that the trend in the increase of computing power in digital electronic computers is exponential. Recently fatigue in the trend has been observed [218]. The features of the semiconductors on the IC have been continuously scaled down, and now limitations are faced due to the (lack of) accuracy of the available lithographic technology. Another important factor is the increase of the frequency that the chips are clocked at. The two contribute to a variety of physical problems such as excessive heating of the chip, excessive power consumption (through leakage currents) [219], increased probability of errors due to manufacturing defects [220] and quantum phenomena appearing (e.g. quantum tunneling) that impede the proper functioning of the device [221].

2.3.9 Unconventional computing

Problems with conventional architectures has led to the emergence of unconventional computing paradigms. Unconventional can be understood as differentiating in any of the above three aspects that define conventional computing. Various such platforms have been proposed, such as optical computers (non-electronic), quantum computers (many different proposals), bio-computers (non-electronic) and wetware computers (non-digital and non-electronic), mechanical computers (non-electronic), analog electronic computers (non-digital), etc. All of the above try to gain advantage over conventional platforms in some metric, for example speed, size, energy consumption, heat generation, etc. Most of these technologies are either in their infancy (optical, quantum, bio) or have been deemed deprecated for scientific computation use (mechanical, analog electronic).

Depending on the properties of the underlying architecture some have proven better suited to linear algebra applications, such as optical computers [222], quantum computers [223, 224] and analog electronic computers. Implementations are usually experimental and of limited practicality, but serve well as concepts to get a better understanding of the technologies. Analog electronic computers are of particular interest to this work and will be analyzed in a separate section.

2.4 Analog electronic computers

In electrical computers data is represented using continuous electrical quantities (e.g. voltage, current, charge, etc.) The architecture of analog computers includes reconfigurable electronic devices such as potentiometers, operational amplifiers, controlled sources etc. Functionality desired from the computer can be obtained by setting the values of the reconfigurable elements as well as the connectivity pattern between them. Their scope of application includes simple arithmetic, the solution of differential equations, learning algorithms, neural network implementations, signal processing, linear algebra operations etc.

The main source of performance gain for analog computers is that the computing force behind them are the laws of physics [225]. The time that it takes to solve a problem on an analog computer depends only on the settling time of the latter. Additionally, this property is little affected by increasing the size of the problem [226]. From a theoretical point of view there is evidence that some analog configurations, such as neural nets, possess extraordinary computing properties [227, 228].

Apart from performance benefits, additional favorable characteristics of analog architectures have long been identified [229]. The ability to represent continuous real-world quantities with analogously continuous ones is a natural advantage [230]. Also, a continuous model of computation fits better the specificities of some kinds of problems such as differential equations [231]. Finally, power efficiency and related problems of digital architectures, is a major drive for a possible re-emergence of analog computing [232, 226].

Analog computers can be coupled with digital in arrangements that are called *hybrid* [233]. They are also called mixed-signal, in the sense that analog and digital signals coexist on the same hardware, and *discrete-continuous* because of the nature of the digital and analog signals. The transition between the digital and the analog domain is realized by arrays of analog-to-digital (ADC) and digital-to-analog converters (DAC).

The main reasons for implementing hybrid systems is to harness the benefits of both worlds, the computational prowess of the analog and the accuracy/precision of the digital. Often, the analog part is used as a seed (initial condition) for a digital part that takes over after to produce the final accurate solution [232]. A seed closer to the correct solution greatly enhances the speed of many iterative numerical algorithms. Various dedicated and general-purposes hybrid platforms have been proposed [234, 235, 232, 4].

Due to the pervasiveness of digital computing in our era, all modern hybrid platforms are “digital-computer-oriented”. The core platform is digital and the analog part is used as a black-box accelerator, dedicated to a specific operation. e.g. “math-coprocessor” in [236]. All analog signals are brought back to the digital domain, and usually no analog interface to the outside world is available.

2.4.1 In linear algebra

Some characteristics of analog platforms make them suitable for linear algebra applications. Points (b), (d), (e) of section 2.2.2 can be handled naturally in analog with meshed grid structures. This is because an analogy is drawn between the linear algebra operation and the physics that govern the behavior of such analog structures. More simply, analog grid computers are governed by linear algebra relations, exactly as they ones they are required to execute; therefore, the “mapping” is natural.

Naturally, the solution will suffer from the inaccuracy due to inaccuracies to the values of the analog components. From a mathematical point of view, this can be seen as epsilon perturbations to the perfect values of the operands (matrices and vectors).

The solution speed depends on the settling time of the device, which -under circumstances- can be significantly faster than that of digital solvers. This is so because laws of physics perform the “computations”. This settling time is owed to the non-instantaneous response of any physical circuit, due to parasitic and possibly non-parasitic capacitances in the circuit (and the resulting RC constants). Any additional auxiliary times, e.g. ADC and DAC conversions, should be added to this settling time. In total, in analog linear algebra, accuracy is traded for performance.

Exploratory studies on linear algebra using analog structures had been carried out in the 60’s and the 70’s [237, 238, 239, 240, 241, 242, 243] and the early 1990’s [244, 245, 246]. Recent relevant work has been based on Field Programmable Analog Arrays (FPAAs) [247, 248]. In [237, 238, 241] the authors dealt with the solution of linear systems that arise in the finite difference solution of partial differential equations, such as the diffusion equation. The resulting matrix has very specific fixed structure, therefore a fixed-connectivity resistor lattice is used. The linear algebraic properties of the connectivity matrix of the resistor network are elaborated in [240]. In [242] it is correctly identified that the use of such fixed topologies of purely passive components significantly narrows the range of matrices that can be handled. As a solution to the problem, the addition of active elements is proposed. This is similar to the use of negative resistor values in [237, 238].

References [237, 238, 240, 241, 242, 245] all use a hybrid (digital-analog) iterative scheme. They integrate the (finitely precise) analog solver into a digital iterative algorithm. In [242, 243, 245] the convergence properties of such a process are examined and [245] gives an insight on its timing. The benefit of this scheme is that analog imprecisions have no effect on the accuracy

Table 2.3 – Analogies between flow networks, power systems, and analog electronic networks

Flow network	Power systems	Analog electronics network
Node $v \in V$	Power system bus	Electrical node
Edge $e \in E$	Power system branch	Electrical branch
Flow	Current	Electrical current
Sources and sinks	Injectors (e.g. generators, loads, etc.)	Current sources
Edge capacity constraints	Branch/transformer ratings	Power ratings of the components of electrical branches

of the final solution.

In [246] an analog CMOS neural network implementation for the solution of (overdetermined) linear system is proposed. The interconnection scheme of [242] is used. Finally, [247, 248] uses the resources of an FPAA [249] to implement 2×2 dense vector-matrix multiplications.

2.4.2 In power systems computing

Analog platforms have a long history in power system applications [250]. This mainly comes from the fact that the two domains, power systems and analog electronics, feature topological similarities, as they are both typical examples of linear flow networks. A flow network consists of vertices V called *nodes* and edges E called *arcs*. The connectivity between edges and nodes is given by a directed graph $G = (V, E)$. Flow $f : V \times V \rightarrow \mathbb{R}/\mathbb{Z}$ is conveyed by edges, which have a given limited capacity $f(u, v) \leq c(u, v)$. On the vertices, *sinks* or *sources* may be present. The aggregate input and output flow to/from a given vertex must be equal, as per the *flow conservation* property. The analogies between a power system, and analog electronic network, and a generic flow network, are shown in table 2.3.

The analogies highlighted above have been exploited to create analog computers dedicated to power systems computations. Up until the 1960s transient network analyzers (TNA), also called AC network analyzers, were the established platforms for power system analysis purposes, e.g. power flow studies, short circuit analysis, transient stability studies, etc. They included scaled-down models of the elements of the real grid and operated with real analog voltage and current quantities.

The main disadvantages of TNAs were their cost (in terms of size, power consumption and money), lack of scalability (in terms of the limited size of systems that could be simulated) and lack of flexibility (in terms of models of the grid components that could be simulated). However they managed to provide real-time simulation of events at a point of time where computational power of digital computers was still insubstantial. This was because they implicitly performed linear algebra operations such as (2.2a) and (2.2b) using analog computing

(through their wirings), and hence they were using the ultra-fast laws of physics as computing force. Additionally, they were immune to issues arising from the numerical solution of the problem, e.g. numerical instability.

2.4.3 Evaluation criteria

The criteria on which analog electronic computers can be evaluated are the following.

Cost. This refers to the dollar cost of the platform. Generally this metric is rather unfavorable for analog computers, since the number of the required resources scales proportionally to the system that has to be analyzed - see also “scalability” hereunder.

Size. This refers to the physical size of the computer. This can be quite critical since an analog computer is comprised from actual physical electronic components. Especially in the past TNAs used to be quite large in size and an entire room used to be dedicated to the TNA.

Interfacing. This refers to the capabilities of the analog computer to be interfaced to other computing platforms. This is especially important nowadays since practically all modern computing infrastructure is digital.

Reconfigurability. This refers to the capabilities of the computer to be reconfigured, in the sense of performing different (mathematical) operations. Reconfiguration implies galvanically connecting the analog components of the computer. This wiring, was commonly done manually in older machines [251].

Scalability. Scalability refers to the ability of the computer to handle a larger amount of work. For power system problems this translates to larger power system sizes). In order to accommodate this growth the analog computer has to be enlarged with real more physical analog electronic components. Additionally larger problems, may affect the performance of the computer, as explained in the next paragraph.

Performance. The solution time of an analog computer greatly depends on the settling time of its electronics. The latter is affected by parasitic and non-parasitic elements (e.g. capacitances).

Accuracy and precision. The solution of an analog computer to a problem can be considered a variable X with a normal distribution $N(\mu, \sigma^2)$. Accuracy is understood as how close the solution for a problem computed in analog x is from the true one x^* . Precision is understood as how close repeated solutions in analog are to each other. Accuracy can be quantified by the mean value μ of X and precision by its standard deviation σ . The main cause of inaccuracy is the inherent inaccuracy of the electrical components that are used as well as losses of all kinds, e.g. leaked currents, parasitic impedances, etc. The main cause for imprecision is the Johnson-Nyquist (thermal) noise, which can often be approximated as white noise. The higher the Signal-to-Noise (SNR) ratio in the system, the higher its precision.

Power efficiency. Power efficiency can be understood as performance per watt. Joule losses account for the majority of the consumption in analog computers.

Functional completeness. Functional completeness refers to the range of computations that can be performed by the analog computer. By definition analog components are not universal as digital resources are in the sense of a Turing machine, i.e. analog components are bound to executing the specific operation they were (a priori) manufactured for. This can significantly hamper the modeling/functional capabilities of the analog computer.

2.5 Outlook

Despite their natural dedicated-ness to power system problems, analog TNAs were abandoned in favor of digital computers running numerical routines, once digital technology of adequate performance was widely available. This was because digital technology was deemed generally superior in almost every aspect. TNAs remained a bit longer in use for education [252], for the simulation of power electronic equipment (e.g. HVDC, SVC) [253] and real-time studies (e.g. HIL) [254], but they were later superseded by Real Time Digital Simulators (RTDSs) in that domain.

An important paradigm that was learned from TNAs was that the calculations for the power grid (the ensemble of interconnecting elements, such as transmission lines and transformers), and the calculations for the components connected to the grid (generators, loads, etc.) were carried out in two connected, but physically different machines [255].

- The grid calculations involved a big linear algebra part (with the admittance matrix) that reflects the interdependence of voltages and currents in the system. This part was handled in a (miniaturized) DC or AC model of the real-world grid.
- The calculations for the bus components were usually differential-algebraic in nature and reflected their dynamic behavior. This part was handled by miniature models of the real-world elements.

Analog computing was brought back to the fore by the work of Fried[256, 257]. The major drive for that was the demand for higher performance highlighted in chapter 1. Great advances in analog electronic technology (especially manufacturing) offered a potential to solve the deficiencies of analog computers in the criteria of section 2.4.3.

In [257], five different approaches with different levels of abstractions and analog computation were proposed. The one that was the most promising is based on the simulation of phasors and uses a real decomposition of the (complex) nodal equation. This gave birth to subsequent related research, centered around three universities:

- the Arizona State University [258, 259, 260],

- the Drexel University [191, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273] and
- the École Polytechnique Fédérale de Lausanne (the home university of the original research of Fried) [256, 257, 274, 275, 276, 277, 278, 279, 280, 281, 4, 282, 283, 284, 285, 286, 287].

This work is a continuation of investigations in the field of the development hybrid (analog and digital) computers dedicated to power system applications. In chapter 3 a related realized prototype will be presented. The prototype will be evaluated based on the criteria of section 2.4.3 and will be compared to related state of the art. Advantages and shortcomings will be underlined and a future conceptual design will be proposed in chapter 4 to overcome the drawbacks of the current design.

3 Realized dedicated mixed signal solver

Chapter 3 presents a hardware platform that is dedicated to the transient simulation of power systems. The platform, termed Field Programmable Power Network System (FPPNS), has an analog and a digital part. Analog computing is used for the matrix solution part, while pipelined digital resources on a field programmable gate array (FPGA) handle all non-linear-algebraic computations.

The FPPNS is connected to a host PC as a USB peripheral. On the PC side a power system analysis suite has been developed, termed *elab-tsaot*. Through a set of OS drivers the dedicated platform is recognized as a hardware accelerator to the computations of the analysis suite. The entire software design and functionality is presented in the chapter.

The mathematical properties of the FPPNS are thoroughly examined. The inaccuracies of the analog hardware are examined and their effect on the final results provided by the FPPNS is shown. The system is evaluated across the nine criteria of section 2.4.3, and a comparison with related state-of the art is conducted.

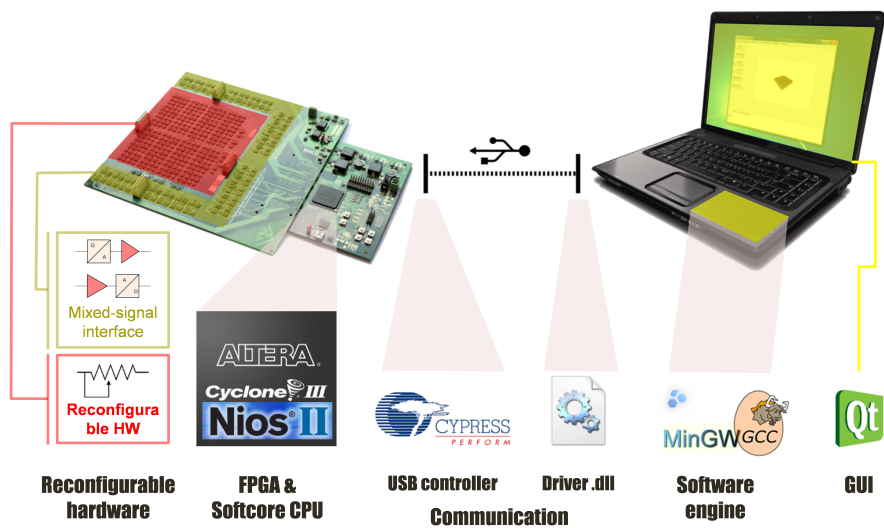


Figure 3.1 – Overview of the multi-platform system

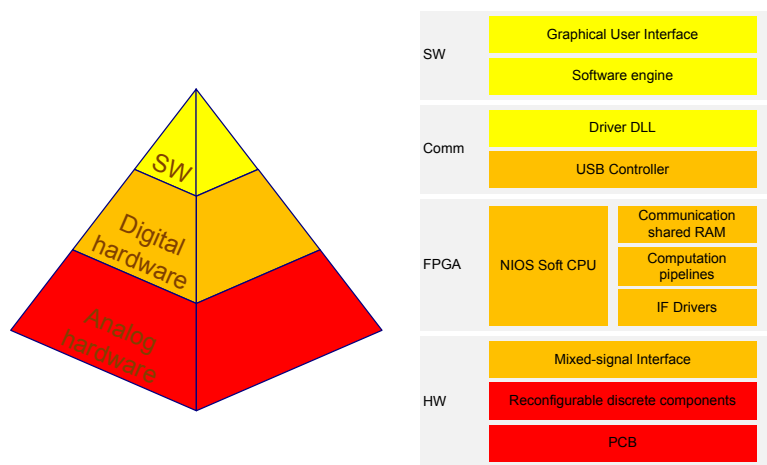


Figure 3.2 – Levels of the multi-platform system

Fig. 3.1 shows a schematic overview of the multi-platform system. On the left there is the dedicated hardware, on the right the software, and in between the USB connection of the two. A color convention has been adopted to denote the different levels of the whole system, shown in Fig. 3.2. A photo of the dedicated hardware connected to a real PC is shown in Fig. 3.3. The hardware part of the system will be referred to as the *Field Programmable Power Network System (FPPNS)* for the rest of this work and the software part as *elab-tsaot*.

3.1 Hardware

The FPPNS consists of two parts, an analog part and a digital one. Analog computing is used for the matrix solution part, while pipelined digital resources on a field programmable gate array

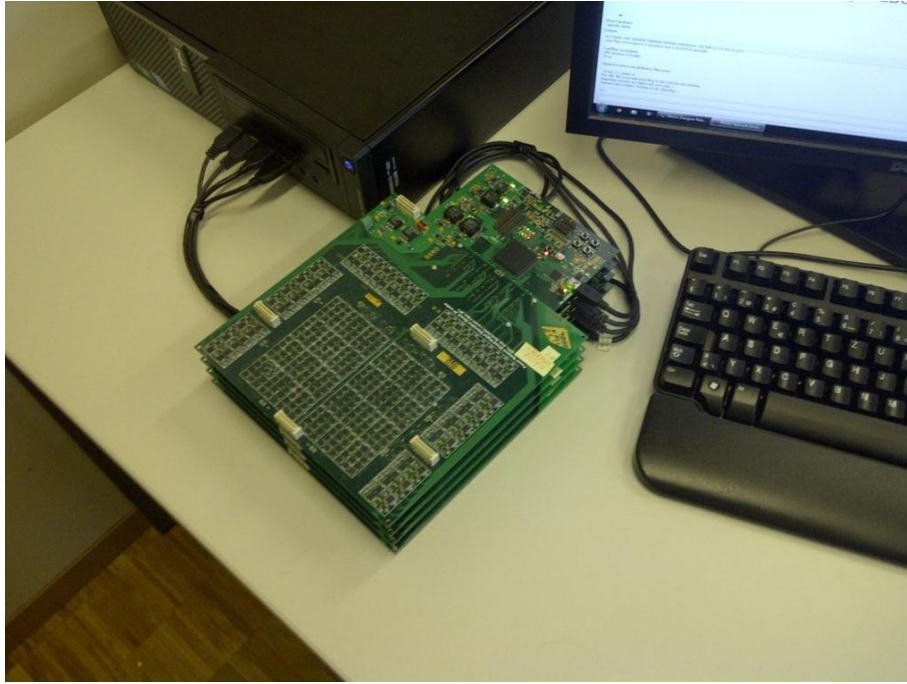


Figure 3.3 – Photo of the multi-platform system

(FPGA) handle all non-linear-algebraic computations. The analog and the digital domains are interfaced using analog-to-digital (ADC) and digital-to-analog (DAC) converters. An overview of the proposed mixed-signal platform can be seen on Fig. 3.4.

The system is targeted to the transient simulation of power system. In particular it handles the transient simulation of *balanced phasors* of electrical quantities for *symmetrical three-phase networks* in the *per-unit* system. All related modeling assumptions and theory concerning the above can be found in appendix B.

The general formulation of the problem to solve is the one of (2.1). FPPNS uses a partitioned (and not simultaneous) approach to solve (2.1) [288, 283]. Equations are separated in grid and injection equations. At each time step of the integration, the nodal flow algebraic equation of the grid is solved and then the dynamic behavior of the injections is determined, as per Fig. 3.5.

The nodal voltage equation of the figure is included in f_a set of (2.1). The injections equations consist of the differential f_i and the algebraic equations g_i for each injection i . Functions f_i and g_i are included in the f_d and f_a sets of (2.1) respectively. I is the vector of currents injected at each bus and V is the vector of bus voltages. Both I and V are included in the set of algebraic variables x_a of (2.1). One current injection I_i and one voltage V_i is associated with each injector. The latter has a set of differential variables x_i and algebraic variables (V_i, I_i, y_i) . The former are included in x_d and the latter in x_a of (2.1). The set of DAEs for each injector is parameterized by a set of parameters λ_i - more on this in section 3.1.2.

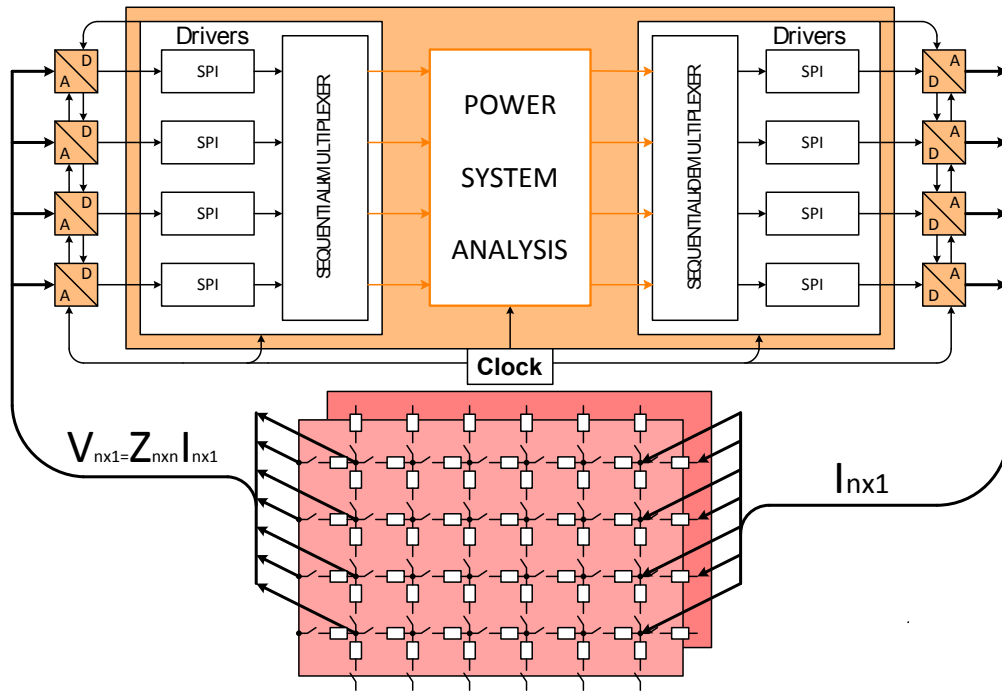


Figure 3.4 – Overview of the existing mixed-signal computer [4].

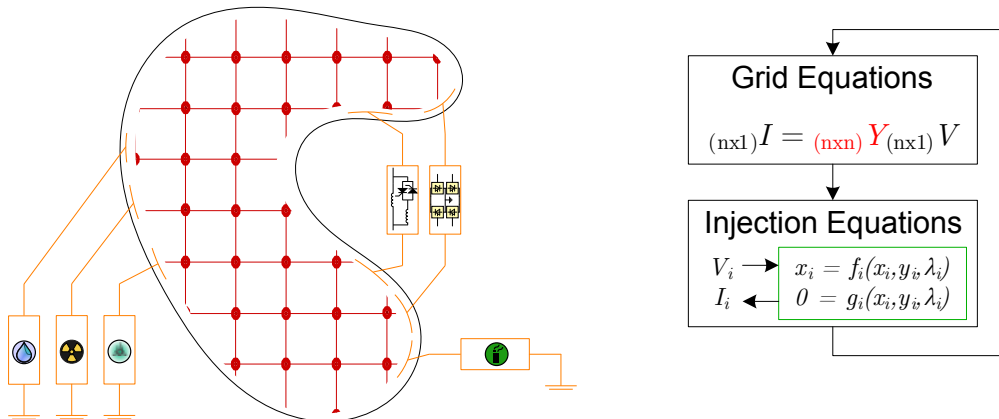


Figure 3.5 – Partitioned solution scheme for power system simulation equations

3.1.1 Analog part

The analog part of the existing platform contains the wealth of reconfigurable electronics that perform the hyper-parallelized analog computation of the linear system (nodal equations), which arises in a power system.

Derivation of the nodal equations

A starting point for the investigation is to consider branches as complex two-port networks. In our case, One port is denoted as the “from” (related quantities subscripted with the letter f) port of the network and another the “to” port (related quantities subscripted with the letter t). Each port is a complex interface for the electrical quantities, current and voltage. The convention is that the currents are entering the two-port network on both sides. For the two port network above the (complex) *admittance parameters* are defined as follows.

$$\begin{bmatrix} i_f \\ i_t \end{bmatrix} = [y] \cdot \begin{bmatrix} v_f \\ v_t \end{bmatrix}, \quad [y] = \begin{bmatrix} y_{ff} & y_{ft} \\ y_{tf} & y_{tt} \end{bmatrix} \quad (3.1)$$

Every interconnecting element of the power system grid can be modeled as a two-port network such as the above. Three-port elements such as three-winding transformers can be considered as a combination of three normal two-winding transformers modeled with their π -equivalent, and one fictional bus [36].

Branch models can have various degrees of complexity depending on the time scale of phenomena of interest. It is very common for studies in the transient time scale to neglect the fast transient (electromagnetic) behavior of the interconnecting elements, e.g. transformers and lines.

In this work branches are modelled with the generalized π -model of Fig. 3.6, where all quantities refer to the phasors of the positive sequence in the per unit system. This model adequately describes the behavior of *short* transmission lines ($l \ll \lambda$), where short refers to the length of the line with respect to the wavelength $\lambda = c/f_n$ of the carrier frequency f_n of the system [36]. In the general case the admittance parameters of the two-port network are as follows.

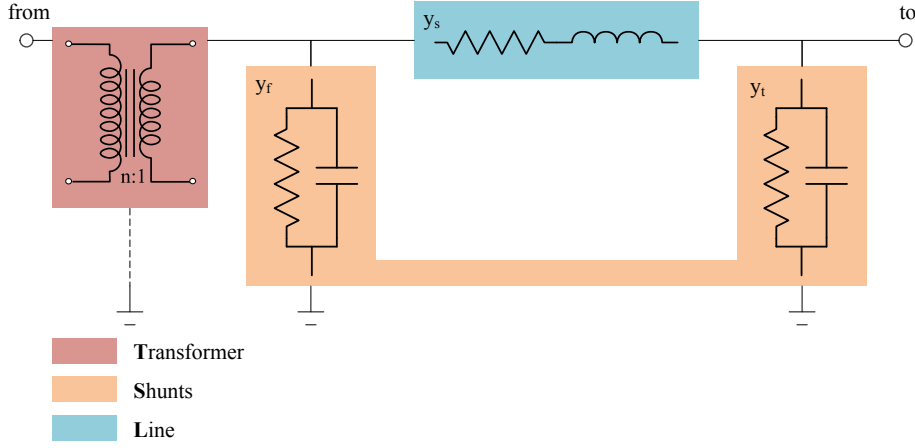


Figure 3.6 – Generalized π model of a generic power system branch

$$y_{ff} = \frac{i_f}{v_f} \Big|_{v_t=0} = \frac{y_s + y_f}{x^2} \quad (3.2a)$$

$$y_{ft} = \frac{i_f}{v_t} \Big|_{v_f=0} = -\frac{y_s}{x \cdot e^{-j \cdot \theta}} \quad (3.2b)$$

$$y_{tf} = \frac{i_t}{v_f} \Big|_{v_t=0} = -\frac{y_s}{x \cdot e^{j \cdot \theta}} \quad (3.2c)$$

$$y_{tt} = \frac{i_t}{v_t} \Big|_{v_f=0} = y_s + y_t \quad (3.2d)$$

Where

- $n = x \cdot e^{j \cdot \theta} [\emptyset]$ is the ratio of the ideal transformer,
- $y_s = g_s + j \cdot b_s = 1/(r_s + j \cdot x_s) [pu] \rightarrow \begin{cases} g_s = \frac{r_s}{r_s^2 + x_s^2} \\ b_s = -\frac{x_s}{r_s^2 + x_s^2} \end{cases}$ is the transversal line admittance,
- $y_f = g_f + j \cdot b_f [pu]$ is the shunt admittance at the from side, and
- $y_t = g_t + j \cdot b_t [pu]$ is the shunt admittance at the to side.

The degrees of freedom in the model are the following 8 real parameters.

- $x [\emptyset]$ is the tap setting of the transformer, and it quantifies the magnitude scaling of quantities on the secondary winding side.
- $\theta [rad]$ is the phase shift introduced by the transformer to quantities on the secondary winding side.

- g_f [pu] is the shunt conductance at the from side.
- b_f [pu] is the shunt susceptance at the from side.
- g_t [pu] is the shunt conductance at the to side.
- b_t [pu] is the shunt susceptance at the to side.
- r_s [pu] is the series resistance of the line.
- x_s [pu] is the series reactance of the line.

In the above the pu units refer to the corresponding physical unit translated in the pu system, e.g. Ohms for impedances, etc.

Apart from branches, that connect two different nodes, it is also very common to have elements that are shunt to buses, i.e. that connect a bus to ground. For these, an one-port network representation is used. Its admittance parameters of this network simply involve the voltage of the node and the current that is injected through the shunt element to ground.

$$i_n = y_{sh} \cdot v_n \quad (3.3)$$

A power system topology is usually provided as a list of all its constituent elements, branches and shunt elements. Branches are provided in lists having values for all the parameters of the models that are involved. Connectivity is usually provided in the form of edge lists, e.g. branch k connecting from-bus f to to-bus t . Let $N = (U, E)$ be the graph of the power system with a bus set $|U| = n$ and a branch set $|E| = m$. Then the to- and the from- incidence matrices of the graph of N can be defined.

$${}_{m \times n} Cf \text{ with } cf_{ji} = \begin{cases} 1, & \text{if branch } j \text{ is from-incident to bus } i \\ 0, & \text{if branch } j \text{ is not from-incident to bus } i \end{cases} \quad (3.4)$$

$${}_{m \times n} Cg \text{ with } ct_{ji} = \begin{cases} 1, & \text{if branch } j \text{ is to-incident to bus } i \\ 0, & \text{if branch } j \text{ is not to-incident to bus } i \end{cases} \quad (3.5)$$

The complex two-port admittance parameters of (3.2) can be collected in four $(m \times 1)$ vectors Y_{ff} , Y_{ft} , Y_{tf} and Y_{tt} and then the contribution of branches to the overall admittance matrix can be formalized as follows.

$${}_{(n \times n)} Y_{br} = C_f^T \cdot \text{diag}(Y_{ff}) \cdot C_f + C_f^T \cdot \text{diag}(Y_{ft}) \cdot C_t + C_t^T \cdot \text{diag}(Y_{tf}) \cdot C_f + C_t^T \cdot \text{diag}(Y_{tt}) \cdot C_t \quad (3.6)$$

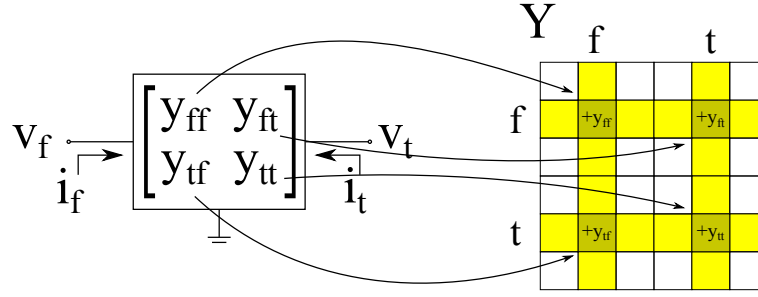


Figure 3.7 – The complex two-port network for a branch connecting buses f and t and the effect it has on the building of the Y matrix

Similarly if there are m_s shunt elements in the system the shunt incidence matrix can be defined.

$$m_s \times n \text{ } Cs \text{ with } cs_{ji} = \begin{cases} 1, & \text{if shunt element } j \text{ is incident to bus } i \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

The complex admittance parameters can be collected to a $(m_s \times 1)$ vector Y_{sh} and the contribution of shunt elements to the overall admittance matrix can be formalized as follows.

$$(n \times n) Y_{sh} = Cs^T \cdot \text{diag}(Y_{sh}) \cdot Cs \quad (3.8)$$

The final complete admittance matrix of the grid is the addition of the effects of branches and bus shunt elements.

$$Y = Y_{br} + Y_{sh} \quad (3.9)$$

The effect that branch k has on the building of the Y matrix according to (3.6) is shown in Fig. 3.7, and the effect of shunt elements according to (3.8) is shown in Fig. 3.8.

For a power system of n buses, the $n \times n$ complex nodal equations that link nodal current injections I to bus voltages V can be decomposed in real parts as follows.

$$I = Y \cdot V \rightarrow \begin{cases} \text{Re}\{I\} & = G \cdot \text{Re}\{V\} - B \cdot \text{Im}\{V\} \\ \text{Im}\{I\} & = G \cdot \text{Im}\{V\} + B \cdot \text{Re}\{V\} \end{cases} \quad (3.10)$$

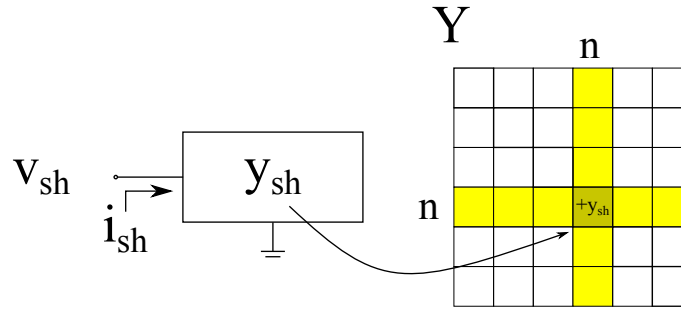


Figure 3.8 – The complex one-port network for a shunt element on bus s and the effect it has on the building of the Y matrix

Where $Y = G + j \cdot B$ is the admittance matrix of the topology. In high-voltage transmission lines the assumption that $x_s \gg r_s$ is often made (Ch. 5 in [289]). In the extreme case $r_s/x_s \rightarrow 0$. Substituting this to transversal line admittance parameters of branches yields the following.

$$g_s = \frac{r_s}{r_s^2 + x_s^2} = \frac{(r_s/x_s)}{x_s \cdot (r_s/x_s)^2 + x_s} \xrightarrow{r_s/x_s \rightarrow 0} 0 \quad (3.11a)$$

$$b_s = -\frac{x_s}{r_s^2 + x_s^2} = -\frac{1}{x_s} \cdot \frac{1}{(r_s/x_s)^2 + 1} \xrightarrow{r_s/x_s \rightarrow 0} -\frac{1}{x_s} \quad (3.11b)$$

An additional simplifying assumption is that no transformers exist in the model, i.e. $x = 1$ and $\theta = 0$ for all branches. Lastly, the conductive part of all shunts (shunt admittances at the *from* and *to* side of branches as well as proper shunt elements) is neglected, i.e. $g_f = 0$ and $g_t = 0$ for all branches, and $g_{sh} = 0$ for all shunt elements. This is a reasonable assumption, since shunt conductance means a galvanic connection between a point in the electric grid and the ground, i.e. a permanent “fault”. With the simplifying assumptions above, $G_{ps} = 0$ and two separate $n \times n$ linear systems are required to model the relations between I and V .

$$\begin{bmatrix} -Im\{I\} \\ Re\{I\} \end{bmatrix} = \begin{bmatrix} -B & 0 \\ 0 & -B \end{bmatrix} \cdot \begin{bmatrix} Re\{V\} \\ Im\{V\} \end{bmatrix} \quad (3.12)$$

The sign of the first subset of equations is changed so that the resulting $-B$ has positive diagonal entries and negative off-diagonal entries. This is explained hereunder.

For the transversal reactance of (3.11) it holds.

$$x_s = x_L - x_c = \omega \cdot L - \frac{1}{\omega \cdot C} \quad (3.13)$$

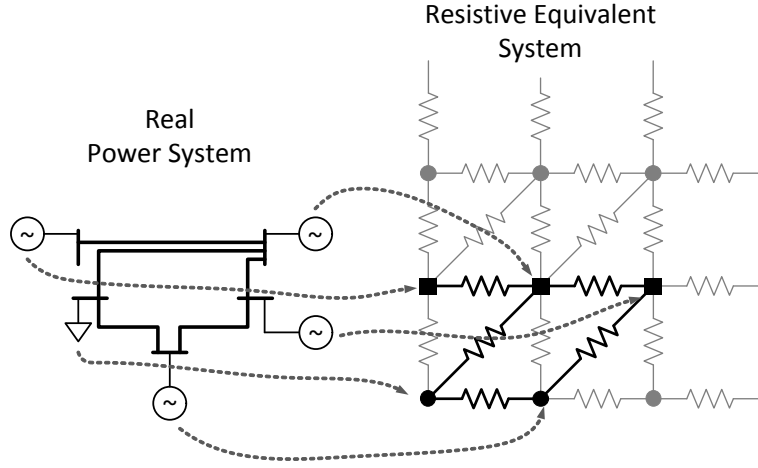


Figure 3.9 – Power system topological mapping into an electronic resistor network equivalent.

Where x_L is the inductive reactance of the line, x_C is the capacitive reactance, L is the inductance, C is the capacitance of the line and ω is the angular frequency. For normal power system transmission lines the reactance is inductive, hence the susceptance is negative¹.

$$x_L > x_C \xrightarrow{(3.13)} x_s > 0 \Rightarrow b_s > 0 \quad (3.14)$$

Hence by inverting the sign of B , its diagonal elements are negative, and the off-diagonal elements are positive.

Creation of the analog equivalent

Each power system branch is associated with one electrical branch, and a resistor network is created that features topological similarity with the B matrix. The process is called *topological mapping* and is illustrated in fig. 3.9.

An electrical branch of the existing prototype is featured in Fig. 3.10. Similar to power system branches, there is a “from” and a “to” side, denoted with an f and a t in the figure. The effective resistance of the branch is handled by two digitally reprogrammable potentiometers, pot_f and pot_t . Both potentiometers have a resolution of 8 bits and a range of 10kOhms. Their nominal final resistance value is given by

$$r_{pot} = \frac{t}{2^M} \cdot R. \quad (3.15)$$

¹This might not be the case for lines where series capacitive compensation is present, but this (exceptional) case is not considered in this work.

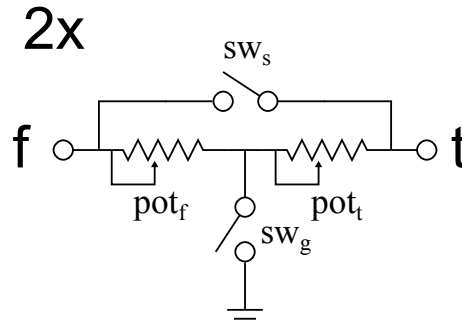


Figure 3.10 – Schematic of an electrical branch of the existing FPPNS

Where $t \in [0, 2^M)$ is the digital tap setting, $M = 8$ is the bit resolution and $R = 10 \text{ k}\Omega$ is the full-scale resistance.

The design also includes a transversal short-circuiting switch sw_s and a switch that can be used to emulate a galvanic connection in the middle of the line. The exact point where the faulty connection is determined by the ratio of the resistances of pot_f and pot_t . In the following equation loc is the emulated percentage of line length measuring from the “from” side, e.g. $loc = 0\% \equiv$ fault at f -node.

$$loc = \frac{pot_f}{pot_f + pot_t} \% \quad (3.16)$$

Notice that there are two duplicate networks of electrical branches (as shown in Fig. 3.4). All connections and values (pot_f , pot_t , etc.) are the same for corresponding branches in the two networks. This is necessary since two equivalents of B are required in (3.10). Sometimes the first network is called “real” since it models the relation between the imaginary current and the real voltage and the second is called “imaginary” since it models the relation between the real current and the imaginary voltage in (3.12).

In the current implementation a bank of electrical branches exist, the connectivity pattern of which can be modified at will (with some limitations) by a set of electronic programmable switches. The topology of the available electrical branch resources is shown in Fig. 3.11. The constrained topology of the electrical branch bank poses limitations to the nature of the topologies that can be mapped on the FPPNS.

Each slice of the FPPNS has a capacity of 24 nodes. Each of the resistors in the figure schematically depicts an electrical branch as the one of Fig. 3.10. On the end of each electrical branch there are digitally controlled electrical switches that can disconnect the branch. The implementation is based on discrete electronics on a custom-made PCB.

The PCB has analog extension connections that allows it to be vertically connected as shown

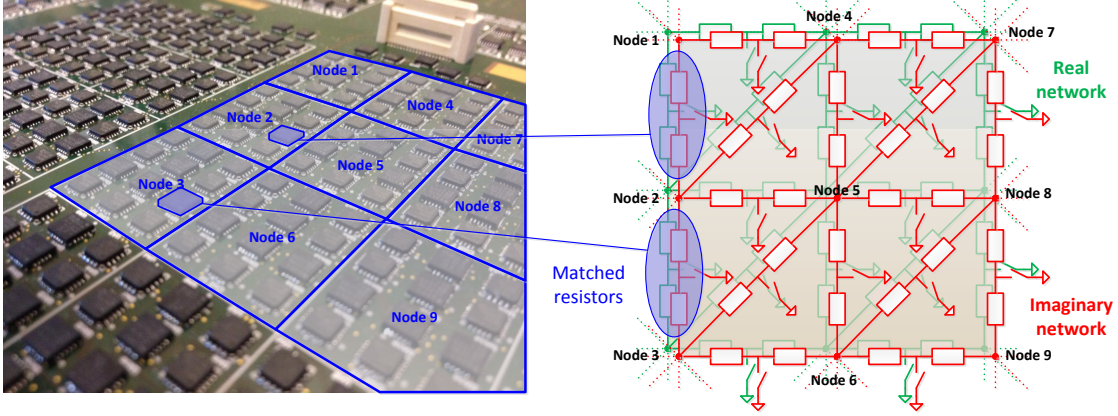


Figure 3.11 – Detail of the implementation of a slice of the FPPNS using discrete electronics

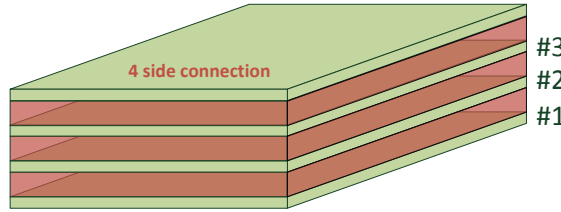


Figure 3.12 – A stack of four FPPNS slices

in Fig. 3.12. Four FPPNS PCBs have been manufactured, yielding a total capacity of 96 (4×24) nodes.

Once the topological mapping of the branches is decided, the mapping of power system buses to electrical nodes is also automatically induced. In every electrical node, currents and/or voltages can be written and read from the resistor network using dedicated circuitry. In the realized prototype, current injection, voltage injection and voltage reading is supported for each node. The synoptical electronic interface of a node of the resistor network in the current implementation is shown in Fig. 3.13. All electrical nodes possess individual ADC/DAC units, therefore all injections and measurements can be done fully in parallel.

In the current implementation a 12-bit DAC has been used. When the DAC is operated in voltage injection mode its output is as follows.

$$V_{DAC} = \frac{t}{2^M} \cdot V_{FS} - V_{off} \quad (3.17)$$

Where $t \in [0, 2^M)$ is the digital tap setting, $M = 12$ is the bit resolution and $V_{FS} = 5V$ is the full-scale voltage of the DAC. $V_{off} = 2.5V$ is an offset voltage that allows the final output of the DAC to assume both positive and negative values ($-2.5, +2.5$) V.

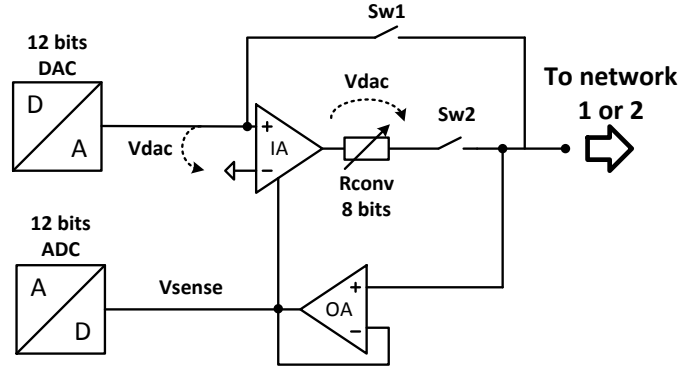


Figure 3.13 – Synoptical schematic of a node of the FPPNS

When the DAC is operated in current injection mode, the instrumentation amplifier (shown schematically as IA in the figure) is connected so that a V_{DAC} voltage drop is applied across a conversion resistor R_{conv} . The resulting current output is

$$I_{DAC} = \frac{V_{DAC}}{R_{conv}}. \quad (3.18)$$

The latter is a function of the voltage DAC tap setting t of (3.17). The exact setting of R_{conv} depends on the desired full-scale output current $(V_{FS} - V_{off}) / R_{conv}$.

Voltage measurement functionality is provided through the A/D block of Fig. 3.13. The following model describes its functioning.

$$D^{\mathbb{R}} = \frac{V_{in}}{V_{FS}} \cdot 2^M \quad (3.19a)$$

$$D = \begin{cases} 0 & \text{if } D^{\mathbb{R}} \leq 0 \\ \lfloor D^{\mathbb{R}} + 0.5 \rfloor & \text{otherwise} \\ 2^M - 1 & \text{if } D^{\mathbb{R}} \geq 2^M - 1 \end{cases} \quad (3.19b)$$

$$V_{ADC} = \frac{D \cdot V_{FS}}{2^M} \quad (3.19c)$$

The analog input to the ADC is V_{in} and the output is the digital word D of length M . $D^{\mathbb{R}}$ would be the output word if it was allowed to assume non-integer values - it is an intermediate result in (3.19a) that facilitates the analysis. $D^{\mathbb{R}}$ is quantized and saturated to D in (3.19b). V_{FS} determines the range of the ADC, since it is the full scale quantity that can be accepted at its input $V_{in} \in [0, V_{FS})$. Finally, V_{ADC} is the value that is actually “read” by the ADC, and that is communicated to digital parts of the system.

Chapter 3. Realized dedicated mixed signal solver

Once the topological mapping is complete, the *value mapping* process takes place. Admittance, current and voltage values are scaled from one domain to the other via multiplicative mapping ratios ρ_Y , ρ_I and ρ_V with units of Ω/pu , A/pu and V/pu respectively.

$$g_{el} = \rho_Y \cdot (-b_{ps}) \quad (3.20a)$$

$$i_{el} = \rho_I \cdot i_{ps} \quad (3.20b)$$

$$v_{el} = \rho_V \cdot v_{ps} \quad (3.20c)$$

In the above, the *el* and the *ps* subscripts are added for clarity purposes. Due to Ohm's law, the three ratios are related and only two degrees of freedom are available for the three ratios.

$$\rho_I = \rho_Y \cdot \rho_V \quad (3.21)$$

The mapping ratio for impedances is simply the inverse of the admittance ratio.

$$\rho_Z = \frac{1}{\rho_Y} \quad (3.22)$$

The result after the topological and value mappings is an electronic system that is equivalent to the power system topology.

$$I_{el} = G_{el} \cdot V_{el} \quad (3.23)$$

The current injection vector I_{el} contains elements as per (3.18), “known” voltage values in V_{el} are injected into the grid by the mechanism described in (3.17), and “unknown” voltage values in V_{el} are measured from the grid as per (3.19).

Mathematical operations

Once the analog of (3.23) has been created then the resistor network of the FPPNS can be seen as a causal system. For a grid of n electrical nodes, there are n node voltages and n current injections to them. For a node, either a voltage or a current injection can be performed. Nodes with voltage injections have unknown current injections, and vice versa nodes with current injections have unknown voltages. Unknown electrical quantities are induced by the physics of the system.

When only voltages are injected, V_{el} is known in (3.23), I_{el} is the unknown vector and the operation performed by the FPPNS corresponds to a *matrix-vector multiplication*. When only current are injected I_{el} is known (rhs vector), V_{el} is the unknown vector, and the operation performed corresponds to a *linear system solving*.

An alternative operation is to inject currents to some of the nodes, and voltages to some others. Without loss of generality, let \mathcal{N} be the set of all electrical nodes with a cardinality of $|\mathcal{N}| = n$, let $\mathcal{J}_n = \{1, \dots, n\}$ be the set of bus indexes. Let the non-empty bus (index) set of buses for which current is injected be $\mathcal{J}_I \subseteq \mathcal{J}_n$. Then the complementary bus (index) set of buses for which voltage is injected is $\mathcal{J}_V = \mathcal{J}_n \setminus \mathcal{J}_I$. Borrowing the notation from [290], for a matrix A and any two index sets \mathcal{J}_{s1} and \mathcal{J}_{s2} , let $A[\mathcal{J}_{s1}, \mathcal{J}_{s2}]$ denote the submatrix of A obtained by the rows indexed by \mathcal{J}_{s1} and by the columns indexed by \mathcal{J}_{s2} . Similarly, for an one-dimensional vector u , let $u[\mathcal{J}_{s1}]$ denote the subvector of u obtained by the elements indexed by \mathcal{J}_{s1} . Then mixed current-voltage injection can be described in matrix notation as follows.

$$\begin{bmatrix} I_{el}[\mathcal{J}_I] \\ I_{el}[\mathcal{J}_V] \end{bmatrix} = \begin{bmatrix} G_{el}[\mathcal{J}_I, \mathcal{J}_I] & G_{el}[\mathcal{J}_I, \mathcal{J}_V] \\ G_{el}[\mathcal{J}_V, \mathcal{J}_I] & G_{el}[\mathcal{J}_V, \mathcal{J}_V] \end{bmatrix} \cdot \begin{bmatrix} V_{el}[\mathcal{J}_I] \\ V_{el}[\mathcal{J}_V] \end{bmatrix} \quad (3.24)$$

Where $I_{el}[\mathcal{J}_I]$ and $V_{el}[\mathcal{J}_V]$ are the unknowns. The above can be reformulated as a linear system problem.

$$\begin{bmatrix} I_{el}[\mathcal{J}_I] \\ V_{el}[\mathcal{J}_V] \end{bmatrix} = \begin{bmatrix} G_{el}/G_{el}[\mathcal{J}_V, \mathcal{J}_V] & G_{el}[\mathcal{J}_I, \mathcal{J}_V] \cdot G_{el}[\mathcal{J}_V, \mathcal{J}_V]^{-1} \\ -G_{el}[\mathcal{J}_V, \mathcal{J}_V]^{-1} \cdot G_{el}[\mathcal{J}_V, \mathcal{J}_I] & G_{el}[\mathcal{J}_V, \mathcal{J}_V]^{-1} \end{bmatrix} \cdot \begin{bmatrix} V_{el}[\mathcal{J}_I] \\ I_{el}[\mathcal{J}_V] \end{bmatrix} \quad (3.25)$$

Where $G_{el}/G_{el}[\mathcal{J}_V, \mathcal{J}_V]$ is the Schur complement of $G_{el}[\mathcal{J}_V, \mathcal{J}_V]$ of G_{el} as follows.

$$G_{el}/G_{el}[\mathcal{J}_V, \mathcal{J}_V] = G_{el}[\mathcal{J}_I, \mathcal{J}_I] - G_{el}[\mathcal{J}_I, \mathcal{J}_V] \cdot G_{el}[\mathcal{J}_V, \mathcal{J}_V]^{-1} \cdot G_{el}[\mathcal{J}_V, \mathcal{J}_I] \quad (3.26)$$

3.1.2 Digital part

The digital part of the platform is realized on reconfigurable hardware. An Altera Cyclone III FPGA has been used, the configuration of which is shown on Fig. 3.14. The Avalon interconnection bus is the backbone of the synthesized system. A softcore NIOS II CPU has been instantiated to act as an orchestrator to the functioning of the computation pipelines as well as to perform all auxiliary functions, i.e. initialization, communication with a PC, etc.

The main purpose of the digital part is to solve the equations that govern the behavior of the elements connected to the grid, generators, loads, etc. This is in sharp contrast to other com-

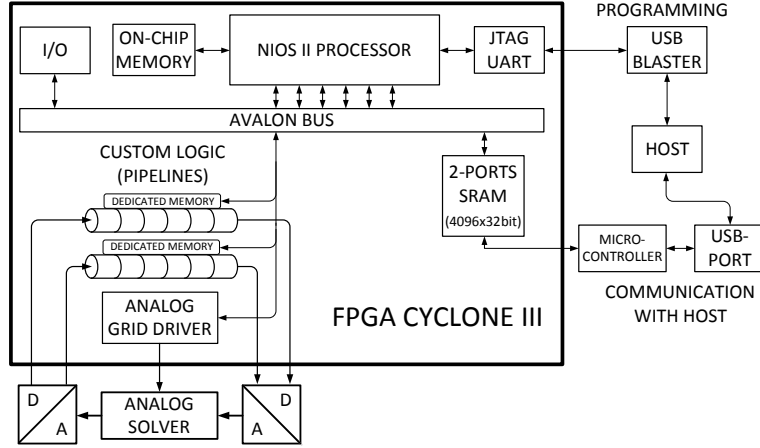


Figure 3.14 – Configuration of the digital part of the existing computer prototype (Altera Cyclone III FPGA).

parable work that uses analog models for the components connected to the grid (generators, loads, etc.) [273]. In our design all related computations are done in the digital domain, and hence practically unlimited flexibility is offered.

At any time, the behavior of any power system element can be fully defined as a set of DAEs, that depend on the complex voltage and complex current at the bus where the element is connected to, as shown in Fig. 3.5.

$$S_i = \begin{cases} \dot{x}_i &= f_i(x_i, y_i, \lambda_i) \\ 0 &= g_i(x_i, y_i, \lambda_i) \end{cases} \quad (3.27)$$

Any other internal algebraic and dynamic variable that might concern injection i is decoupled from the solution of other injections hence this part of the computations is amenable to heavy parallelization. In the current implementation, a tradeoff between computational optimization, flexibility, and resource usage is achieved by adopting a pipelined computation scheme [4].

The set of injections E is partitioned in K equivalence classes E_k of structurally similar equations, i.e. equations that differ only in a set of parameters. Parallelization along the above partitioning scheme is suggested: One pipelined computational module C_k exists per equivalence class E_k that processes injections of the class. In the pipelines the numerical integration of the DAE sets of each injector are performed. At each pipeline clock cycle, the parameters of the equations to be solved on C_k are iterated across injections that belong to the class E_k .

In the current implementation two explicit algorithms have been used, the Forward Euler (FE) (1st order) and the 2-step Adams-Bashforth (2nd order) method (AB2). FE is a single step

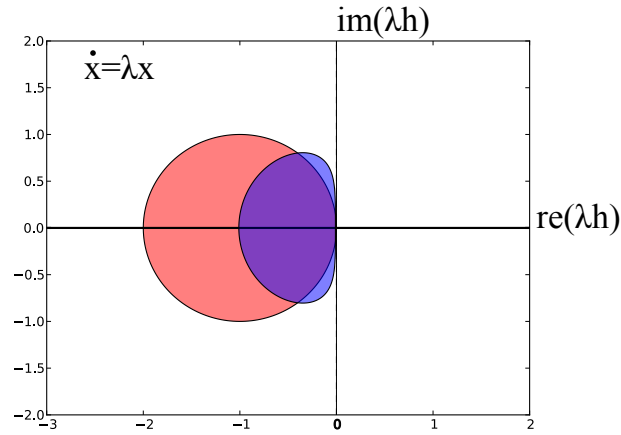


Figure 3.15 – Stability region of the Forward Euler method and the 2-step Adams-Bashforth method

method, and AB2 is a multistep method.

$$\text{FE: } x_{k+1} = x_k + h \cdot f(x_k) \quad (3.28)$$

$$\text{AB2: } x_{k+1} = x_k + h \cdot \left[\frac{3}{2} \cdot f(x_k) - \frac{1}{2} \cdot f(x_{k-1}) \right] \quad (3.29)$$

The stability region of the two methods is shown in Fig. 3.15, in red for FE and in blue for AB2. This shows that AB2 has a smaller region of stability. However, when stable, it is expected to perform better than FE since it is a second order method, and thus limits error propagation between steps. The latter can be of great importance since the implementation of the algorithm resides in a low-accuracy environment (see more in section 3.3).

Based on the integration scheme, the dedicated computation pipelines are synthesized on the FPGA. A schematic representation of pipelined version of FE and AB2 is shown in Figs. 3.16. *Fixed-point arithmetic* has been used along the datapath of each pipeline. The characteristic of fixed-point arithmetic is the fixed number of decimal bits in the representation of a real number. There are two benefits from adopting it in a datapath.

- faster execution of arithmetic operations, and
- lower silicon footprint of arithmetic modules, i.e. resource utilization in the FPGA.

However, this comes at the cost of accuracy loss as it will be explained in section 3.3.

After all pipelined computations are finished, the nodal injections are updated in parallel and the analog part performs the linear algebra operation for the next step. This interaction with the grid is done using a dedicated interface to the FPPNS, seen on the bottom of Fig. 3.14. The

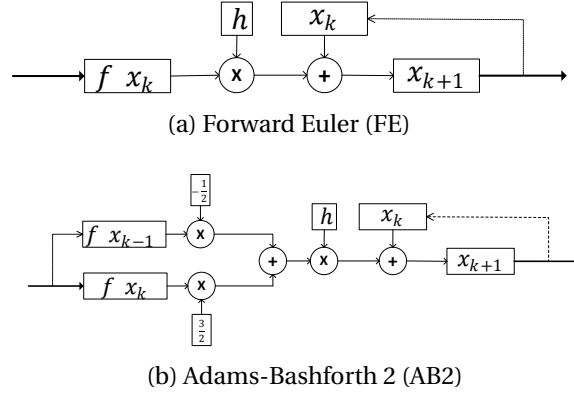


Figure 3.16 – Pipelined versions of the FE and the AB2 algorithms

computational pipelines operate in a DMA fashion, reading and writing data to the ADC/DAC array of the previous section using dedicated I/O pins. The analog grid driver also manages the connectivity of the FPPNS via the digitally controlled switches of the latter. Also notice that any parameter of the analog part (e.g. line, switch status) can be changed in runtime without manual intervention.

Example

A simple example for the above is a set of generators modeled using the classical generator model. They are all put under the same equivalence class E_k , and they are all described using a similar set of DAEs (the swing equation), adapted to the different parameters of each individual generator (mechanical starting time and transient impedance). The dynamics of generators modeled by the classical generator model are given by the swing equation.

$$\dot{\omega} = \frac{P_m - P_e - D \cdot (\omega - \omega_0)}{M} \quad (3.30a)$$

$$\dot{\delta} = \omega - \omega_0 \quad (3.30b)$$

$$I = \frac{V - (E' \angle \delta)}{r_a + j \cdot x'} \quad (3.30c)$$

$$re\{P_e\} = V \cdot I^* \quad (3.30d)$$

Where ω is the synchronous speed of the generator, P_m is the mechanical power acting on the rotor of the generator, P_e is the electrical power that is supplied by the generator to the grid, D is a damping coefficient, ω_0 is the nominal synchronous speed, M is the mechanical starting time, δ is the machine internal angle, E' is the machine internal voltage amplitude, V is the complex terminal voltage, I is the complex current that is injected into the grid, r_a is the armature resistance, x' is the transient reactance.

It is customary for “classical” transient stability studies to model generators with the above set of equations [291]. When first-swing only instability is of concern, damping is also often neglected, and results are rather on the pessimistic side. The above can be written in the form of (3.27) as follows.

$$x := \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \omega \\ \delta \end{bmatrix} \quad (3.31a)$$

$$y := \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} V \\ I \\ P_e \end{bmatrix} \quad (3.31b)$$

$$\lambda := [P_m \quad D \quad \omega_0 \quad M \quad E' \quad r_a + j \cdot x'] \quad (3.31c)$$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} (\lambda_1 - y_3 - \lambda_2 \cdot (x_1 - \lambda_3)) / \lambda_4 \\ x_1 - \lambda_3 \end{bmatrix} := \begin{bmatrix} f_1(x, y, \lambda) \\ f_2(x, y, \lambda) \end{bmatrix} \quad (3.31d)$$

$$0 = \begin{bmatrix} y_1 - (\lambda_5 \cdot e^{j \cdot x_2}) - y_2 \cdot \lambda_6 \\ \operatorname{re}\{y_1 \cdot y_2^*\} - y_3 \end{bmatrix} := \begin{bmatrix} g_1(x, y, \lambda) \\ g_2(x, y, \lambda) \end{bmatrix} \quad (3.31e)$$

The dedicated pipeline for the computations of generators modeled with the classic generator model using FE integration is shown in Fig. 3.17. The figure presents a detailed view on the fixed-point datapath. The width of the pipeline appears in green, and the time spent on each block (in FPGA clock cycles) is shown in brown.

Similar classes of equations can be derived for other power system elements connected to buses, e.g. dynamic loads, synchronous condensers, induction machines, etc. For each one of the equivalence classes similar dedicated pipelines can be created based on the integration scheme that is used. In the current implementation 8 pipelines have been created, for generators with the classical model, for constant impedance/current/constant power loads, each using FE or AB2 numerical integration.

Communication interface

A Cypress CY7C68016 USB 2.0 controller is interfaced to the FPGA via a two-port RAM of 4096 positions for words of 32 bits. The configuration is seen in Fig. 3.18. Access to the shared RAM is provided to the host PC by a dedicated driver utility as described in the next section. Threaded USB 2.0 read and write operations of up to 65.69 Mbps [292] are supported.

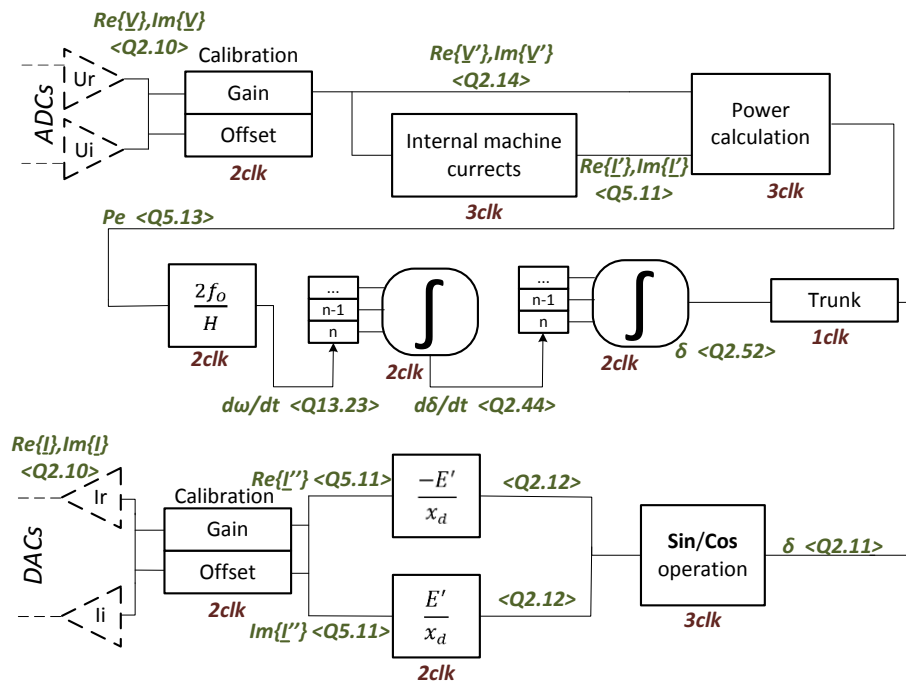


Figure 3.17 – Datapath of the synthesized pipeline for generators that are modeled with the classical generator model of (3.30) using the FE integration scheme of (3.28)

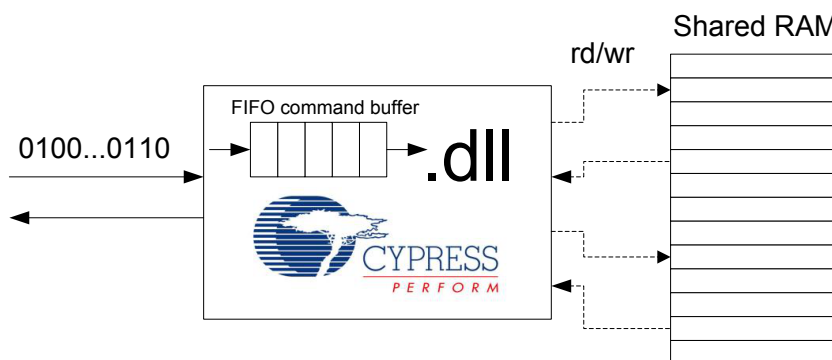


Figure 3.18 – The USB controller and the shared RAM that is interfaced to the FPGA

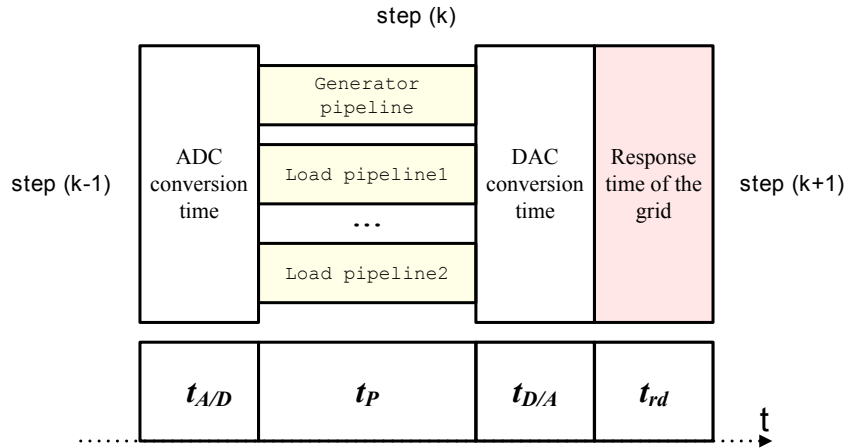


Figure 3.19 – Schematic diagram of the timing break up of the operations of the FPPNS for one simulation step

3.1.3 Timing

Fig. 3.19 presents a schematic diagram of the timing break of the operations of the FPPNS. The total time per iteration is as follows.

$$t_{it} = t_{A/D} + t_P + t_{D/A} + t_{rd} \quad (3.32)$$

Where

- $t_{A/D} = 200ns$ is the time for the ADC array to read the values from the analog grid. The current implementation is based on AD7356 converters operated at 5 MSPS (Million Samples Per Second).
- t_P is the pipeline computing time. This value is explained hereunder.
- $t_{D/A} = 400ns$ is the time for the DAC array to update the analog inputs (injections) to the analog grid. The current implementation is based on AD5449 converters.
- t_{rd} is the waiting time for the analog electronics to settle, and will be explained in section 3.3.1.

Regarding t_P , the time that takes each pipeline to complete is

$$t_{Pi} = (\ell_{Pi} + N_{Pi} - 1) \cdot t_{clk}. \quad (3.33)$$

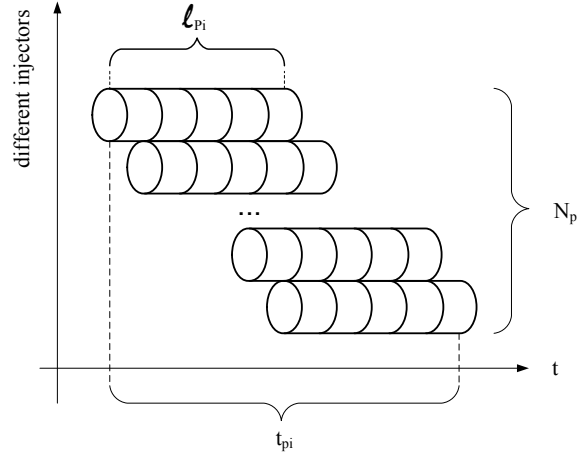


Figure 3.20 – Diagram of the timing break up of a computing pipeline

Where ℓ_{P_i} is the length of pipeline i in clock cycles. In the example of Fig. 3.17 it is $\ell_{P_i} = 22$, $t_{clk} = 1/f_{clk} = 8ns$ is the clock period (in the current implementation the digital clock frequency is $f_{clk} = 125 MHz$) and N_{P_i} is the number of injections to be processed by the pipeline. The equation becomes clearer in view of Fig. 3.20. Since all pipelines run in parallel, the time that the digital part of the computations takes to complete is the maximum time spent in any of the pipelines, $t_P = \max_i t_{P_i}$

3.2 Software

Alongside the dedicated hardware a co-design software has been developed, named `elab-tsaot`². The entire application is written in C++. Compilation has been done using MinGW 4.7 32 bit. The frontend uses the Qt framework and widgets from the Qwt library. Parts of the Boost C++ Libraries have been used in the backend (Random, Timer, Chrono, Any, uBLAS, Pointer Container, Bimap). Linear algebra functionality is provided through the BLAS and LAPACK from Netlib. An overview is shown on Fig. 3.21

3.2.1 Backend

The backend is the core of the application. A software model of the power system has been created (`Power system model`). Models can be imported and exported from XML files that adhere to a specially tailored schema. The power system model can be fed into the steady state engine (`SS engine`) which is responsible of making all sorts of steady calculations on the power system.

An interface design pattern has been followed for `SS engine` as shown in Fig. 3.22a. Imple-

²Available on the GIT repository <https://kyriakid@git.epfl.ch/repo/elab-tsaot2.git>. Access is upon request to the author.

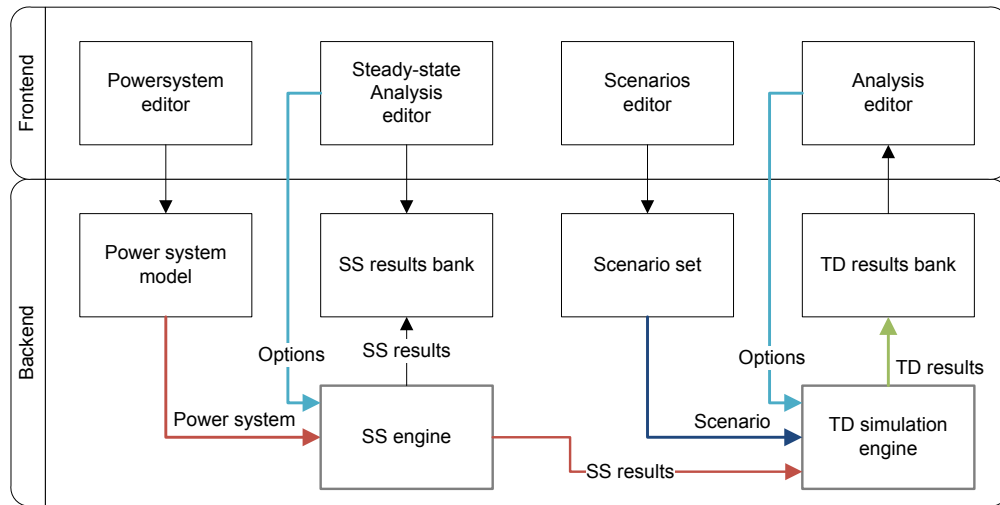


Figure 3.21 – Architecture overview of elab-tsaot

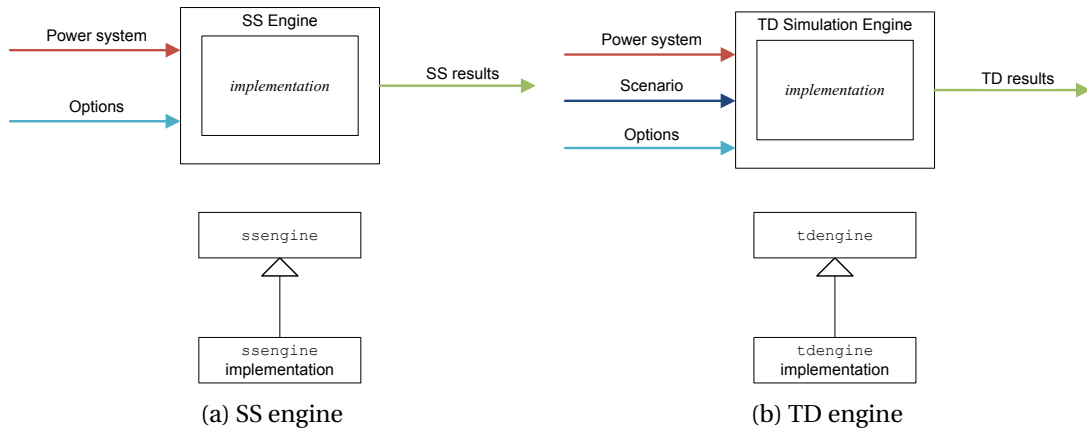


Figure 3.22 – Interface design pattern for the SS and TD engines

mentations that want to qualify as SS engines for elab-tsaot have to respect the API defined by the abstract class `ssengine`. Its inputs are shown schematically in the figure: the power system to be solved and the options on which analysis to perform and how to perform it. After the analysis is complete, its results are stored in the `SS results` bank.

A current `ssengine` software implementation can perform power flow analysis on a given power system. It implements a Newton-Raphson power flow using polar coordinates [37]. Linear algebra functionality for the above is offered by the Netlib BLAS and LAPACK libraries.

Once the power flow has been solved, the power system model is updated with its steady state values and it is fed as an input to the time domain simulation engine (TD simulation engine). A similar interface pattern has been used for it, as shown in Fig. 3.22b.

The three inputs to the TD engine are the power system, the scenario to be executed, and

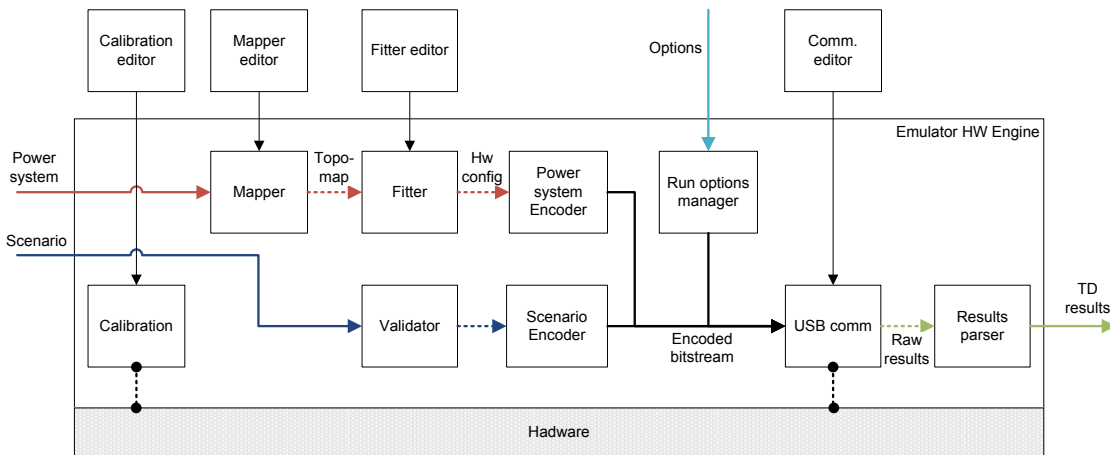


Figure 3.23 – Hardware Abstraction Layer (HAL) of the dedicated hardware

options for the simulation. For a *power system* to be ready for TD analysis, the power flow needs to be already available. A *scenario* is the set of events that will take place in the time-domain simulation window. Scenarios are created in a special user interface *Scenario editor* and stored in a dedicated store (*Scenario set*). They can also be imported/exported as an xml file that adheres to a dedicated schema. *Options* are auxiliary inputs to the TD engine, such as the time step, duration of the simulation, the desired variables to store in the results etc. They are provided to the engine by the user. The result of a run of the TD engine is a set of waveforms (*TD results*) which are stored in a dedicated store (*TD results bank*). Results can be imported from and exported to text files with a special format.

Two TD engine implementations are provided in the current prototype. The first one concerns the software class that uses the dedicated hardware to perform the TD analysis. In the core of the it there is a Hardware Abstraction Layer (HAL) of the hardware, as shown in Fig. 3.23. The HAL respects the API of Fig. 3.22b through the inputs *Power system*, *Scenario*, *Options* and *TD results* - color coding of Fig. 3.22b has been retained.

The HAL contains a model of the hardware and provides a set of functionality to bridge the software with the hardware world.

Calibration

The purpose of calibration is to increase the accuracy of the hardware platform. Calibration is detailed in section 3.3.4.

Mapper

The mapper utility creates the association between the power system topology and the analog and digital resources of the hardware. It corresponds to the topological mapping operation of section 3.1.1. It is similar to the “place and route” operation in the design flow of modern FPGAs. Mapper information can be imported from and exported to an xml file that adheres to a dedicated schema.

Fitter

After the mapping is complete the fitter is responsible of calculating all the exact parameters for all the resources of the hardware. This involves translating values from the power system to the electronics domain, as described in the *value mapping* operation of section 3.1.1.

Power system and Scenario Encode

The next step is to encode the information of the hardware model into a format that the machine itself can understand, i.e. into an equivalent of a “binary” code. In our platform this is called the *bitstream*. The bitstream contains the binary encoding of the entire hardware of the dedicated platform (values of potentiometers, status of switches, etc.) which are to be communicated to the platform. This operation is similar to the “assembler” operation in the design flow of modern FPGAs.

USB Comm

The HAL encapsulates communication to the hardware through a dedicated USB driver. A wrapper dynamic-link library (DLL) has been developed based on the Cypress CyUSB library. The drivers provide direct access to the shared memory of the USB comm module of the hardware (see Fig. 3.18). Threaded USB 2.0 read and write operations are supported. The protocol used to communicate commands and data between the two platforms is a flag-pollled shared memory scheme.

Results parser

After an operation is finished executing on the hardware, the results are communicated back to the software application. These raw results are translated into power system quantities by the HAL. It is the result of this parsing that meets the TD results requirements of the `tdengine` software API.

A second TD engine implementation is a purely software based one. It implements the partitioned solution scheme of Fig. 3.5 by purely software techniques. BLAS and LAPACK provide the linear algebra functionality to solve the grid nodal equation $I = Y \cdot V$. The FE and AB2 numerical integration algorithms of (3.28) and (3.29) are used to solve the DAE sets (3.27) of each injector. The purpose of this research-grade software engine is to provide an accuracy benchmark of the hardware-based one.

3.2.2 Frontend

Fig. 3.24 shows the software design pattern followed for the frontend of `elab-tsaot`. The *model* is the core of the application. It maintains the state and the data represented for every view of the GUI. When changes occur, the model updates all its views. The *controller* is the interface presented to the user (buttons, tabular interfaces, menus, etc.) that allows it to manipulate the application. Finally, the *view* is the user interface which displays the

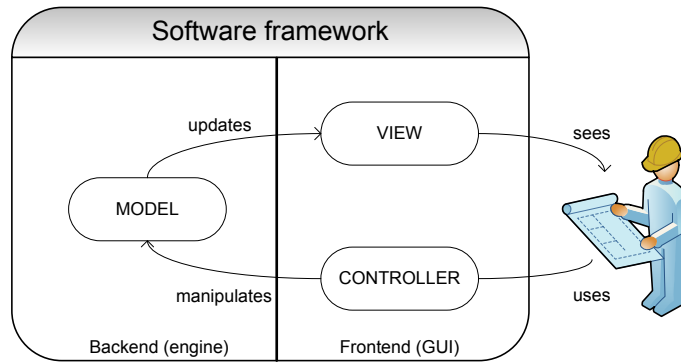


Figure 3.24 – The model-view-controller (MVC) software architectural pattern used to implement the frontend of `elab-tsaot`

information contained in the model. Any object that needs information about the model has to be a registered view with the model. The user *sees* the views and that he *manipulates/uses* the controller.

Frontend equivalent of HAL blocks have also been created for the facilitation of the user, as shown in Fig. 3.23. Calibration, mapper, fitter and communication operations have their dedicated frontends - Calibration editor, Mapper editor, Fitter editor, and Comm. editor respectively.

3.3 Inaccuracy

The inaccuracy sources of the hardware platform can be categorized as digital and analog.

3.3.1 Analog inaccuracy

On the analog side, both the topological mapping of Fig. 3.9 and the value mapping of (3.20) are in the mathematical sense perfectly accurate. By that, we mean that there is a perfect translation of the power system quantities/topology to the electronic quantities/topology. By the time the electronic quantities are realized in real hardware, inaccuracy has to be considered.

The main causes of inaccuracy can be categorized in points (L) that refer to the accuracy of the electrical lines, and points (N) that refer to the accuracy of the electrical nodes and their injections (currents and voltages).

- (a) the quantization and saturation of the resistor values due to the finite resolution and finite range of the potentiometers (L),
- (b) the inherent inaccuracy of the potentiometer (L),

- (c) parasitic impedances of the lines and the configuration switches (L),
- (d) the quantization and saturation of the written voltage/current values due to the finite resolution and range of the DACs (N),
- (e) the quantization and saturation of the read voltage values due to the finite resolution and range of the ADCs (N),
- (f) non-linearities (integral and differential) and errors (offset and gain) of the DACs and ADCs (N), and
- (g) the finite waiting time for the analog electronics to settle due to RC time constant effects (N).

In what follows an effort is made to analyze and quantify the above. In this section, bold typeface is used as a notational convention for random variables.

Line inaccuracies

Given (3.15) for the functioning of the potentiometer the maximum attainable value is

$$r_{pot} = \frac{2^M - 1}{2^M} \cdot R. \quad (3.34)$$

When a specific value r_{req} is requested from the potentiometer, t has to be calculated so that the resulting r_{pot} from (3.15) best approximates r_{req} . This tap setting can be calculated from (3.15). However, the real full-scale value R is a priori unavailable. Instead it is known in the form of a random distribution.

$$\mathbf{R} \sim \mathcal{N}(\hat{R}, \sigma_R^2) \quad (3.35)$$

This distribution may be known from the manufacturing process or may be the result of measurements. Equation (3.15) is solved using the expected full-scale value.

$$t^{\mathbb{R}} = \frac{r_{req}}{\hat{R}} \cdot 2^M \quad (3.36)$$

Notice that $t^{\mathbb{R}}$ in (3.36) is not random, and also not integer, as required by (3.15). Due to the

discrete nature of the reconfigurable element, it has to be *quantized* and *saturated* according to the limits.

$$\mathbf{t} = \begin{cases} 0 & \text{if } t^{\mathbb{R}} \leq 0 \\ \lfloor t^{\mathbb{R}} + 0.5 \rfloor & \text{otherwise} \\ 2^M - 1 & \text{if } t^{\mathbb{R}} \geq 2^M - 1 \end{cases} \quad (3.37)$$

In the non-saturated case t is rounded half towards positive infinity. If the requested value is outside the attainable limits of the element, then the tap setting saturates accordingly, to the value closest to the one requested.

Assuming a uniformly random r_{req} within the attainable limits of (3.34), the resulting $t^{\mathbb{R}}$ from (3.36) is also uniformly random within the tap limits. Accordingly, the \mathbf{t} calculated from (3.37) is also random: it is always selected to the closest bit rounding to $t^{\mathbb{R}}$.

$$\mathbf{t} = t^{\mathbb{R}} + \mathbf{q}_t \quad (3.38)$$

Where \mathbf{q}_t is the effect of quantization on the tap setting. According to the above it is safe to assume that $\mathbf{q}_t \sim \mathcal{U}(-1/2, +1/2)$, i.e. the quantization error is at most half a bit. The mean and variance of q_t are according to the uniform distribution.

$$E[\mathbf{q}_t] = 0 \quad (3.39a)$$

$$var[\mathbf{q}_t] = 1/12 \quad (3.39b)$$

$$(3.39c)$$

In turn the mean and variance of \mathbf{t} are

$$E[\mathbf{t}] = t^{\mathbb{R}}, \quad (3.40a)$$

$$var[\mathbf{t}] = 1/12. \quad (3.40b)$$

Substituting (3.38) into (3.15) yields the final (random) output of the reconfigurable element.

$$\mathbf{r}_{\text{pot}} = \frac{r_{\text{req}}}{\hat{R}} \cdot \mathbf{R} + \frac{1}{2^M} \cdot \mathbf{R} \cdot \mathbf{q}_t \quad (3.41)$$

The mean and variance of (3.41) are as follows.

$$E[\mathbf{r}_{\text{pot}}] = r_{\text{req}} \quad (3.42a)$$

$$\text{var}[\mathbf{r}_{\text{pot}}] = \underbrace{\left(\frac{r_{\text{req}}}{\hat{R}}\right)^2 \cdot \sigma_R^2}_{\text{due to analog inaccuracies}} + \underbrace{\frac{1}{12} \cdot \left(\frac{\hat{R}}{2^M}\right)^2}_{\text{due to quantization}} \quad (3.42b)$$

From (3.42a) it becomes clear that in the way the reconfigurable potentiometer is operated its output is on average equal to the desired value X_{req} . In (3.42b) the two sources of the inaccuracy are highlighted.

An error variable can be defined as $\epsilon_{\mathbf{X}} = \mathbf{r}_{\text{pot}} - r_{\text{req}}$. Then for this error variable it holds.

$$E[\epsilon_{\mathbf{r}}] = 0 \quad (3.43a)$$

$$\text{var}[\epsilon_{\mathbf{r}}] = \text{var}[\mathbf{r}_{\text{pot}}] \quad (3.43b)$$

The above analysis covers points (a) and (b) from the list in the beginning of this subsection. In the current implementation the expected full scale resistance is $\hat{R} = 10000 \Omega$, the bit resolution is $M = 8$ bits and variance of the full-scale resistance is $\sigma_R^2 = (666.7 \Omega)^2$. The last value is inferred by the nominal tolerance which is provided by the manufacturer of the potentiometers as a $\pm 3\sigma$ value in percent - in the case of our potentiometers 20%. Then (3.43b) becomes

$$\text{var}[\epsilon_R] = 4.4 \cdot 10^{-3} \cdot r_{\text{req}}^2 + 127.15. \quad (3.44)$$

Regarding point (c) of the list, an approximated schematic of the actual realization of the circuit between the two end nodes of an electrical branch that contains a potentiometer can be seen in Fig. 3.25.

Parasitic capacitors (c_{ndf} , c_{cd} , c_{pot} , c_{sw} and c_{ndt}) and inductors (l_{cd} , l_{pot} and l_{sw}) are not considered here since they do not affect the steady state functioning of the circuit. The actual

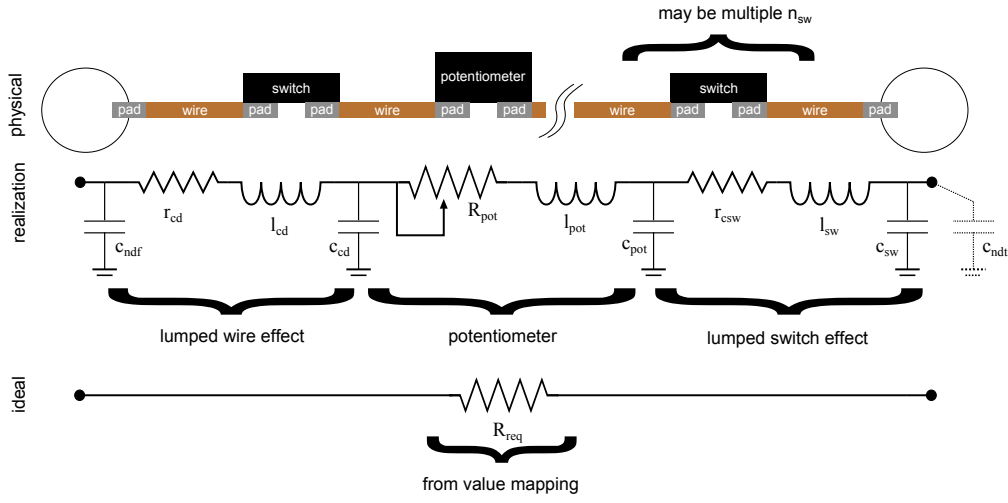


Figure 3.25 – Physical and schematic representation of parasitics for an electrical branch that contains a potentiometer

final resistance of the branch is given by the following.

$$r = r_{pot} + \underbrace{r_{cd} + r_{csw}}_{r_{off}} \quad (3.45)$$

The additional offset resistance r_{off} is the combined result of the resistance of the conductors that are involved in the electrical path r_{cd} as well as the imperfect non-zero resistance of the electronic switches in the same path.

Ideally the conductors have a zero resistance. However in reality their effective resistance is given by Pouillet's law.

$$r_{cd} = \rho_{cd} \cdot \frac{\ell_{cd}}{A_{cd}} \quad (3.46)$$

Where ρ_{cd} is the resistivity of the material, ℓ_{cd} is the total length and A_{cd} is the area of the cross-section of the conductor. Fixed cross-section across the length is considered in the above. In the current implementation the conductors are copper PCB traces with a rectangular

cross-sections.

$$\rho_{cd} = 1.68 \cdot 10^{-8} \Omega \cdot \text{m} \text{ (the resistivity of copper)} \quad (3.47a)$$

$$A_{cd} = 254 \text{ um} \times 35 \text{ um} = 8890 \text{ um}^2 \quad (3.47b)$$

The length ℓ_{cd} varies depending on the exact electrical branch considered on the prototype. A good approximation in the current prototype is as follows.

$$\ell_{cd} \approx 5.5 \text{ cm} . \quad (3.48)$$

By using (3.46) the average conductor resistance is $r_{cd} \approx 0.1 \Omega$.

The reconfiguration of the topology of the resistor network is effectuated by the switching of electronic switches. Depending on the routing of the line on the electronic board a different number n_{sw} of switches can be in its path. The term r_{csw} in (3.45) is the lumped effect of all of them.

$$r_{csw} = n_{sw} \cdot r_{sw} \quad (3.49)$$

When a switch is open its nominal resistance is $r_{sw} \rightarrow \infty$ and when closed it is $r_{sw} \rightarrow 0$. However the latter is not always accurate due to manufacturing imperfections. In our case the average on-resistance of the digitally controlled switches is $r_{sw} \approx 2.5 \Omega$. On average $n_{sw} = 2$ switches are involved in the electrical path of a potentiometer (one on each side of it). Hence the combined effect of them is $r_{csw} \approx 5 \Omega$.

In total the offset value in (3.45) is

$$r_{off} \approx 5.1 \Omega . \quad (3.50)$$

This value is at least two orders of magnitude smaller than the effective values of resistors used in the topology in most of the times and thus it can be neglected.

Voltage injection inaccuracies

Equation (3.17) holds for the voltage DACs of the FPPNS. An inaccurate version of it is as follows (V_{off} is neglected since it does not affect the computations).

$$\mathbf{V}_{DAC} = \frac{\mathbf{t}}{2^M} \cdot (\mathbf{V}_{FS} + \epsilon_{DAC1}) + \epsilon_{DAC2} \quad (3.51)$$

The random variable \mathbf{V}_{FS} comes from a circuit that generates the full-scale reference voltage of the DAC. It is a combination of an Analog Devices AD5624 16-bit, B Grade nanoDAC with a fixed output, in series with a buffer Analog Devices AD8685 low-noise, precision CMOS op. amp. The random variable for the generated voltage is as follows.

$$\mathbf{V}_{FS} \sim \mathcal{N}(\hat{V}_{FS}, \sigma_{FS}^2) \quad (3.52)$$

The reference is set to $\hat{V}_{FS} = 5 \text{ V}$ and the variance is $\sigma_{FS}^2 = (305 \cdot 10^{-6} + 50 \cdot 10^{-6})^2$. The latter is taken from the datasheets of the components:

- a maximum (i.e. $3 \cdot \sigma$) inaccuracy of ± 12 bits for 5 V for the AD5624, and
- a typical (i.e. $1 \cdot \sigma$) inaccuracy of $50 \text{ }\mu\text{V}$ for a supply voltage of 5 V for the AD8658.

The additional error terms in (3.51) are determined by the characteristics of the Analog Devices AD5449 12-bit DAC that has been used in the design.

- $\epsilon_{DAC1} \sim \mathcal{N}(0, \sigma_{DAC1}^2)$, corresponds to a gain error of $V_{FS} \pm 0.5\%$. Hence, for $V_{FS} = 5 \text{ V}$ it is $\sigma_{DAC1} = 25 \cdot 10^{-3}$.
- $\epsilon_{DAC2} \sim \mathcal{N}(0, \sigma_{DAC2}^2)$, corresponds to a relative accuracy of $\pm 1 \text{ LSB}$. The value of an LSB can be calculated from (3.17) $\sigma_{DAC2} = 1.22 \cdot 10^{-3}$.

The functioning of voltage DACs follows a similar line to the functioning of potentiometers. When a V_{req} is requested from the voltage DAC, t has to be calculated so that the resulting V_{DAC} from (3.17) best approximates V_{req} . So, the latter is solved for t .

$$t^{\mathbb{R}} = \frac{V_{req}}{\hat{V}_{FS}} \cdot 2^M \quad (3.53)$$

Notice that $t^{\mathbb{R}}$ in (3.53) is not random, and also not integer, as required by (3.17). Due to the

discrete nature of the reconfigurable element, it has to be *quantized* and *saturated* according to the limits, similarly to (3.37). Again, in the non-saturated case t is rounded half towards positive infinity. If the requested value is outside the attainable limits of the element, then the sign tap setting saturate accordingly, to the value closest to the one requested.

Assuming a uniformly random V_{req} within the attainable limits of the voltage DAC, the resulting $t^{\mathbb{R}}$ from (3.53) is also uniformly random within the tap limits. Accordingly, the \mathbf{t} calculated from (3.37) is also random: it is always selected to the closest bit rounding to $t^{\mathbb{R}}$, as per (3.38). The term $\mathbf{q}_{\mathbf{t}}$ stands for the effect of quantization on the tap setting and follows the uniform distribution of (3.39). By using the distributions of all the random variables, the mean and variance of (3.51) can be calculated.

$$E[\mathbf{V}_{\text{DAC}}] = V_{req} \quad (3.54a)$$

$$var[\mathbf{V}_{\text{DAC}}] = \underbrace{\left(\frac{V_{req}}{\hat{V}_{FS}}\right)^2 \cdot (\sigma_{FS}^2 + \sigma_{DAC1}^2) + \frac{1}{12} \cdot \left(\frac{1}{2^M}\right)^2 \cdot (\sigma_{FS}^2 + \sigma_{DAC1}^2) + \sigma_{DAC2}^2}_{\text{due to analog inaccuracies}} + \underbrace{\frac{1}{12} \cdot \left(\frac{\hat{V}_{FS}}{2^M}\right)^2}_{\text{due to quantization}} \quad (3.54b)$$

From (3.54a) it becomes clear in the way the reconfigurable is operated, its output is on average equal to the desired value V_{req} . In (3.54b) the two sources of the inaccuracy are highlighted. Again, an error variable can be defined as $\epsilon_V = \mathbf{V}_{\text{DAC}} - V_{req}$. Then for this error variable it holds.

$$E[\epsilon_V] = 0 \quad (3.55a)$$

$$var[\epsilon_V] = var[\mathbf{V}_{\text{DAC}}] \quad (3.55b)$$

Substituting values for the current implementation in (3.54b) yields the expression for the variance of voltage injections in the current system.

$$var[\mathbf{V}_{\text{DAC}}] = V_{req}^2 \cdot 2.5 \cdot 10^{-5} + 1.5 \cdot 10^{-6} + 1.24 \cdot 10^{-7} \quad [V^2] \quad (3.56)$$

The first two terms are due to the inaccuracy of the analog components and the last one is due to quantization. This analysis covers points (d) and (f) of the list in the beginning of the section for voltage injections.

Current injection inaccuracies

A similar line can be followed for current injections in the FPPNS. It was presented in (3.18) that current injections are done by the voltage-to-current converted topology of Fig. 3.13. The voltage of the nominator of (3.18) follows the analysis of (3.51) and the conversion resistance in the denominator of (3.18) follows the analysis of (3.41). The mean value of I_{DAC} is as follows.

$$E[\mathbf{I}_{DAC}] = \frac{E[\mathbf{V}_{DAC}]}{E[\mathbf{R}_{conv}]} \stackrel{(3.54a)(3.42a)}{=} \frac{V_{req}}{R_{conv}} \quad (3.57)$$

In the above R_{conv} is the fixed resistance value asked by the conversion resistance and V_{req} is the value asked to the voltage DAC. The latter is computed as

$$V_{req} = I_{req} \cdot R_{conv} \quad (3.58)$$

Where I_{req} is the current asked as a current injection.

The variance of I_{DAC} is determined by the rule to find the variance of the fraction of two independent random variables.

$$var[\mathbf{I}_{DAC}] = var\left[\frac{\mathbf{V}_{DAC}}{\mathbf{R}_{conv}}\right] = \frac{var[\mathbf{V}_{DAC}] \cdot E[\mathbf{R}_{conv}]^2 + var[\mathbf{R}_{conv}] \cdot E[\mathbf{V}_{DAC}]^2}{E[\mathbf{R}_{conv}]^2} \quad (3.59)$$

The analytical expression of the result is omitted here for brevity sake. By defining an error variable can be defined as $\epsilon_I = \mathbf{I}_{DAC} - I_{req}$ the distribution of this error can be found.

$$E[\epsilon_I] = 0 \quad (3.60a)$$

$$var[\epsilon_I] = var[\mathbf{I}_{DAC}] \quad (3.60b)$$

This analysis covers the points (d) and (f) of the list in the beginning of the section, for current injections.

Voltage measurement inaccuracies

A similar line of analysis is followed for all measurement devices in the system (voltage and current ADCs). Equations (3.19) describe the perfect functioning of the voltage ADCs in the

system. The voltage measurement subsystem of the FPPNS is based on a AD7356 differential input 12-bit successive approximation (SAR) ADC by Analog Devices. Assuming a random V_{in} within the limits of the ADC an inaccurate version of (3.19a) is given hereunder.

$$\mathbf{D}^{\mathbb{R}} = \frac{V_{in}}{V_{FS}} \cdot (2^M + \epsilon_{FS}) + \epsilon_{mid} + \epsilon_{INL} \quad (3.61)$$

V_{FS} is the full-scale reference voltage of the ADC converter. It is provided by an Analog Devices ADR445B LDO XFET ultralow noise voltage reference. The multiplicative error term ϵ_{FS} is a full-scale gain error. The additive error term ϵ_{mid} is a mid-scale offset. The additive error term ϵ_{INL} is due to the integral non-linearity of the converter. Based on the datasheets of the components of the actual implementations, the random distributions are

- $V_{FS} \sim \mathcal{N}(\hat{V}_{FS}, \sigma_{V_{FS}}^2)$, where $\hat{V}_{FS} = 5 \text{ V}$ and $\sigma_{V_{FS}} = 2 \cdot 10^{-3} \text{ V}$.
- $\epsilon_{FS} \sim \mathcal{N}(0, \sigma_{FS})$, where $\sigma_{FS} = 1 \text{ bit}$.
- $\epsilon_{mid} \sim \mathcal{N}(\mu_{mid}, \sigma_{mid}^2)$, where $\mu_{mid} = 5 \text{ bits}$ and $\sigma_{mid} = 1.66 \text{ bits}$.
- $\epsilon_{INL} \sim \mathcal{N}(0, \sigma_{INL}^2)$, where $\sigma_{INL} = 0.5 \text{ bit}$.

If the input is outside the allowable limits of the ADC, then the digital output saturates to the value that represents as close as possible the input. If the input is assumed uniformly random within the saturation limits then the quantization operation of (3.19b) can be rewritten as follows.

$$\mathbf{D} = \mathbf{D}^{\mathbb{R}} + \mathbf{q}_D \quad (3.62)$$

The quantization term \mathbf{q}_D follows a uniform distribution.

$$\mathbf{q}_D \sim \mathcal{U}(-1/2, +1/2) \quad (3.63)$$

Through (3.19c) this yields the random value expression for V_{ADC} .

$$\mathbf{V}_{ADC} = \underbrace{\frac{\hat{V}_{FS}}{2^M} \cdot \left(\frac{V_{in}}{V_{FS}} \cdot (2^M + \epsilon_{FS}) + \epsilon_{mid} + \epsilon_{INL} \right)}_{\text{due to analog inaccuracies}} + \underbrace{\frac{\hat{V}_{FS}}{2^M} \cdot \mathbf{q}_D}_{\text{due to quantization}} \quad (3.64)$$

The mean and the variance of V_{ADC} are as follows.

$$E[V_{ADC}] = V_{in} + \frac{\mu_{mid} \cdot \hat{V}_{FS}}{2^M} \quad (3.65a)$$

$$var[V_{ADC}] = \underbrace{V_{in}^2 \cdot \left(\frac{\sigma_{V_{FS}}^2}{V_{FS}^2} + \frac{\sigma_{FS}^2}{2^{2 \cdot M}} \right) + \left(\frac{\hat{V}_{FS}}{2^M} \right)^2 \cdot (\sigma_{mid}^2 + \sigma_{INL}^2)}_{\text{due to analog inaccuracies}} + \underbrace{\frac{1}{12} \cdot \left(\frac{\hat{V}_{FS}}{2^M} \right)^2}_{\text{due to quantization}} \quad (3.65b)$$

An error variable can be defined as $\epsilon_{V_{ADC}} = V_{ADC} - V_{in}$. Then for this error variable it holds.

$$E[\epsilon_{V_{ADC}}] = 0 \quad (3.66a)$$

$$var[\epsilon_{V_{ADC}}] = var[V_{ADC}] \quad (3.66b)$$

Substituting the values for the current implementation in (3.65b) an analytical expression for the variance is derived.

$$var[V_{ADC}] = V_{in}^2 \cdot 2.2 \cdot 10^{-7} + 4.48 \cdot 10^{-6} + 1.24 \cdot 10^{-7} \quad (3.67)$$

The first two terms are due to the inaccuracy of the analog components and the last one is due to quantization. This analysis covers points (e) and (f) of the list in the beginning of the section.

Parasitical capacitances

Point (g) strongly relates to the parasitical capacitances and inductances that appear in the electronic circuit. This can be partly seen in Fig. 3.25. Parasitic inductances for this scale of conductive lines, i.e. lengths in the order of few centimeters up to few tens of centimeters, can safely be neglected. However, this is not the case for capacitances.

Parasitic capacitances appear not only in external nodes, at the ends of integrated circuits (ICs) as show schematically in Fig. 3.25, but in every electrical node of the circuit, including internal nodes of the ICs. A simplifying assumption is made that parasitic capacitances appear only as shunts from each electrical node to ground and not as series in electrical branches of the circuit.

The FPPNS is operated in *steady-state*, i.e. with all variables that describe its behavior (voltages, currents) being invariant with time. The result of the parasitic capacitances is that changes in

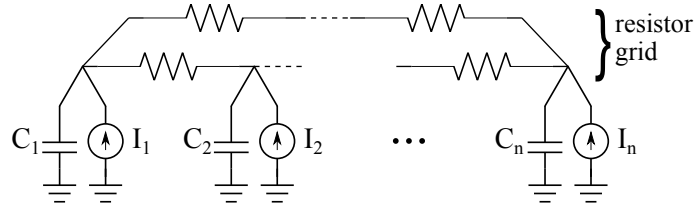


Figure 3.26 – Schematic representation of the resulting electrical circuit taking into account the node parasitic capacitances

Table 3.1 – Time domain and Laplace domain

Component	Time domain	Laplace domain
resistor	$v = R \cdot i$	$V(s) = R \cdot I(s)$
capacitor	$\frac{\partial v}{\partial t} = \frac{1}{C} \cdot i$	$s \cdot C \cdot V(s) - C \cdot V_0 = I(s)$

the electrical steady state cannot occur instantaneously. Instead the circuit goes through a *transient state* during which variables change non-periodically. Gradually the transient state vanishes and the new steady state is reached. This transient is mainly due to the physical requirement to charge the parasitic capacitors. This procedure cannot be instantaneous since this would require infinite power [293].

When a read command is issued by the computational pipelines to the ADCs, the latter sample-and-hold the voltage at their input. All other things considered perfect, these FPPNS voltages are not in their final (correct) values until the transient is over. Therefore if the read command is issued before the new steady state is reached, erroneous results will be output by the FPPNS for the solution of the linear system.

The exact way to model this transient of the electrical circuit is out of the scope of this work. The reader is referred to relevant bibliography instead [293, 294, 295]. Hereunder there is brief attempt to gain an insight on the mechanism of the phenomenon with the help of Laplace transform.

The resulting analog circuit contains a resistor grid, and at each node a parasitic capacitor and a current source. Fig. 3.26 shows a schematic representation. Some of the nodes may contain voltage source and the analysis remains the same. Table 3.1 contains the Laplace-domain equivalents of the resistor and capacitor.

The time-domain equations of the grid are follows.

$$\underbrace{G \cdot V}_{\text{resistor grid}} + \underbrace{\text{diag}(C) \cdot \begin{bmatrix} \frac{\partial V_1}{\partial t} \\ \frac{\partial V_2}{\partial t} \\ \vdots \\ \frac{\partial V_n}{\partial t} \end{bmatrix}}_{\text{parasitic capacitors}} = \underbrace{I}_{\text{current sources}} \quad (3.68)$$

Where G is the conductance matrix resulting from the resistor network, V is the node voltage vector of the circuit, $C = [C_1 \ \dots \ C_n]^T$ is the vector of the node parasitic capacitances and I is the current injection vector. The Laplace transform $\mathcal{L}\{\cdot\}$ is applied to the above. An initial voltage condition at each node (capacitor) of $V^0 = [V_1^0 \ \dots \ V_n^0]$ is considered. Current injections are considered to be step inputs, for node k : $0 \rightarrow I_k$ at $t = 0$. Hence, the Laplace transform of the current injection vector is $I(s) = (1/s) \cdot I$.

$$(s \cdot C + G) \cdot V(s) = \frac{1}{s} \cdot I + C \cdot V^0 \quad (3.69)$$

Mathematically the solution for the voltage in the s-domain is given as follows.

$$V(s) = (s \cdot C + G)^{-1} \cdot \left(\frac{1}{s} \cdot I + C \cdot V^0 \right) \quad (3.70)$$

According to the *initial value theorem*, the initial value of the solution for V in the time-domain is as follows. As expected it coincides with V^0 .

$$V|_{t=0^+} = \lim_{s \rightarrow \infty} s \cdot V(s) = \lim_{s \rightarrow \infty} \left[s \cdot \left(C + \frac{G}{s} \right)^{-1} \right] \cdot s \cdot \left(\frac{1}{s} \cdot I + C \cdot V^0 \right) = V^0 \quad (3.71)$$

According to the *final value theorem*, the final value of the solution for V in the time-domain is given in the following equation. As expected it only depends on the resistor grid and not on the capacitors. It is therefore established that the final solution will be what is expected from the solver.

$$V|_{t=\infty} = \lim_{s \rightarrow 0} s \cdot V(s) = \lim_{s \rightarrow 0} (s \cdot C + G)^{-1} \cdot (I + s \cdot C \cdot V^0) = G^{-1} \cdot I \quad (3.72)$$

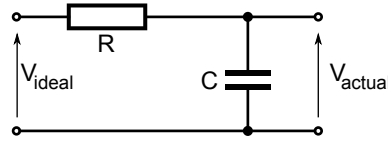


Figure 3.27 – An RC circuit in which R and C are in series.

The actual trajectory between $V^0 \rightarrow G^{-1} \cdot I$ depends on the actual time-domain solution of the above. This can be derived by using the inverse Laplace transform $\mathcal{L}^{-1}\{\cdot\}$ on (3.70). In reality this $V(t)$ solution is tedious to attain. Hence, an approximation of the mechanism is proposed.

An one-pole model, similar to a series RC circuit, is assumed for the transition. Such a circuit is shown on Fig. 3.27. V_{ideal} on the left of the figure is the “ideal” voltage imposed to the node by the collective effect of the resistor grid and voltages/currents of other nodes. V_{actual} is the actual voltage on the node that is emulated by the one-pole model. The transfer function of such a system is as follows.

$$H(S) = \frac{1}{\tau \cdot s + 1} \quad (3.73)$$

Where the RC constant is $\tau = R \cdot C$. The system is assumed to be initially in a steady state and the initial condition is $V_{actual}(t = 0) = V_{actual}^0 = V_{ideal}^0$. A step input for V_{ideal} is defined as follows.

$$V_{ideal}(t) = V_{ideal}^0 + (V_{ideal}^* - V_{ideal}^0) \cdot \mathbf{1}_{[0,\infty)}, \text{ where } \mathbf{1}_{[0,\infty)} = \begin{cases} 0 & \text{if } t < 0 \\ 1 & \text{if } t \geq 0 \end{cases} \quad (3.74)$$

The time domain output of V_{actual} is then

$$V_{actual}(t) = V_{ideal}^0 \cdot e^{-t/T} + V_{ideal}^* \cdot (1 - e^{-t/\tau}). \quad (3.75)$$

An ADC read command is issued at some time t_{rd} and the ADCs sample-and-hold the voltage at the exact time instant $V_{actual}(t_{rd})$. This sampled-and-held voltage is then converted to digital, and corresponds to the solution of the analog part for the electrical node in question. A similar model can be applied to all nodes of the system.

The mechanism described hereinabove introduces a tradeoff between the accuracy of the solution and the waiting time t . The longer the waiting time, the closer the FPPNS solution to

the true electronic solution, i.e. to the electronic steady-state. This speed-versus-accuracy tradeoff is a feature that can be readily exploited when/where required.

The above simplified model of (3.73)-(3.75) is of course inaccurate and should only be used as a rough reference when deciding on the waiting time between the new current injection and the ADC read command. The selection of this waiting time is actually a design choice that decides the well-functioning of the FPPNS.

It would be closer to reality to assume different RC constants τ_k for different nodes $k = 1 \dots n$, however assuming a system wide constant τ may be sufficient. This universal τ cannot be obtained analytically. Instead it can be determined by a series of empirical investigations, by measuring how “fast” the voltage value at the nodes of the system stabilizes to a final value. This empirically determined value is of course going to vary between different power systems, and different topological and value mappings of the same system. This is because all conductances and capacitances that are involved change in each case. Related results are presented in section 3.4.6.

3.3.2 Digital inaccuracy

There are two main inaccuracy sources in the digital part of the hardware. The use of fixed-point arithmetic and the inherent inaccuracies of the integration algorithms.

Fixed-point arithmetic

A fixed-point arithmetic is fully defined by a bit width and a scaling factor $\langle B, S \rangle$. Assume a positive real number $x \in \mathbb{R}^+$ without loss of generality. Negative numbers can be treated with any common signed number representation, e.g. signed magnitude, one's complement, two's complement, etc. Then the number is amenable to a fixed-point representation $\hat{x} = x_{int} \cdot S$, where $x_{int} \in \mathbb{Z}$ is an integer and $S \in \mathbb{R}^+$ is a scaling factor. If x_{int} is expressed in the binary numeral system, as it is done in modern digital computers, a sequence of B bits are used to represent it.

$$x_{int} = 0b \, b_B b_{B-1} \dots b_2 b_1 \quad (3.76)$$

Where b_i in the above are bits. The difference between the actual value of a number and its representation is the error introduced by the fixed-point representation.

$$e_x = x - \hat{x} \quad (3.77)$$

The minimum representable value by x_{int} is obviously zero, $\min(x_{int}) = 0$ - when all bits are equal to zero, and the maximum representable value is $\max(x_{int}) = 2^B - 1$ - for all bits equal to one. Scaled by S these correspond to a minimum and a maximum value for \hat{x} .

$$\begin{aligned}\min(\hat{x}) &= 0 \\ \max(\hat{x}) &= (2^B - 1) \cdot S\end{aligned}\tag{3.78}$$

If the number to be represented is outside the above limits, then it is not representable with the $\langle B, S \rangle$ fixed-point arithmetic, i.e. an *overflow* occurs. Either B or S have to be increased. The increase of B does not degrade the accuracy/granularity of the arithmetic, but comes at the cost of additional hardware resources. The increase of S is free from a hardware point of view, but results in coarser representations, i.e. increased errors. If none of the two measures are taken in the datapath, this results to information loss due to *truncation*.

Also, for a given $\langle B, S \rangle$, by construction the scaling factor is also the value resolution of \hat{x} ; since the minimum increase in x_{int} is one, then the minimum increase in x is $1 \cdot S$. Assuming a $x \in [\min(\hat{x}), \max(\hat{x})]$ then this value resolution introduces a *quantization* inaccuracy. If a well behaved $x \rightarrow \hat{x}$ mapping exists, for this inaccuracy it holds

$$|e_x| \leq \frac{S}{2}\tag{3.79}$$

This is so, since if the error exceeds $\pm S/2$, x should be mapped to the next or previous integer x_{int} value.

In the current implementation care has been taken to minimize truncation inaccuracies. Most datapath truncations occur in a dedicated truncation block - see “trunk” in Fig. 3.17. Minor truncations occur also elsewhere, but attention has been paid to minimize the probability of losing information due to truncation. This has been achieved by designing the pipelines by taking into account the range of the operands at each stage.

As for the *quantization* inaccuracies, if examined per se, of course they introduce errors in the fixed-point datapath. However, it should be taken into account that the digital part operates in conjunction with the analog part. The interface between the two is the array of analog-to-digital (ADC) and digital-to-analog (DAC) converters. By definition the latter only provide fixed-point representations of analog quantities, hence, the very first step of the computing pipeline forcibly uses fixed-point arithmetic. In the current implementation the bit length of this first stage is 12 bit (bit resolution of both ADCs and DACs).

This limit is actually the quantization accuracy bottleneck for the whole design. Every latter

Table 3.2 – Stages of inaccuracy of linear operations performed by the mixed-signal computer

Stage	Perturbed equation	Due to
a	$\tilde{G}_{el} = G_{el} + E_G$	inaccuracies in potentiometer admittances
b	$\tilde{I}_{el} = I_{el} + \epsilon_I$	inaccuracies in current DAC injections
c	$\tilde{V}_{el} = V_{el} + \epsilon_{V1}$	combined effect of stages (a) and (b) to voltage
d	$\tilde{V}'_{el} = \tilde{V}_{el} + \epsilon_{V2}$	inaccuracies in voltage ADC measurements

stage has been designed to be more granular than this first stage. Hence the effect of fixed-point quantization in the pipeline can be *neglected*.

Numerical integration algorithms

Another source of digital inaccuracy are the numerical integration algorithms. Every numerical integration algorithm, including the FE and AB2 which have been implemented in this work, only produces an approximation to the perfect solution of the DAEs/ODEs. The exact mechanism of the error, called *truncation error*, is briefly introduced in section C.3 of appendix C. This is the same whether the algorithm is implemented in conventional PC software or in the FPGA computational pipelines as described in section 3.1.2. In that sense the use of the FPPNS does not introduce any additional inaccuracy compared to standard power system analysis tools.

There is one point of interest however. As explained in the previous sections, there are many sources of digital and analog inaccuracy in the system. This raises concerns on the usability of the final results with respect to the integration algorithm alone - all other inaccuracy sources considered equal. A series of studies to investigate the phenomenon have been carried out [286, 287], and have confirmed this suspicion. Relevant results will be presented in section 3.4.

3.3.3 Effect on the mathematical operation

Individual inaccuracy components described in the previous sections contribute to the overall inaccuracy of operations performed by the FPPNS. This section focuses specially on the analysis of the linear system operation of (3.23). The analysis of the inaccuracy of other operations (e.g. matrix-vector multiplication) goes along the same lines.

Inaccuracy in linear system solving is manifested in four stages, as it is formalized in table 3.2.

Stage a) Inaccuracies of the conductances of the potentiometers that make up G_{el} result in an additive error E_G term in the building of matrix. Errors in individual entries of E_G are contributed by every potentiometer that is incident to the corresponding electrical node in question. Based on the assumptions of the previous subsection, these contributions follow the distribution of (3.42).

Stage b) Inaccuracies of the injected values of DACs result in the additive error term ϵ_I in the I_{el} injection vector. The distribution of (3.60) is followed for the individual components of ϵ_I .

Stage c) The imperfect injections \tilde{I}_{el} applied on the imperfect grid \tilde{G}_{el} induce voltages that are different from the ones that would occur in case the perfectly accurate I_{el} and G_{el} had been used. Stage c is exactly this inaccuracy to the voltage because of imperfect current injections and grid conductances.

$$\left. \begin{array}{l} \text{imperfect } V_{el} = G_{el}^{-1} \cdot I_{el} \\ \text{perfect } \tilde{V}_{el} = \tilde{G}_{el}^{-1} \cdot \tilde{I}_{el} \end{array} \right\} \quad \tilde{V}_{el} = V_{el} + \epsilon_{V1} \text{ definition of error} \quad (3.80)$$

Stage d) This last stage comprises of two parts. For this reason its effect can be broken up in two subparts.

$$\epsilon_{V2} = \epsilon_{V2a} + \epsilon_{V2b} \quad (3.81)$$

The first part concerns the fact that \tilde{V}_{el} is the steady-state voltage that would be induced to the resistor grid if I_{el} was applied. However as discussed in section 3.3.1, this steady state voltage is not instantaneously reached due to parasitic RC phenomena. This lag can be formalized by the introduction of an additive error ϵ_{V2a} into \tilde{V}_{el} . The longer the waiting time for the analog grid to settle, the smaller is the magnitude of this additional error term. If a τ system-wide RC constant is assumed, based on (3.75), ϵ_{V2a} can be approximated as follows.

$$\epsilon_{V2a} = \tilde{V}_{el}^* - \tilde{V}_{el}(t) = (\tilde{V}_{el}^* - \tilde{V}_{el}^0) \cdot e^{-\frac{t}{\tau}} \quad (3.82)$$

Where \tilde{V}_{el}^* is the final settling value of the electronics if they were allowed enough time to settle, \tilde{V}_{el}^0 is the initial analog steady state condition, and $\tilde{V}_{el}(t)$ is the time-dependent value that is finally read by the ADCs. It is seen from the above that as expected ϵ_{V2a} features an exponential decay with time.

Assume that at the time the ADC read command is issued the real voltage on the FPPNS is $V_{rd} = \tilde{V}_{el}(t)|_{t=t_{rd}} = \tilde{V}_{el} + \epsilon_{V2a}$. Then, the V_{rd} value cannot be read accurately, since only finite precision measurement devices are available. This introduces the additive error vector ϵ_{V2b} . Individual entries of ϵ_{V2b} follow the normal random distribution of (3.66).

The total absolute error of the solution of the analog solver compared to one provided by an

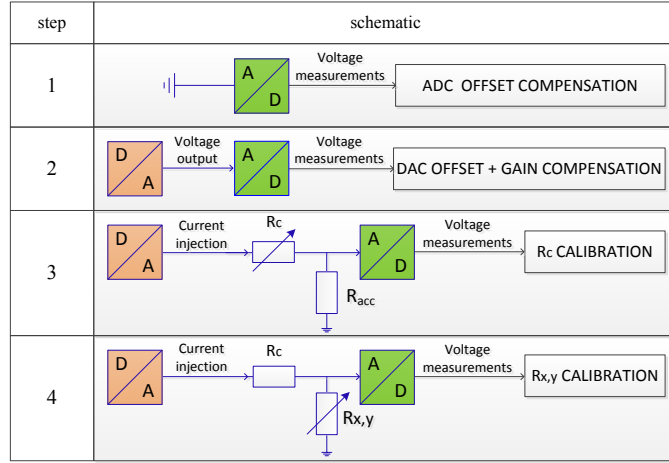


Figure 3.28 – Calibration procedure

infinite precision machine is as follows.

$$\epsilon_V := V_{el} - \tilde{V}'_{el} = \epsilon_{V1} + \underbrace{\epsilon_{V2a} + \epsilon_{V2b}}_{\epsilon_{V2}} \quad (3.83)$$

3.3.4 Calibration

A calibration system has been developed, for the analog part of the FPPNS [284]. The purpose of calibration is to improve the level of accuracy of the analog system. This translates into updated expected values and lower variances for many of the “unknown” inaccurate variables in the system, e.g. the full-scale resistance of the potentiometers (3.35), the combined inaccuracies of voltage DACs (3.51), the combined inaccuracies of voltage ADCs (3.61). The new expected values result in better tap setting calculations for potentiometers and voltage sources through (3.36) and (3.53). This combined with lower variances leads to smaller E_G , ϵ_I , ϵ_{V1} , ϵ_{V2b} in what has been detailed in the section 3.3.3. In turn this leads to a more accurate solution of the original problem.

The calibration sequence is fully automated in the sense that no user intervention is required. Hence, it can be carried out as often as required, for example due to drift of the analog parameters inherent to changes in temperature. It is realized using the probing/measurement resources that are already available on the boards of the prototype implementation. Fig. 3.28 illustrates the procedure. The steps described hereunder suffer naturally from the finite accuracy of the on-board measurement devices as well as from their quantized nature.

The process starts by correcting the offset of the ADC by connecting its input to the analog ground (step 1). This step reduces the collective effect of σ_{VFS} , σ_{FS} , σ_{mid} , σ_{INL} in (3.61). Indirectly this reduces the ϵ_{V2b} in step (d) of table 3.2.

Step 2 takes the calibrated ADCs as reference to correct the output characteristic of the DACs. This reduces the collective effect of σ_{FS} , σ_{DAC1} , σ_{DAC2} in (3.54). In turn this amelioration has a positive impact on the current injection mechanism through (3.59).

Step 3 calibrates the conversion resistors R_{conv} which is part of the current injection mechanism. For the conversion resistor of every node, an accurate voltage divider is build by putting R_{conv} in series to with a resistor of very high accuracy ($\pm 0.1\%$), R_{acc} in the figure. In this way $E[R_{conv}]$ is updated and $var[R_{conv}]$ is reduced, therefore the current injection through (3.59) is rendered more accurate. Steps 2 and 3 of the calibration process combined reduce the ϵ_I component of (b) of table 3.2.

Step 4 uses the already calibrated values of the conversion resistors to calibrate the potentiometers of the resistors network ($R_{(x,y)}$ in the figure). This step suffers from the naturally imperfect calibration of the conversion resistors in the previous step. Overall it reduces the effect of σ_R of (3.41) and in turn E_G in step (a) of table 3.2.

There are two comments to make regarding the sub-optimality of the current calibration procedure.

- Given that it utilizes measurement instruments that are available on-board, its correction capabilities are limited by the inherent inaccuracies of the instruments themselves (ADC inaccuracy). In the ideal case, calibration measurements would be provided by a dedicated (possibly external) measurement instrument.
- It assumes a linear model for the functioning of the DACs, which is only roughly true. The output of the DAC is essentially non-linear because of manufacturing integral and differential non-linearities, INL and DNL respectively. In the ideal case, a correction look-up table for all codes of every single DAC should be available. However, due to resource utilization considerations (limited memory of the digital core), this has not been implemented in the current prototype.

As it will be shown in section 3.4 calibration has a positive effect on the quality of the results of the FPPNS. Where relevant, “automatic” calibration will refer to the actual calibration on the prototype implementation. “Ideal” will refer to a calibrating procedure which involves external measurement devices and the use of (manually calculated) corrections for each DAC output code. “Ideal” calibration is a non-automatic (i.e. manual) procedure that counter-balances the two points mentioned hereinabove.

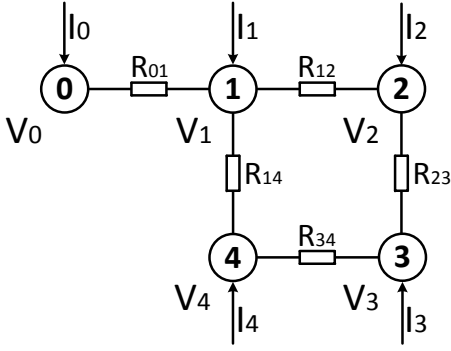


Figure 3.29 – Sample topology with 5 electrical branches and 5 electrical nodes

3.4 Results

3.4.1 Linear system solving

First, in order to demonstrate the effect of inaccurate electronics on the solution of a mathematical problem, the resistor topology of Fig. 3.29 is used as a case study (or sample) for non-calibrated electronics.

Every resistor in the grid is $R_{xy} = 2000 \Omega$, hence every conductance is $G_{xy} = 5 \cdot 10^{-4} S$. External measurements with a Fluke 45 multimeter (resistor measurement precision of $\pm 0.05\%$) showed the following discrepancies in the real value of the resistance of the potentiometers.

$$\begin{bmatrix} R_{01} \\ R_{12} \\ R_{23} \\ R_{34} \\ R_{14} \end{bmatrix} = \begin{bmatrix} 2000 \\ 2000 \\ 2000 \\ 2000 \\ 2000 \end{bmatrix} \Omega, \quad \begin{bmatrix} \tilde{R}_{01} \\ \tilde{R}_{12} \\ \tilde{R}_{23} \\ \tilde{R}_{34} \\ \tilde{R}_{14} \end{bmatrix} = \begin{bmatrix} 2096 \\ 2083 \\ 2047 \\ 2095 \\ 2059 \end{bmatrix} \Omega \quad (3.84)$$

The additive error introduced by the inaccuracies is

$$\epsilon_R = \begin{bmatrix} 96 \\ 83 \\ 47 \\ 95 \\ 59 \end{bmatrix} \Omega. \quad (3.85)$$

Substituting $R_{req} = 2000 \Omega$ into (3.44) yields an a priori estimate of the variance.

$$\text{var} [\epsilon_R] = (133.8 \Omega)^2 \quad (3.86)$$

This value is notably higher than the values of (3.85). This is because the 20% tolerance value given for the full-scale resistance of the potentiometers by the manufacturer is very much on the conservative side. Empirical tests of the potentiometers showed a tolerance of approximately 5%. This would result in a variance of $\sigma_R^2 = (166, \bar{6} \Omega)^2$. Substituting this to (3.43b) yields the following formula.

$$\text{var} [\epsilon_R] 2.7 \cdot 10^{-4} \cdot R_{req}^2 + 127.15 \quad (3.87)$$

In the latter substituting $R_{req} = 2000 \Omega$ yields a more realistic a priori estimate of the variance.

$$\text{var} [\epsilon_R] = (35.2 \Omega)^2 \quad (3.88)$$

The latter is closer to the values observed in (3.85).

The perfect conductance matrix of the resistor grid would be as follows.

$$G_{el} = \begin{bmatrix} 0.5 & -0.5 & 0 & 0 & 0 \\ -0.5 & 1.5 & -0.5 & 0 & -0.5 \\ 0 & -0.5 & 1 & -0.5 & 0 \\ 0 & 0 & -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0 & -0.5 & 1 \end{bmatrix} \cdot 10^{-3} S \quad (3.89)$$

Due to inaccuracies, it is actually

$$\tilde{G}_{el} = \begin{bmatrix} 0.477 & -0.477 & 0 & 0 & 0 \\ -0.477 & 1.443 & -0.48 & 0 & -0.486 \\ 0 & -0.48 & 0.969 & -0.489 & 0 \\ 0 & 0 & -0.489 & 0.966 & -0.477 \\ 0 & -0.486 & 0 & -0.477 & 0.963 \end{bmatrix} \cdot 10^{-3} S. \quad (3.90)$$

The additive error introduced by the inaccuracies is as follows.

$$E_G = \begin{bmatrix} -0.023 & 0.023 & 0 & 0 & \\ 0.023 & -0.057 & 0.02 & 0 & 0.014 \\ 0 & 0.02 & -0.031 & 0.011 & 0 \\ 0 & 0 & 0.011 & -0.034 & 0.023 \\ 0 & 0.014 & 0 & 0.023 & -0.037 \end{bmatrix} \cdot 10^{-3} S \quad (3.91)$$

Node 0 is kept as constant voltage, i.e. a voltage injection is performed. Nodes 1-4 are used as current injection points. The constant voltage node is used to set a reference node with known voltage magnitude and angle of $1\angle 0^\circ$. Hence the operation performed by the resistor grid is a mixed-current and voltage injection one, as described in (3.25)-(3.26). The voltage injection set is $\mathcal{J}_V = \{0\}$ and the current injection node set is $\mathcal{J}_I = \{1, 2, 3, 4\}$. The actual perfect matrix of the operation can be calculated from (3.25)-(3.26).

$$\begin{bmatrix} V_0 \\ I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = D \cdot \begin{bmatrix} I_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \quad (3.92)$$

Where,

$$D = \begin{bmatrix} 2 \cdot 10^6 & 1000 & 0 & 0 & 0 \\ -1000 & 1 & -0.5 & 0 & -0.5 \\ 0 & -0.5 & 1 & -0.5 & 0 \\ 0 & 0 & -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0 & -0.5 & 1 \end{bmatrix} \cdot 10^{-3} S. \quad (3.93)$$

Due to inaccuracies, the actual matrix used in the solver is as follows.

$$\tilde{D} = \begin{bmatrix} 2.096 \cdot 10^6 & 1000 & 0 & 0 & 0 \\ -1000 & 0.966 & -0.48 & 0 & -0.486 \\ 0 & -0.48 & 0.969 & -0.489 & 0 \\ 0 & 0 & -0.489 & 0.966 & -0.477 \\ 0 & -0.486 & 0 & -0.477 & 0.963 \end{bmatrix} \cdot 10^{-3} S \quad (3.94)$$

The voltages in nodes 1-4 can also be given by the following.

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = G_{el}[2:5, 2:5]^{-1} \cdot \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} \quad (3.95)$$

Also, as said, due to electronic inaccuracies, the injected quantities are not precise. The perfect and the actual vectors of the injections follow. Actual injections were measured using the same instrument as for the resistor measurement (voltage measurement precision of $\pm 0.025\%$).

$$\begin{bmatrix} V_0 \\ I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 70 \\ 110 \\ 35 \\ -125 \end{bmatrix} \begin{matrix} mV \\ \mu A \\ \mu A \\ \mu A \\ \mu A \end{matrix}, \quad \begin{bmatrix} \tilde{V}_0 \\ \tilde{I}_1 \\ \tilde{I}_2 \\ \tilde{I}_3 \\ \tilde{I}_4 \end{bmatrix} = \begin{bmatrix} 1.7 \\ 69 \\ 109.8 \\ 35.9 \\ -124.1 \end{bmatrix} \begin{matrix} mV \\ \mu A \\ \mu A \\ \mu A \\ \mu A \end{matrix} \quad (3.96)$$

The error vector is as follows.

$$\begin{bmatrix} \epsilon_{V0} \\ \epsilon_{I1} \\ \epsilon_{I2} \\ \epsilon_{I3} \\ \epsilon_{I4} \end{bmatrix} = \begin{bmatrix} 1.7 \\ -1 \\ -0.2 \\ 0.9 \\ 0.9 \end{bmatrix} \begin{matrix} mV \\ \mu A \\ \mu A \\ \mu A \\ \mu A \end{matrix} \quad (3.97)$$

Regarding the voltage injection at node 0, substituting $V_{req} = 2.5 \text{ V}$ in (3.56) yields an error for the a priori variance of the voltage injection³.

$$var[\mathbf{V}_{DAC}] = (2.8 \text{ mV})^2 \quad (3.98)$$

This is similar to the result of (3.97).

By disregarding the inaccuracy of the conversion resistor in (3.59), and considering a value

³According to (3.17) the ground voltage offset is at $V_{off} = 2.5 \text{ V}$ so a voltage DAC output of $V_{req} = 2.5 \text{ V}$ is required for a final voltage injection of 0 V

Chapter 3. Realized dedicated mixed signal solver

of $R_{conv} = 3125 \Omega$, the voltage DAC inaccuracy can be translated into a current injection inaccuracy (through (3.18)).

$$var[\mathbf{I}_{DAC}] = \frac{var[\mathbf{V}_{DAC}]}{R_{conv}^2} = (0.9 \mu A)^2 \quad (3.99)$$

Again, values of current inaccuracies observed in (3.97) are remarkably close to the theoretically expected values of (3.99).

The voltages of the nodes 1-4 can be determined from (3.95). The perfect solution and the solution expected due to the DAC and grid imprecisions are given hereunder.

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = G_{el}[2:5, 2:5]^{-1} \cdot \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} 180 \\ 317.5 \\ 235 \\ 82.5 \end{bmatrix} mV \quad \begin{bmatrix} \tilde{V}_1 \\ \tilde{V}_2 \\ \tilde{V}_3 \\ \tilde{V}_4 \end{bmatrix} = \tilde{G}_{el}[2:5, 2:5]^{-1} \cdot \begin{bmatrix} \tilde{I}_1 \\ \tilde{I}_2 \\ \tilde{I}_3 \\ \tilde{I}_4 \end{bmatrix} = \begin{bmatrix} 191.6 \\ 336 \\ 253.2 \\ 93.3 \end{bmatrix} mV \quad (3.100)$$

Then, the voltage error of stage (c) of table 3.2 is as follows.

$$\epsilon_{V1} = \begin{bmatrix} 11.6 \\ 18.5 \\ 18.2 \\ 10.8 \end{bmatrix} mV \quad (3.101)$$

Due to ADC imperfections, the exact \tilde{V} values cannot be measured with infinite accuracy. Instead, the following is provided as the solution by the analog solver.

$$\begin{bmatrix} \tilde{V}'_1 \\ \tilde{V}'_2 \\ \tilde{V}'_3 \\ \tilde{V}'_4 \end{bmatrix} = \begin{bmatrix} 188 \\ 333.3 \\ 250.2 \\ 89.1 \end{bmatrix} mV \quad (3.102)$$

The grid is allowed to completely settle for the measurement of the above, hence $\epsilon_{V2a} = 0$ in

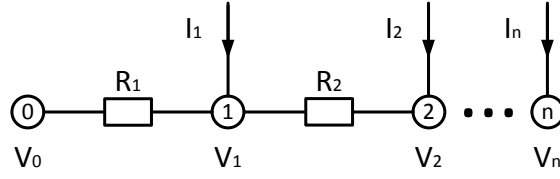


Figure 3.30 – Sample radial topology

(3.81), and the entire stage (d) is due to ADC inaccuracies.

$$\epsilon_{V2} = \epsilon_{V2b} = \begin{bmatrix} -3.6 \\ -2.75 \\ -3.04 \\ -4.19 \end{bmatrix} mV \quad (3.103)$$

The total absolute error in the voltage solution of the FPPNS is as per (3.83).

$$\epsilon_V = \begin{bmatrix} -8 \\ -15.8 \\ -15.2 \\ -6.6 \end{bmatrix} mV, \quad \frac{\epsilon_V}{V} = \begin{bmatrix} 4.44 \\ 4.98 \\ 6.47 \\ 8.00 \end{bmatrix} \% \quad (3.104)$$

3.4.2 Sample radial and meshed topologies

This subsection investigates the effect of the underlying topology graph on the error. Two topological archetypes are chosen, a purely radial topology, and a perfectly meshed topology, in an attempt to gain an insight on the propagation of the error. This insight is valuable as every real-world power system is composed by a set of radial and meshed parts.

Radial

A trivial radial architecture of $n + 1$ nodes is shown on Fig. 3.30. There is a voltage injection at node 0 (constant voltage) and current injections to nodes $1 \rightarrow n$. All injections (voltage and current) are considered inaccurate, and so are the resistor values. The voltage inaccuracy at node 0 is due to the inaccuracy of the injection, while at nodes $1 \rightarrow n$ is a result of the other inaccuracies.

An analytical expression for the voltage of node k is as follows.

$$\tilde{V}_k = \tilde{V}_0 + \sum_{i=1}^k \tilde{R}_i \sum_{j=i}^n \tilde{I}_j \quad (3.105)$$

Additive error models can be assumed for all resistors, voltages and currents.

$$\tilde{\mathbf{R}}_i = R_i + \epsilon_{\mathbf{R}i} \quad (3.106a)$$

$$\tilde{\mathbf{V}}_i = V_i + \epsilon_{\mathbf{V}i} \quad (3.106b)$$

$$\tilde{\mathbf{I}}_i = I_i + \epsilon_{\mathbf{I}i} \quad (3.106c)$$

The distribution of $\epsilon_{\mathbf{R}i}$ is given by (3.43), the distribution $\epsilon_{\mathbf{V}i}$ of is given by (3.55) and $\epsilon_{\mathbf{I}i}$ follows the distribution of (3.60). Based on the above an expression for the error part of \tilde{V}_k can be derived from (3.105).

$$\epsilon_{V_k} = \epsilon_{V0} + \sum_{i=1}^k R_i \sum_{j=i}^n \epsilon_{Ij} + \sum_{i=1}^k \epsilon_{Ri} \sum_{j=i}^n I_j + \sum_{i=1}^k \epsilon_{Ri} \sum_{j=i}^n \epsilon_{Ij} \quad (3.107)$$

Let σ_R , σ_V and σ_I be the variances of individual components of resistance, voltage and current injections respectively. Then an expression for the variance of ϵ_{V_k} is as follows.

$$\text{var}[\epsilon_{V_k}] = \sigma_V^2 + \sigma_I^2 \sum_{i=1}^k (n-i+1) \cdot R_i^2 + \sigma_R^2 \sum_{i=1}^k \left(\sum_{j=i}^n \right)^2 + \sigma_R^2 \cdot \sigma_I^2 \cdot \frac{k}{2} \cdot (2 \cdot n - k + 1) \quad (3.108)$$

From the above it can be seen that for an increasing distance k from the fixed voltage node the variance increases. Also for the same k (distance from the fixed voltage node) the variance is greater for larger n (total number of nodes). This is in accordance with common sense intuition that expects larger (radial) topologies to be more prone to uncertainty than smaller ones.

Meshed

A trivially meshed topology (single loop) is shown in Fig. 3.31.

The voltage of node k can be found by reaching the node through the traversal $0 \rightarrow 1 \rightarrow \dots \rightarrow k$.

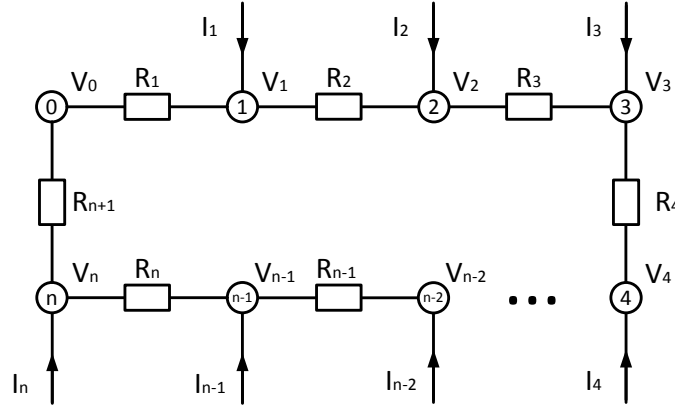


Figure 3.31 – Sample meshed topology

An analytical expression (including the inaccuracies) is as follows.

$$\tilde{V}_k = \tilde{V}_0 + \underbrace{\frac{1}{\sum_{q=1}^{n+1}} \sum_{j=1}^n \tilde{I}_j}_{k \text{ independent}} \underbrace{\sum_{r=j+1}^{n+1} \tilde{R}_r \sum_{i=1}^k \tilde{R}_i - \sum_{i=1}^k \tilde{R}_i \sum_{j=i}^{i-1} \tilde{I}_j}_{k \text{ dependent, } \propto k} \quad (3.109)$$

Alternatively, node k can be reached by traversing $0 \rightarrow n \rightarrow \dots \rightarrow k$. Then, the analytical expression for the voltage is analogously calculated.

$$\tilde{V}_k = \tilde{V}_0 + \underbrace{\frac{1}{\sum_{q=1}^{n+1}} \sum_{j=1}^n \tilde{I}_j \sum_{r=1}^j \tilde{R}_r}_{k \text{ independent}} \underbrace{\sum_{i=k+1}^{n+1} \tilde{R}_i - \sum_{i=k+1}^{n+1} \tilde{R}_i \sum_{j=i}^n \tilde{I}_j}_{k \text{ dependent, } \propto k} \quad (3.110)$$

Analytical derivation of the above is omitted for space economy. It is very easy to show that (3.109) and (3.110) are equivalent. An analysis in the lines of (3.107)-(3.108) is possible but too tedious to present here. Instead, few insightful remarks will be made.

- Unlike the radial case, in the meshed one the voltage of node k involves all resistors and currents in the topology, regardless of k ; see the k -independent underbraced quantity in both (3.109) and (3.110).
- The number of inaccurate terms in the sums increases the further we move away from node 0.
 - For nodes closer to 0 in the $0 \rightarrow 1 \rightarrow \dots \rightarrow k$ direction, the k -dependent underbraced quantity in (3.109) increases as k does so.

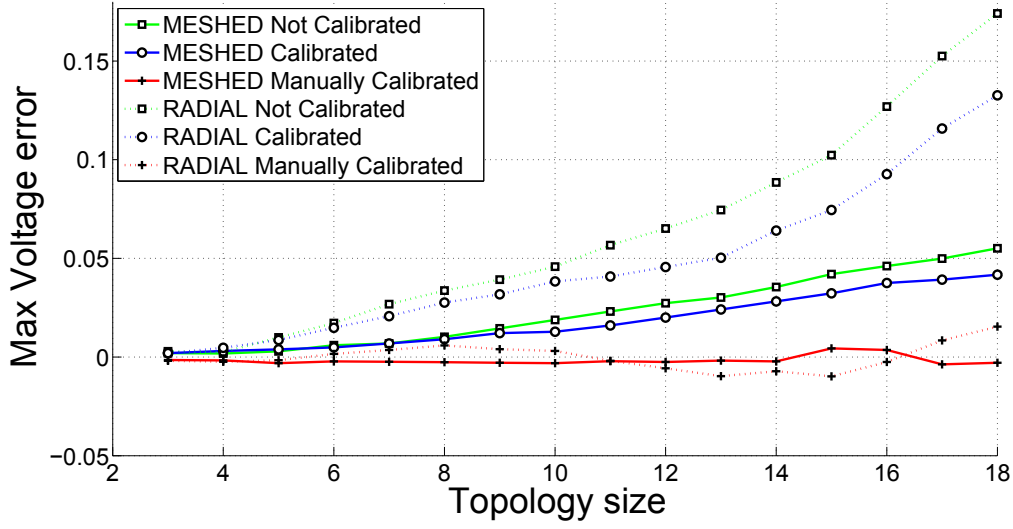


Figure 3.32 – Maximum absolute voltage error for radial and meshed topologies of increasing size

- For nodes closer to 0 in the $0 \rightarrow n \rightarrow \dots \rightarrow k$ direction, the k -dependent under-braced quantity in (3.110) increases as k does so.
- As in the radial case, for the same k , larger n translates in any case into more inaccurate terms in the sums of (3.109) and (3.110), hence the total variance is higher.

The above observations are illustrated with an example. A radial and a meshed topology with a single loop have been implemented and tested. In both cases, all branch resistances have been set to $R_k = 2000 \Omega$, $k = 1 \dots n$ and current injections to $I_k = 5 \mu A$. Topologies of size $n = 2 \dots 18$ have been created. Fig. 3.32 shows the maximum absolute error (across different k values) of (3.105) and (3.109)-(3.110) for different n values. The perfect values have been calculated in software using double-precision arithmetic. The curves correspond to the different states of calibration of the FPPNS.

In any case, the detrimental accuracy effect due to the size is clearly demonstrated. This is particularly true for the radial case, in which the node “furthest away” from node 0 is mostly affected. For an arbitrary topology, containing radial and meshed subparts, a quantitative conclusion cannot be a priori drawn but this general trend is expected to hold. The more “radial” a topology is, the more severe the inaccuracy possibly induced to the voltage results of the analog solver. This will be illustrated in a real world example in a following section.

Another conclusion that can be drawn from this subsection is the greatly beneficial effect calibration has on the quality of the results of the solver. As mentioned in section 3.3.4, calibration helps reducing the error of each individual component.

In the case of *automatic* calibration (with circular markers), trends similar to the non-calibrated case are observed. This can be explained by the fact that due to its inherent limited accuracy,

the measurement circuit (ADC) is unable to perfectly correct the DAC output characteristic – step 2 of Fig. 3.28. The effect of this residual error is accumulated for larger topologies.

In the case of *ideal* calibration, manual measurements of voltage quantities are made offline and are passed asynchronously to the digital core. Correction tables for the DACs that are involved in the topology are created and stored in external memory. In the *ideal* calibration inaccuracies are limited to the quantization limits and actual quantities are distributed randomly (± 1 bit) around the ideal value. Therefore, a node injecting a too high current value might be partly counterbalanced by a neighbor node injecting a smaller value. The effect of this is that an almost size-indifferent behavior of the analog solver is demonstrated.

3.4.3 Transient simulation

The FPPNS has been used for the transient simulation of power systems as per Fig. 3.5. Most tests have been performed on the systems `rc3`, `lanz5`, `fabre18`, `fabre36`, `fabre59`. Details and references on the test systems are given in appendix A.

Typical perturbation scenarios will be used for the 18- and 59-bus systems. A short-circuit in the topology can create some serious power imbalances, i.e. non-zero $P_m - P_e$ terms in (3.30), which in turn will initiate a dynamic response of the angles. This response has been calculated by the FPPNS, as well as by a reference software running on a conventional PC [296]. In the current FPPNS-elab-tsaot implementation, all machine dynamic variables, as well as any voltage and current injection in the system can be asked as an output result. The results are stored in the `TD results` bank of the software and can be visualized in the `Analysis editor`.

The fully digital implementation (notation identifier “sim”) of [296] will be used as a benchmark reference. Difference of the angle trajectories will be used as a metric to measure the quality of the results of the dedicated hardware. Results with no calibration (notation identifier “nc”) and with automatic calibration (notation identifier “ac”) will be presented. For this application, ideal calibration is not available, since the current digital part of the solver lacks the memory necessary to store calibration corrections for the entire voltage output range.

Fig. 3.33 shows the rotor angle oscillations for one generator (gen. #3) in the 18-bus topology. The scenario is a short-circuit at the middle of the branch connecting buses #3 and #4, active for 70ms, with a post-fault tripping of the line.

The accuracy of the solver $|\delta_3^{\text{sim}} - \delta_3^{\text{nc}}|$ and $|\delta_3^{\text{sim}} - \delta_3^{\text{ac}}|$ is less than two degrees for the entire simulation window. For most applications this margin is very well tolerable. The high accuracy of the solver can be explained partly by the small size of the topology and partly by the fact that the topology is highly meshed, as discussed in section 3.4.2. The effect of automatic calibration is hardly noticeable. This is because, as mentioned in section 3.3.4, the on-board converters apparently cannot measure, and therefore correct, beyond a certain point of accuracy; this point is the residual inaccuracy shown in the figure.

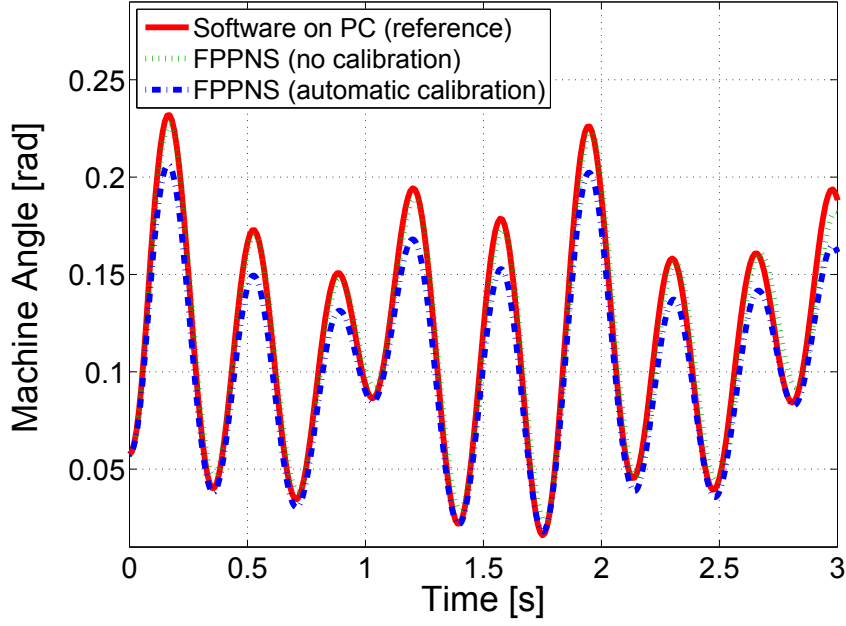


Figure 3.33 – Rotor angle oscillations of generator #3 of the 18-bus topology using different simulators

The same flow of operations is performed in the 59-bus topology. The scenario is a short-circuit at the middle of the branch connecting buses #9 and #10, active for 200ms, without a post-fault tripping of the line, i.e. the fault disappears while the line stays connected. Figs. 3.34 and 3.35 show the rotor angle oscillations for generators #11 and #54. A very interesting phenomenon is observed.

While the angle of gen. #11 is highly accurate ($|\delta_3^{\text{sim}} - \delta_3^{\text{nc}}|, |\delta_3^{\text{sim}} - \delta_3^{\text{ac}}| \leq 1^\circ \forall t$), the one for gen #54 is not so. This is one case in which calibration significantly increases the quality of the results, as demonstrated by the following.

$$|\delta_{54}^{\text{sim}}(t) - \delta_{54}^{\text{nc}}| \leq 7.6^\circ \forall t \quad (3.111a)$$

$$|\delta_{54}^{\text{sim}}(t) - \delta_{54}^{\text{ac}}| \leq 2.6^\circ \forall t \quad (3.111b)$$

Bad accuracy for generator #54 can be partly explained by examining the topology architecture and by recalling what has been discussed in section 3.4.2. Fig. 3.36 contains a simplified view of the 57-bus topology. It shows that this system can be split in two sub-networks with a single radial connection. According to what was stated in section 3.4.2, the voltage results of sub-network 2 are expected to be less accurate than the ones of sub-network 1, as table 3.3 attests. It shows the maximum absolute angle deviation for all synchronous machines of this topology. The impact of the calibration as well as the accuracy dependence on the concerned sub-network can be clearly seen.

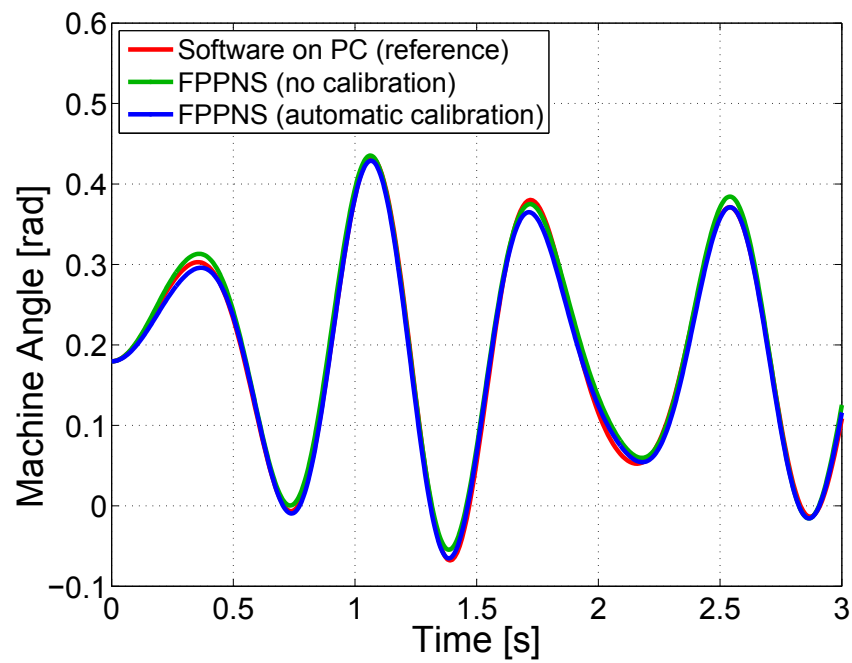


Figure 3.34 – Rotor angle oscillations of generator #11 of the 59-bus topology using different simulators

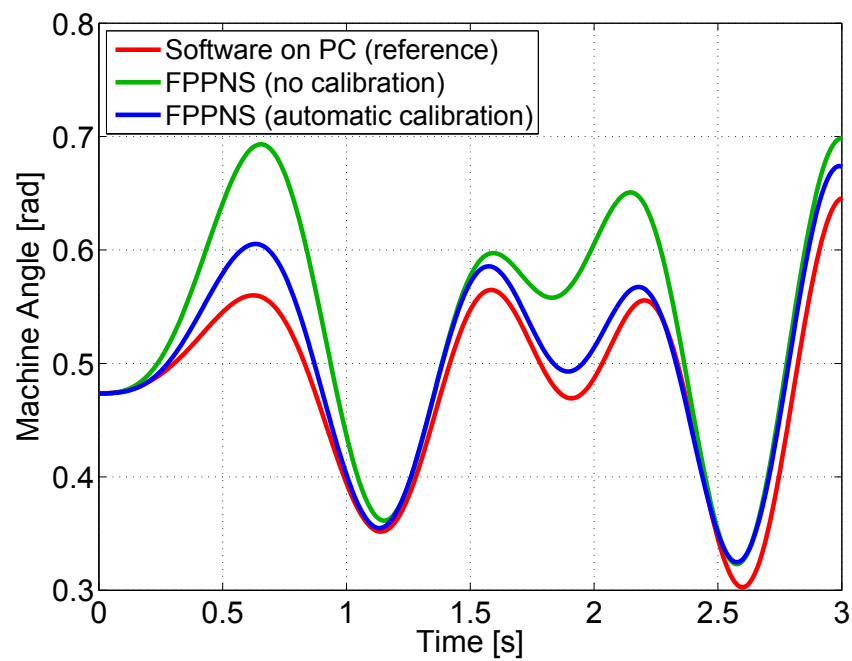


Figure 3.35 – Rotor angle oscillations of generator #54 of the 59-bus topology using different simulators

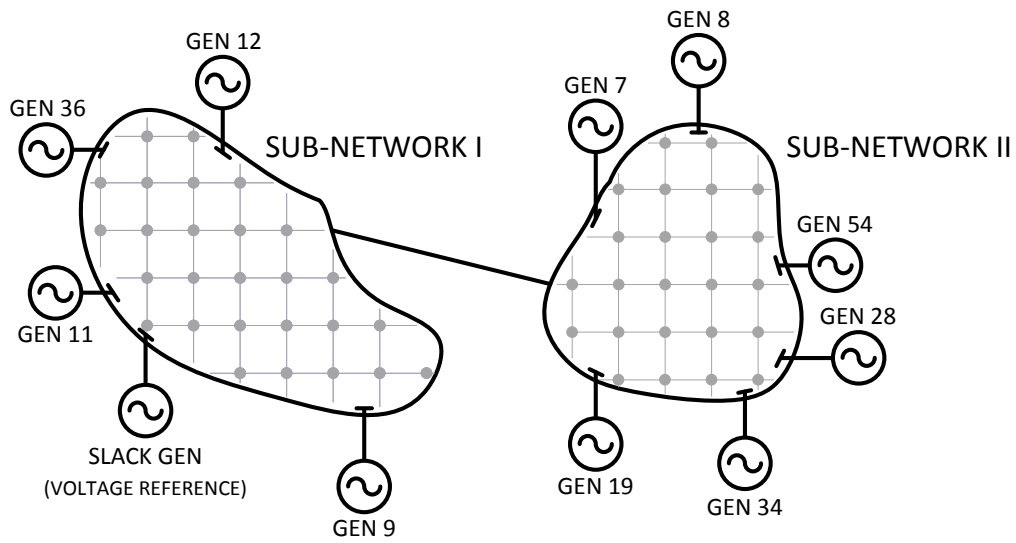


Figure 3.36 – Reduced schematic of the 59-bus test case

Table 3.3 – 59-bus test case maximum absolute rotor angle deviation for TSA

Generator #	Sub-network	Angle nc [°]	Angle ac [°]
7	2	5.5	2.1
8	2	5.8	1.9
9	1	0.5	1.1
11	1	0.6	0.4
12	1	0.7	0.4
19	2	4.9	1.6
28	2	5.7	1.8
34	2	5.4	1.7
36	1	1.9	0.4
54	2	7.6	2.6

nc: refers to non-calibrated hardware

ac: refers to auto-calibrated hardware

Generally transient simulation results provided by the solver have tolerable levels of inaccuracy. A factor that smooths out part of the overall analog inaccuracy is locality. The power mismatch of (3.30) depends on voltage differences between that bus and the direct-neighbor bus(es). This local relative inaccuracy is expected to be lower than the absolute system-wide one. Therefore it yields results that are closer to the correct value than expected.

Power consumption

The standard average consumption of one PCB is approximately 5.85 W (for steady voltage and current values of 6.5 V and 0.9 A), yielding a total of approximately 23.4 W for the entire platform. This figure compares favorably with consumptions of other dedicated accelerator hardware such as GPUs, e.g. 49W(max) for a low-range desktop ATI Radeon X800 GTO, and 250 W for a high-end NVIDIA GeForce GTX TITAN X. An exact watt per FLOPS/Joule is difficult to attain, since it is difficult to quantify the operation of the analog part in floating point operations.

Timing breakup

A timing breakup of the calculation is presented in Fig. 3.37. It is compared to a research grade software (MatDyn [296]) and to an industrial grade software (Ramses [6]) in table 3.4.

Time profiling of the FPPNS starts at the time when all initialization data have been written to the 2-port RAM of Fig. 3.18, and stops when the results have been written back to that memory. This is honest in the sense that this RAM is the interface of the FPPNS-accelerator to the `elab-tsaot` software. Time spent on the hardware platform can be divided in three steps: pre-processing, processing and post-processing. The first part is dedicated to the initialization of the digital and analog resources. The second one is dedicated to calculations of the partitioned scheme of Fig. 3.5 on the pipelines. The timing break-up of each iteration is according to Fig. 3.19. Within the processing time, time to reconfigure the grid according to the predefined fault is also accounted for (“Fault on” and “Fault off”). The post-processing stage extracts the results and writes them in the shared memory. As it can be seen, a significant percentage of the over-all time is spent on auxiliary tasks, and not specifically on the calculations (pre- and post-processing).

The program flow of MatDyn is similar to the one used implicitly in the FPPNS. A partitioned approach is used to solve the set of DAEs for the TD simulation. Classical generator models have been used for all generators without exciters or governors. A predictor-corrector Euler method has been used for numerical integration since it has similar complexity and properties to the FE/AB2 methods used in the FPPNS.

RAMSES uses a simultaneous scheme to solve the simulation equations. Afterwards an implicit integration scheme is applied, and the resulting non-linear system of equations is solved using a dishonest Newton method. This flow is significantly different as compared to the one used

0.15 ms		4.26 ms						5.64 ms
Pre-processing		Processing						Post-processing
152 us		4106 us						1380 us
RAM to processing units	Grid settings	Start	Fault on	Computation	Fault off	Computation	Stop	Processing units to RAM
117 us	35 us	40 us	40 us	274 us	40 us	3832 us	~0 us	1380 us

Figure 3.37 – Timing breakup of computation time for transient simulation of the 18-bus

Table 3.4 – Speed comparison between different engines for the TD simulation of the 18-bus system

Tool	Computation time (ms)
MatDyn (academic sw)	~ 360
RAMSES (industrial sw)	~ 24
FPPNS	5.6

in FPPNS and MatDyn, however results of the comparison are included for completeness sake.

A time that does not appear in Fig. 3.37 is the time used to communicate between the dedicated hardware and a host PC. Table 3.5 shows the timing breakup between the different platforms, for the time domain simulation that has been presented in this section. Operations on the PC involve all preconditioning actions that take place in the `elab-tsaot` software. These are mainly auxiliary tasks, as shown in Fig. 3.23. USB timings in the table comprise the entire communication cycle. Normally for a standard FPPNS operation there are two stages of writing to the shared RAM.

- Write the bitstream to the FPPNS slices. This bitstream contains information on the topology to be mapped on the analog hardware and on the scenarios to be executed. In the current implementation, each slice has to be written separately.
- Issue the start commands that correspond to the analysis that is to be executed. These commands also contain generic information on the simulation, e.g. time step, duration, etc.

The FPPNS part of table 3.5 comprises the computation of Fig. 3.37. After the FPPNS computation is finished, the results are available in the internal memory of the FPGA. They can be retrieved by the `elab-tsaot` by issuing specific request-commands through the USB interface.

As seen, in the current implementation the USB communication is a bottleneck that can have a very detrimental effect to the overall throughput performance of the platform. This motivates the use of it in a serial “batch” execution of analyses/scenarios as will be shown in the following section [283].

Table 3.5 – Timing break-up between PC, USB communication and the FPPNS for a transient stability operation

	Time
PC (elab-tsaot)	0.5 ms
USB comm	152.79 + 8.99 ms
FPPNS	5.64 ms

Table 3.6 – Timing results summary for the n-1 branch contingency analysis

System size	Branches examined	Time sim [s]	Time FPPNS [s]	Speedup
5	5	0.9	0.23	3.9x
18	20	10.2	0.5	20.4x
36	39	106.7	0.92	116x
59	55	434.9	1.13	384.9x

3.4.4 Dynamic stability analysis

Dynamic Stability Assessment (DSA) is an analysis concerned with the quantitative/qualitative characterization of the ability of the system to retain a state of operating equilibrium after being subjected to severe disturbances. The transient events examined during DSA studies, are handled using TD simulations.

Branch n-1 contingency analysis

A common procedure in DSA studies is n-1 contingency analysis. In the latter, the operator is interested in knowing whether the system (all generators) retains or not its stability, after an outage/perturbation is applied on each one of its elements. The outcome of this study, is a set of simple boolean answers (stable/unstable) for each of the contingencies in the test set. The answer from the hardware platform can therefore be communicated with a simple boolean statement for each of the scenarios (stable/unstable). This greatly reduces the communication requirements.

A variation of this procedure was applied to different test systems. A set of branches was selected to be examined and perfect 3ϕ faults were applied in the middle of each branch for 200 ms. Table 3.6 summarizes the timing results for the software-only (sim) and the mixed-platform (FPPNS) solution. The numbers given for the timings include the time for the USB communication with the FPPNS. The software implementation of the n-1 DSA module is based on the TD engine detailed in section 3.2.1.

Contingency analysis results can be visualized in the Analysis editor as shown in the screenshot of Fig. 3.38. The results of a n-1 analysis for the 18-bus system are shown with branch “D0” is selected (in dashed blue). Generators that lose their stability in case of a fault

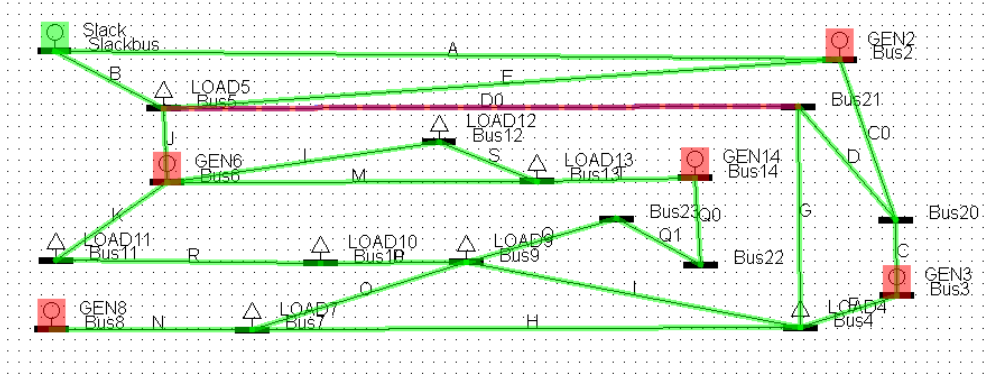


Figure 3.38 – Screenshot of the Analysis editor of the elab-tsaot visualizing results for a n-1 branch contingency analysis on an 18-bus system

Table 3.7 – Timing results summary for the CCT analysis

System size	Branches examined	TD runs	Time sim [s]	Time FPPNS [s]	Speedup
5	5	40	7.2	1.13	6.4x
18	20	160	95.2	3.39	28.1x
36	39	312	676.8	4.64	145.9x
59	55	440	4354.5	11.4	383.3x

on the selected branch are highlighted in red, while the ones that retain their stability are shown in green.

Critical Clearing Time analysis

The Critical Clearing Time (CCT) analysis determines the maximum duration of a fault on a given branch for which the system (even marginally) maintains its stability. Naturally, CCT depends on TD simulations. The most common algorithm to do it is to perform transient simulations in a binary search fashion for different fault durations. The CCT value is approximated by an upper and a lower bound, the difference of which needs to be under an asked-for precision.

A CCT algorithm similar to the one described here, was implemented on the FPGA. Perfect fugitive 3ϕ faults in the middle of the branches were considered. A binary search was performed for each fault location, in the search window $t_{src} \in [0.0s, 1.5s]$, and a precision of $t_{prec} = 10ms$ was requested. Timing results for the software-only (sim) and the mixed-platform (FPPNS) architectures are summarized in table 3.7. The software-only implementation of the CCT DSA module is based on the TD engine detailed in section 3.2.1.

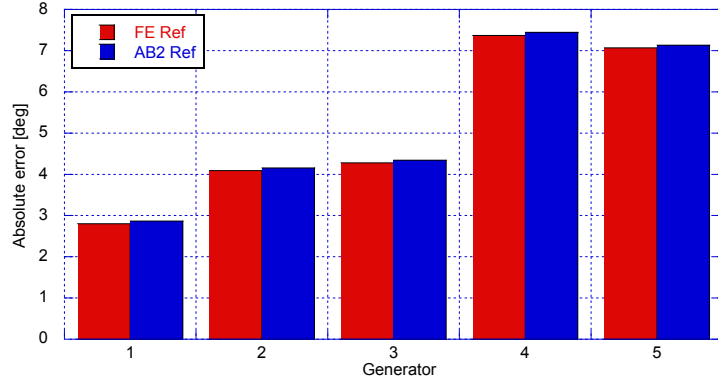


Figure 3.39 – Minimal time-step FPPNS error compared to PC software simulation reference for the 18-bus test system

3.4.5 Effect of the integration algorithm on the results

The integration algorithms that are synthesized in the fixed-point datapaths of the pipelines in the FPGA reside in a low accuracy environment, as discussed in section 3.3. This way their inherent numerical inaccuracy is exacerbated. The purpose of this section is to investigate the effect of different integration algorithms on the quality and the usability of the results. Related studies have been performed on the `fabre18` test system.

Assessment of absolute precision for different time-steps

A metric to quantify the relative precision of two simulators is the maximum absolute difference in the trajectories of the internal angle of the same generator, $L_i^\infty = \max_t |\Delta \delta_i(t)|$.

Fig. 3.39 presents L_i^∞ for all generators in the test system. Simulations are run on the emulator using a time-step $h = 60 \mu s$, for both FE and AB2. This value is orders of magnitude less than common practice in TRANSCOs [297]. Before the FPPNS run, the calibration procedure of section 3.3.4 was carried out to ensure minimum relative inaccuracy. Results coming from a PC software 4th-order Runge-Kutta (RK4) implementation were used as a “angle solution” reference. Discrepancies between the best possible analog solution and the software reference can be attributed exclusively to analog imprecision.

As the time step of the hardware emulator increases, its accuracy decreases. This deterioration is due to the effect of digital imprecisions that increase with the increase of the time step. Fig. 3.40 shows this added inaccuracy against the minimum timestep case that is taken as a relative reference. L_i^∞ for generators are displayed in groups of time steps. Naturally, as higher time steps are used the accuracy decreases. However this decrease in the quality, and therefore usability, of the results is less pronounced for AB2 than for FE. For example, using a timestep of 15.6 ms yields acceptable results for AB2, while FE collapses. In cases of very high time step (e.g. 62.5 msec), both algorithms collapse and a quantitative assessment of L_i^∞ is irrelevant

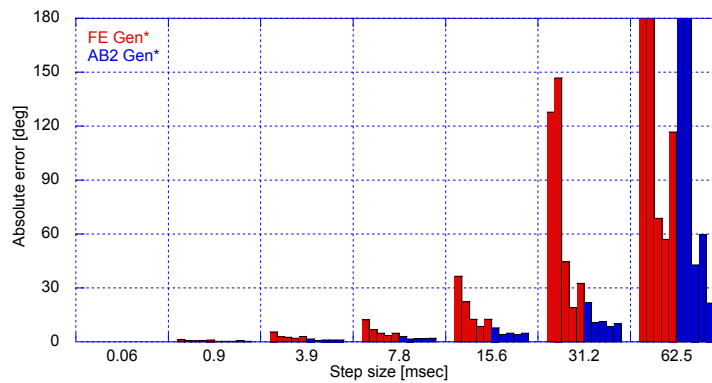


Figure 3.40 – FPPNS error due to digital imprecision for different time steps for the 18-bus test system

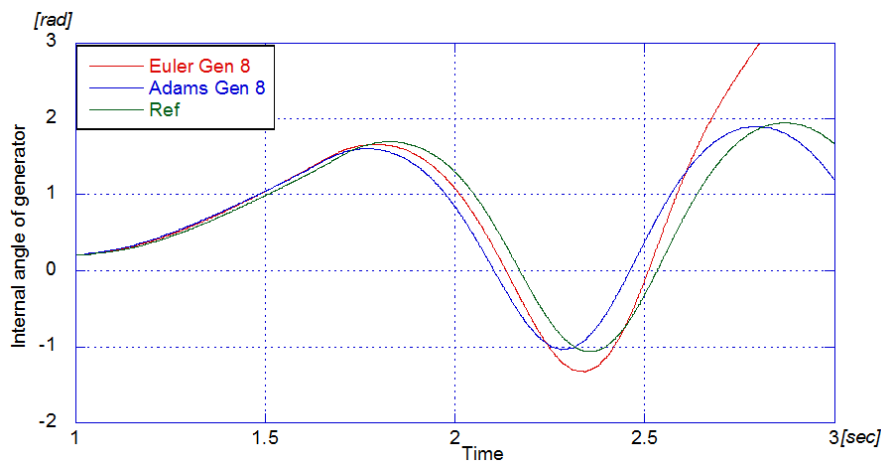


Figure 3.41 – FE instability while AB2 succeeds in retaining the stability of the numerical solution

A selected case where the contrasting behavior of FE and AB2 is well illustrated is shown on Fig. 3.41. In it a detailed view on the trajectory of the internal machine angle of a generator after a perturbation is shown. The differences on the amplitude and the exact time instant of the first peak of the trajectory are quantifying the global accumulated truncation error of the algorithms. It is clear that the AB2 trajectory is way closer to the “real” one coming from RK4 on software. As the phenomenon evolves, local truncation errors further accumulate in the FE case, and finally render the trajectory unstable after the second peak. It is crucial to note, that this instability is a numerical artifact of the integration algorithm (FE), and it is not owed to a real-life phenomenon. This clearly demonstrates that such results given by FE are unusable and potentially dangerous for real world operation analyses. This issue is further analyzed in the following subsection.

Table 3.8 – n-1 branch contingency results for different integration algorithms and time steps

contingency	sim	$h = 7.8 \text{ ms}$		$h = 15.6 \text{ ms}$		$h = 31.3 \text{ ms}$	
		FE	AB2	FE	AB2	FE	AB2
250ms @br#1	S	S	S	U	S	U	S
250ms @br#6	S	U	S	U	S	U	S
250ms @br#10	S	S	S	S	S	U	S
250ms @br#13	S	S	S	U	S	U	S
250ms @br#17	S	S	S	U	S	U	S
250ms @br#20	S	S	S	S	S	U	S
250ms @br#31	S	S	S	U	S	U	S
410ms @br#32	U	U	U	U	U	U	U
250ms @br#33	S	S	S	U	S	U	S

Effect on n-1 branch contingency analysis

Table 3.8 summarizes the results of a n-1 contingency analysis for the test system, for a selected set of branch contingencies. Under the “sim” column reference results coming from a purely software RK4 simulator are shown. It is clearly seen that as the time steps become larger ($a \rightarrow b \rightarrow c$) FE implementations fail to be in accord with the reference, while AB2 implementations succeed in doing so. Boolean results in bold typeface show this mismatch between FE results and the reference.

Effect on Critical Clearing Time analysis

Potentially erroneous results for CCT analysis can be critical for the safety of the system, since many of the coordinated protection schemes rely on it. Fig. 3.42 presents CCT results for branches #4, #10 and #30. It is clear from the graphs that AB2 is able to provide reasonable results ($\pm 2.2\%$ of the minimum step reference) with time steps as high as $\sim 16 \text{ ms}$. On the contrary, for FE results to be in the same range, time step sizes of $\leq 2 \text{ ms}$ have to be used. Since CCT analysis is a repetitive procedure due to the binary search scheme (~ 90 transient simulations for each point of the figures for 10 ms CCT precision), higher time steps result in considerable computational savings.

This phenomenon becomes even more pronounced when the accuracy of the environment surrounding the numerical integration blocks is further reduced. A related set of tests were performed for the power system, this time without the calibration of the analog part. This makes the approximation of the solution of the linear system even worse, hence reducing the overall accuracy of the solver. Fig. 3.43 presents relevant result for branch #30, for the two algorithms. It is clear that AB2 is far less affected by the drop of accuracy of the linear solution compared to FE. For the latter, we observe results that are off by a margin of 8% compared to, for time steps as low as 4 ms.

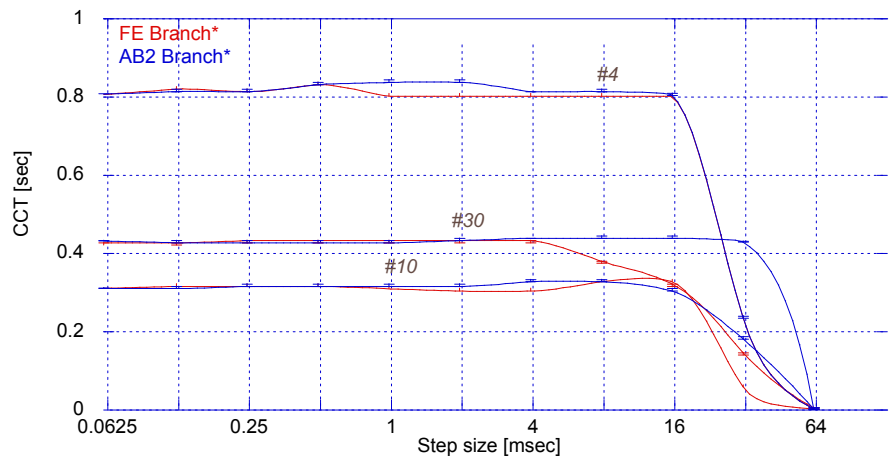


Figure 3.42 – CCT for branches #4, #10, #30 of the 18-bus system with varying timesteps using FE & AB2

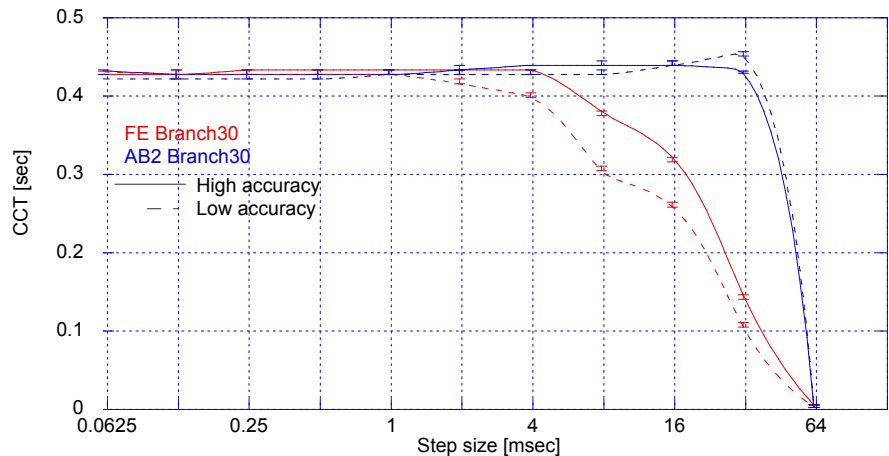


Figure 3.43 – CCT for branch #30 of the 18-bus system with varying timesteps using FE & AB2, in a calibrated and an non-calibrated FPPNS environment

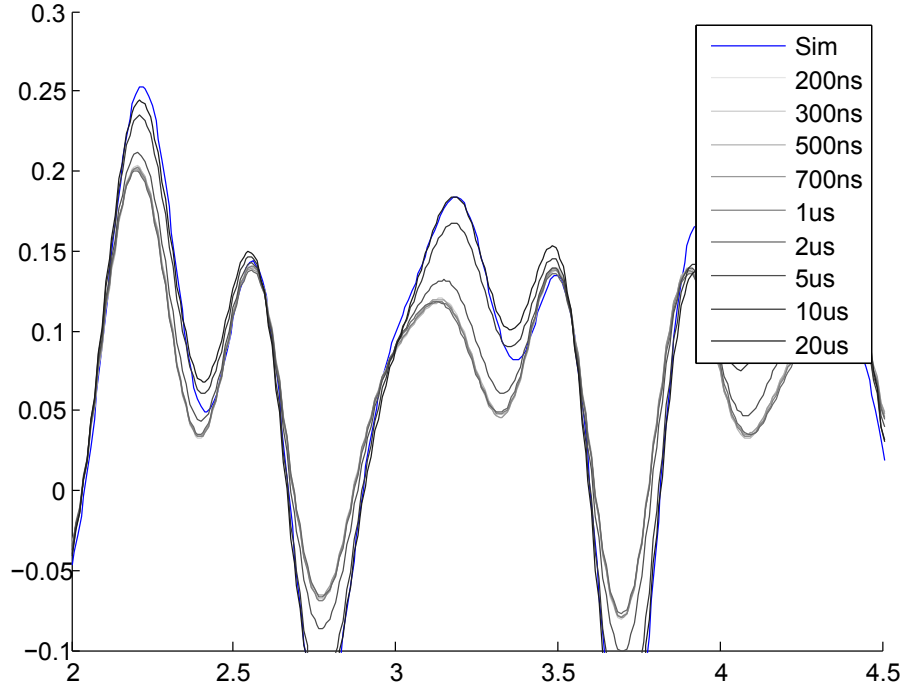


Figure 3.44 – Internal angle of generator #4 after a transient event in the 18-bus case, for different ADC waiting times.

3.4.6 Effect of time step versus waiting time

The expression for the error on the final voltage due to the parasitic RC capacitance (3.82) can be rewritten as follows.

$$\epsilon_{V2a} = \epsilon_{V2a}(\Delta \tilde{V}_{el}, t_{rd}) = \Delta \tilde{V}_{el} \cdot e^{-\frac{t_{rd}}{\tau}} \quad (3.112)$$

Where $\Delta \tilde{V}_{el} := \tilde{V}_{el}^* - \tilde{V}_{el}^0$ is the ideal voltage change induced by the updated current/injections of the new step. It is seen that the magnitude of ϵ_{V2a} is directly proportional to the magnitude of this voltage change. The latter is expected to be larger for larger time steps, regardless of the numerical integration algorithm that is used. This is easily understood since a larger time step, allows the dynamics of the system to manifest for longer, hence the state of the system is moved further away compared to a smaller time step. This relation $h \propto \epsilon_{V2a}$ might require a longer waiting t_{rd} to compensate, i.e. for larger time steps the analog grid might require longer time to settle.

Relevant tests have been conducted for the 18-bus system. Fig. 3.44 shows the internal angle of generator #4 for a 3ph fault scenario. The blue line is the true angle, obtained with a software simulator, while grey lines correspond to the solutions given by FPPNS for different waiting times t_{rd} .

It is clearly seen that the longer the waiting time, the closer the FPPNS result is to the true value. This is particularly evident in parts of the graph where there are peaks (and valleys). Peaks are smaller for shorter waiting times. During peaks $\Delta \tilde{V}_{el}$ variations are large, and through (3.112) larger errors are introduced to the voltage solutions. The shorter the waiting times, the more difficult for the FPPNS solution to follow the true one. Empirical studies show that $t_{rd} = 1 \mu s$ is a good trade off between speed and accuracy in most of the cases that have been tried.

3.5 Conclusions

To the knowledge of the author the FPPNS presented in this chapter is the most advanced mixed-signal power system computer. Table 3.9 provides an overview of the existing prototype based on the criteria established in section 3.1.1. All figures given in the table concern a research-grade prototype that has been realized using off-the-shelf discrete components as a proof of concept. Significant improvements can be expected in more optimized implementations, such as the one involving customized integrated circuits, as suggested in [4].

3.5.1 Comparison with related work

If only the linear algebra capabilities of the FPPNS is concerned, related work has been presented in section 2.4.1. Most of the efforts presented in that section however are conceptual and provide no implementation of the proposed system, unlike the work presented in this chapter. Additionally, many of the studies deal with the solution of linear systems that arise in the finite difference solution of partial differential equations which result in matrices of fixed specific structures. Of course our work is suitable for the solution of such linear systems. It goes well beyond that, in offering complete reconfigurability of the resistor network, so that an arbitrary topology (i.e. matrix structure) can be mapped onto it. In our approach the analog part performs the linear algebra operation in one-shot, in no iterative scheme. Hence the inaccuracies of the electronics have a direct impact of the accuracy of the precision of the operation. An alternative has been proposed in [241, 240, 242, 245] to use *iterative refinement* techniques in a effort to decrease the negative impact of analog imprecision to the final solution. The drawback of this approach is that iterative solutions (using the residuals) of the same linear system are required, hence the performance is decreased. Recent FPAA-based work is in very experimental stage as it concerns only dense algebra operations of system sizes up to 2 [248]. In our current implementation matrices of size up to 2×96 can be handled.

Works related to analog-computing for power system studies have been presented in section 2.4.2. The work Fried [257] gave birth to research primarily centered around two universities, EPFL and Drexel University [271, 268, 273, 269, 270]. Work of the two research groups can be compared along the points of section 2.4.3, wherever relevant data is available.

Table 3.9 – Characterization of the existing platform

Criterion	Remarks
Cost	The cost of all the discrete components that make up each one of the prototype PCBs is in the order of $5k$ CHF, yielding a total cost of $20k$ CHF ($4 \times 5k$).
Size	The entire computer is self-contained in a $32 \times 26 \times 10$ cm case. Power supply and communication connections are provided.
Interfacing	A USB connection is provided to the platform. A shared memory is available both to the mixed-signal computer central processing unit (a synthesized NIOS II CPU) and to the CPU of the conventional host computer (through dedicated drivers).
Reconfigurability	The topology of the resistor network is a fixed 5-degree lattice, the edges of which can be enabled/disabled through the use of digitally controlled switches. The value of all resistors, current and voltage sources are reconfigurable (even in runtime) according to their respective bit resolution.
Scalability	A total capacity of 96 power system nodes (4×24) is provided. Systems up to 57 buses have been successfully analyzed. There is evidence that the favorable linear algebra computing properties scale well with system size, but a detailed investigation is pending.
Performance	Against research grade solvers, speeds of up to three orders of magnitude have been achieved in dynamic stability assessment [283]. Communication time between the platform and the conventional host PC is a main bottleneck, and hence performance benefits are maximized when analyses are performed in series.
Accuracy	The main source of accuracy errors are the inaccurate analog electronics and the quantized discrete values they can assume (quantization). An automated calibration scheme has been developed in order to ameliorate the phenomenon [284]. In total, an average relative error of $<6\%$ has been estimated and observed for the linear system solution.
Power efficiency	The standard average consumption of one PCB is approximately 5.85 W (for steady voltage and current values of 6.5 V and 0.9 A), yielding a total of approximately 23.4 W for the entire platform.
Functional completeness	On the analog part, only linear algebra operations are supported, by injecting currents and measuring voltages (see also section 3.1.1). The fixed predefined topology of the available (positive resistance) potentiometer bank creates limitations on the nature and value range of admittance matrices that can be mapped on the FPPNS.

Scalability and Accuracy. The largest network that has been presented was one of 5 nodes and 2 machines in [273]. There are claims of potential to scale up to thousands, however the precision of the results is already questionably acceptable for the 5 bus case (fact that the authors of [273] acknowledge in section VII). The current capacity of the FPPNS is 96 buses, and networks up to 59 buses have been realized and tested while retaining acceptable precision for the results.

Reconfigurability. In [273] the topology of the power network is connected by manually wiring analog boards of fixed sub-topology. It is difficult to imagine how a system of more than a few tens of buses could possibly be mapped manually to an array of FPAA boards by manually wiring the connections. Additionally, the FPAAs that are used are configured once and retain their configuration for the rest of the study. On the contrary our system has complete runtime reconfigurability. The topology of the power system is automatically mapped to a suitable resistor network and any parameter (e.g. line, or injection) or system value (e.g. simulation step size) can be changed on-the-fly while calculations are performed, without manual intervention. The commands to perform this changes are issued by the NIOS CPU which can be programmed at will (e.g. in C programming language).

Interfacing. In [273] it is mentioned that DAQ hardware by National Instruments and Labview software has been used as the interface of the platform to the user, via a conventional PC. The interface presented in [271] is mostly dedicated to the configuration of the fixed topology and all preprocessing/post-processing/interpretation of the results is done by the human user.

On the other side, in the software presented in section 3.2 a complete integration of the proposed platform into the frame of power system analysis application is provided. All operations executed by the dedicated hardware are transparent to the user that has the look and feel a purely software solution. On the developers side, an API has been created that allows the integration of the platform as a linear algebra accelerator, into any program by a simple function call.

Functional completeness. Drexel prototypes retain all four subnetwork from the real decomposition of (3.10) [298, 271, 268]. This allows them to retain both conductive and susceptible parts of the admittance matrix of the power system, unlike this work which neglects the conductive part.

The way bus injectors are treated is a fundamental difference between this work and the approach in Drexel. The latter use analog for the components connected to the grid. In the opinion of the author, this adds unnecessary complexity to the design as it is very difficult and time-consuming to reproduce accurate analog models of real system components that may vary arbitrarily in complexity. In our design all computations related to components connected to buses are done in the digital domain, and hence practically unlimited flexibility is offered.

3.5.2 Limitations

Despite its many merits the current architecture has inherent shortcomings that limit its applicability.

- Neglecting the conductance part of the admittance matrix is required.
- Negative electronic conductance values are not supported.
- Non-symmetric admittance matrix non-zero patterns are not supported.
- Despite the digitally controlled switches the pre-fixed topology of the resistor network poses limitations to the power systems (i.e. admittance matrices) that can be handled.
- The USB communication is a major performance bottleneck.

Chapter 4 presents a conceptual new design which retains some of the features of the FPPNS, but in the same time tries to mitigate many of its shortcomings.

4 Concept future solver

In this chapter, a hybrid analog-digital computer, called mixed-signal computer (MSC), is proposed as a linear algebra accelerator specially adapted to power system analysis requirements. The MSC retains some design principles of the FPPNS while overcoming its limitations. It works in conjunction with a host PC conventional power system simulation software. The program flow is diverted to the MSC when intensive linear algebra operations are required. The latter returns the result and the flow of the conventional software continues.

The analog and the digital architecture of the proposed MSC are presented, alongside a detailed analysis of the design procedure and the mathematical properties of the platform. A software emulator of the proposed architecture is created in order to validate the concept, assess its functionality. With the help of the emulator an exploration of the design space is conducted and a set of design guidelines is given. The computational prowess of a suggested implementation is demonstrated by its application to small- and mid-sized realistic topologies.

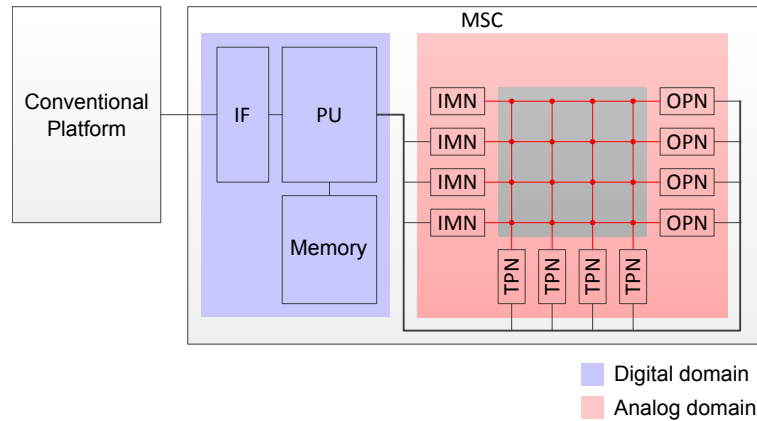


Figure 4.1 – A schematic overview of the proposed mixed-signal computer (MSC)

The proposed new design concerns a linear-algebra enabled accelerator dedicated to power system computations. The new design has an analog part and a digital part and will be hereafter referred to as a Mixed-Signal Computer (MSC). The new system retains some design concepts of the existing FPPNS prototype while in the same time it has provisions to overcome its limitations.

It is meant to be connected to the frame (e.g. motherboard) of a conventional digital computer, through a standard digital interface. The conventional computer runs the power system analysis application, and linear algebra computations are offloaded to the MSC. Fig. 4.1 gives a conceptual overview.

On the left of the figure there is the conventional platform that runs the power system analysis software. In this study a typical desktop PC has been used a platform and the RAMSES phasor transient simulator of [6] has been used a power system analysis software.

On the right the MSC is shown schematically. Parts shaded in light blue are in the digital domain, while parts shaded in red are in the analog domain. The connection between the conventional platform and the MSC is realized through a standard electronic interface, e.g. PCI express, the controller of which is depicted in the IF box. The digital part does not have any computing capabilities. Instead its processing unit PU handles all tasks that concern the configuration of the analog part as well as the communication/interfacing of the MSC with the conventional computer.

There is a digital control bus between the digital PU and the analog parts of the MSC, denoted by a bold black line in the figure. The analog part of the MSC is the computing core dedicated to linear algebra operations. It consists of four main parts: Injection and Measurement Nodes (IMN), Two-Port Networks (TPN), One-Port Networks (OPN), and the interconnection fabric between them - shown as a grid of analog (red) lines shaded in gray. The exact functioning of the analog grid and its components is explained in the next subsections.

Table 4.1 – Comparison of the roles of different platforms and domains in the FPPNS and the MSC

Platform/Domain	FPPNS	MSC
PC	Only handles user interface and auxiliary tasks	Controls the flow of the power system analysis program (RAMSES); also handles auxiliary tasks for the MSC
Digital hardware	Controls the flow of the power system analysis program	Only handles auxiliary and interfacing tasks
Analog hardware	Handles the linear algebra part of the computations	Handles the linear algebra part of the computations

Table 4.1 summarizes differences in the design philosophy between the FPPNS of chapter 3 and the MSC of this chapter.

4.1 RAMSES overview

The hardware has been designed by taking into consideration the specificities of the tasks that it needs to execute. This is so, in order to increase the coherence between the hardware platform and the problem specificities as per section 2.2.2 of chapter 2. In this section a short introduction on RAMSES is given, which is the software that will host the use of the MSC.

RAMSES is a relaxable accuracy multi-threaded software simulator for electrical power systems. The general formulation of power system simulation problems has been described in (2.1) in chapter 2 (the equation is copied hereunder for the facilitation of the reader).

$$x = \begin{bmatrix} x_d \\ x_a \end{bmatrix}, \begin{bmatrix} \dot{x}_d \\ 0 \end{bmatrix} = \begin{bmatrix} f_d(x_d, x_a) \\ f_a(x_d, x_a) \end{bmatrix} \quad (2.1)$$

The dynamic x_d and algebraic x_a variables may correspond to the grid (interconnecting components such as branches, transformers etc.) or to injectors connected to the grid (generators, loads, etc.) RAMSES uses the simultaneous approach to solve (2.1). The vector of unknown variables is ordered into subvectors that correspond to injector-related variables $x = [x_1 \ x_2 \ \dots \ x_n]^T$ followed by bus voltages V ; x_i elements are dynamic or algebraic variables and V elements are algebraic variables. Similarly, equations are grouped in sets of DAEs ϕ_i that correspond to injection i , and to a large linear equation g that corresponds to the grid. An implicit integration algorithms, e.g. second order Backward Differentiation Formula (BDF), is used to discretize ϕ_i 's into f_i 's. The combination of the above results in a non-linear system

that has to be solved in each time instant k .

$$\begin{aligned} \dot{x}_i &= \phi_i(x_i, V) \xrightarrow{BDF} 0 = f_i(x_i, V), \quad i = 1 \dots n \\ 0 &= g(x_1, x_2, \dots, x_n, V) \xrightarrow{BDF} 0 = g(x_1, x_2, \dots, x_n, V) \end{aligned} \quad (4.1)$$

For the solution of the non-linear system of (4.1) a Newton scheme is used ($m=1,2,\dots$). For each Newton iteration the values of x_i and V are initialized from the values in the previous time step $\begin{bmatrix} x_i^{m=0} \\ V^{m=0} \end{bmatrix} \leftarrow \begin{bmatrix} x_i(t-h) \\ V(t-h) \end{bmatrix}$ and then the following is performed.

$$x_i^m = x_i^{m-1} + \Delta x_i^m, \quad i = 1 \dots n \quad (4.2a)$$

$$V^m = V^{m-1} + \Delta V^m \quad (4.2b)$$

In (4.2), the Newton corrections Δx_i^m and ΔV^m at iteration m , come from the solution of the following linear system.

$$\begin{bmatrix} A_1 & & & B_1 \\ & A_2 & & B_2 \\ & & \ddots & \vdots \\ & & & A_n & B_n \\ -C_1 & -C_2 & \dots & -C_n & D \end{bmatrix} \cdot \begin{bmatrix} \Delta x_1^m \\ \Delta x_2^m \\ \vdots \\ \Delta x_n^m \\ \Delta V^m \end{bmatrix} = - \begin{bmatrix} f_1(x_1^{m-1}, V^{m-1}) \\ f_2(x_2^{m-1}, V^{m-1}) \\ \vdots \\ f_n(x_n^{m-1}, V^{m-1}) \\ g(x^{m-1}, V^{m-1}) \end{bmatrix} \quad (4.3)$$

In (4.3) A_i, B_i are the Jacobians of f_i w.r.t. x_i and V respectively. C_i are matrices to extract current components from x_i and D corresponds to a reordered real expansion of the admittance matrix Y of the grid.

$$I = D \cdot V \Leftrightarrow \begin{bmatrix} i_1^I \\ i_1^R \\ i_2^I \\ i_2^R \\ \vdots \\ i_N^I \\ i_N^R \end{bmatrix} = \begin{bmatrix} b_{11} & g_{11} & b_{12} & g_{12} & \dots & b_{1N} & g_{1N} \\ g_{11} & -b_{11} & g_{12} & -b_{12} & \dots & g_{1N} & -b_{1N} \\ b_{21} & g_{21} & b_{22} & g_{22} & \dots & b_{2N} & g_{2N} \\ g_{21} & -b_{21} & g_{22} & -b_{22} & \dots & g_{2N} & -b_{2N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{N1} & g_{N1} & b_{N2} & g_{N2} & \dots & b_{NN} & g_{NN} \\ g_{N1} & -b_{N1} & g_{N2} & -b_{N2} & \dots & g_{NN} & -b_{NN} \end{bmatrix} \cdot \begin{bmatrix} v_1^R \\ v_1^I \\ v_2^R \\ v_2^I \\ \vdots \\ v_N^R \\ v_N^I \end{bmatrix} \quad (4.4)$$

The Bordered Block Diagonal (BBD) structure of the Jacobian of (4.2) is exploited to decompose

the problem into $n + 1$ subsystems using the Schur complement domain decomposition method.

$$\tilde{D} \underbrace{\Delta V^m}_x = - \underbrace{g(x^{m-1}, V^{m-1}) - \sum_{i=1}^n C_i A_i^{-1} f_i(x_i^{m-1}, V^{m-1})}_b \quad (4.5a)$$

$$\text{where } \tilde{D} = D + \sum_{i=1}^n C_i A_i^{-1} B_i \quad (4.5b)$$

$$A_i \cdot \Delta x_i^m = -f_i(x_i^{m-1}, V^{m-1}) - B_i \Delta V^m, \quad i = 1 \dots n \quad (4.5c)$$

The (reordered real expansion of the) admittance matrix is Schur-augmented in (4.5b). By construction, the factors $C_i A_i^{-1} B_i$ are 2×2 correction matrices that are added along the diagonal of D , in the position that corresponds to the bus that the injector is connected to. They result from the discretization of the differential algebraic equations that govern the dynamic behavior of injections connected to power system buses.

$$K_i = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} & \vdots \\ 0 & \dots & 0 \end{bmatrix} \quad (4.6)$$

In (4.5a), the unknown vector x and the r.h.s. vector b have units of voltage and current respectively, but are not exactly voltage and current quantities. Actually x is a voltage correction vector and b is a pseudo-current vector, the explanation of which can be found in [5]. Analogously to (4.4) voltage corrections and pseudo-currents can be written in real and complex quantities $x = [x_1^R \ x_2^I \ \dots \ x_N^R \ x_N^I]^T$ and $b = [b_1^I \ b_1^R \ \dots \ b_N^I \ b_N^R]^T$.

Finally, the linear system of (4.5a) is solved for $x \equiv \Delta V^m$ and (4.5c) is solved for Δx_i^m , for every injector i . Having ΔV^m and Δx_i^m , the next step in the Newton iteration is calculated from (4.2). The above flow is summarized in Fig. 4.2. The demanding linear algebra operation of (4.5a) is highlighted in red.

For reasons that will become evident later on in this work, the signs of the rows corresponding to imaginary pseudo-current quantities are inverted. The final linear system is written hereunder. The solution of (4.7) is equivalent to (4.5a). \mathcal{I} is the pseudo-current vector, $\Psi = \mathcal{Y} + \mathcal{K}$ is the augmented, reordered, sign-inverted (for odd rows only) admittance matrix, and \mathcal{V} is the pseudo-voltage solution. \mathcal{K} is the ensemble of the individual diagonal corrections \mathcal{K}_i that

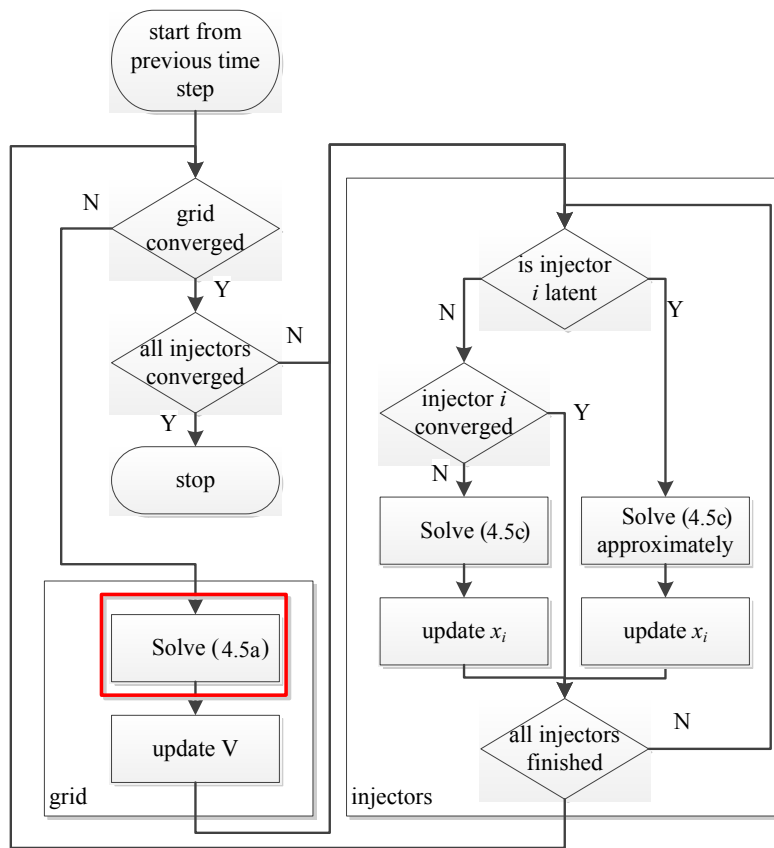


Figure 4.2 – The flow of the RAMSES transient simulator with the linear algebra operation identified in red

correspond to injectors i .

$$\underbrace{\begin{bmatrix} -b_1^I \\ b_1^R \\ -b_2^I \\ b_2^R \\ \vdots \\ -b_N^I \\ b_N^R \end{bmatrix}}_{\mathcal{I}} = \underbrace{\left(\underbrace{\begin{bmatrix} -b_{11} & -g_{11} & -b_{12} & -g_{12} & \dots & -b_{1N} & -g_{1N} \\ g_{11} & -b_{11} & g_{12} & -b_{12} & \dots & g_{1N} & -b_{1N} \\ -b_{21} & -g_{21} & -b_{22} & -g_{22} & \dots & -b_{2N} & -g_{2N} \\ g_{21} & -b_{21} & g_{22} & -b_{22} & \dots & g_{2N} & -b_{2N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -b_{N1} & -g_{N1} & -b_{N2} & -g_{N2} & \dots & -b_{NN} & -g_{NN} \\ g_{N1} & -b_{N1} & g_{N2} & -b_{N2} & \dots & g_{NN} & -b_{NN} \end{bmatrix}}_{\mathcal{Y}} + \sum_{i=1}^n \underbrace{\begin{bmatrix} 0 & \dots & 0 \\ \vdots & \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} & \vdots \\ 0 & \dots & 0 \end{bmatrix}_i}_{\mathcal{K}_i} \right)}_{\Psi} \cdot \underbrace{\begin{bmatrix} x_1^R \\ x_1^I \\ x_2^R \\ x_2^I \\ \vdots \\ x_N^R \\ x_N^I \end{bmatrix}}_{\mathcal{V}} \quad (4.7)$$

The main purpose for the design of the MSC is to tackle the computational needs of this linear system solving operation. Particularities of the original power system problem have been taken into account in the design of the MSC. The next sections detail how.

4.2 Design methodology

There are three steps in creating (4.7).

1. The admittance parameters of the power system branches and shunt elements are expressed using an ordering and a sign convention that is compatible with (4.7).
2. The \mathcal{Y} matrix is created by connecting the branches between the buses, according to the given (graph) topology of the system. The same is done with the bus shunt elements.
3. Ψ is generated by augmenting \mathcal{Y} with the diagonal corrections that correspond to the injectors. These corrections are also expressed to respect the ordering and sign convention compatible with the above, and they are applied to 2×2 position (rows and columns) of the main diagonal of \mathcal{Y} that corresponds to the bus where the injector is connected to.

The design of the MSC is based on the steps above through the mapping principle of Fig. 3.9, which has been retained.

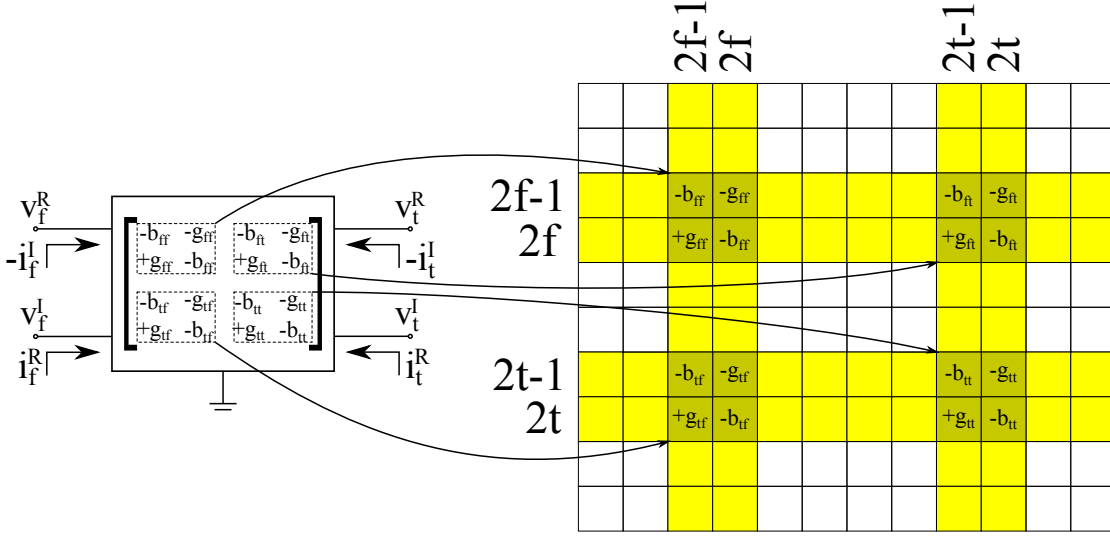


Figure 4.3 – The two-port, four-pole network (2 poles per port) defined for a branch by (4.8) and the effect it has on the building of the \mathcal{Y}

4.2.1 Power system components and matrix building

The electrical nature of a branch can be modeled using its complex y-parameters of (3.1). These parameters can be written in a real-expanded form using an ordering and a sign convention compatible to the one of \mathcal{Y} .

$$\begin{bmatrix} -i_f^I \\ i_f^R \\ -i_t^I \\ i_t^R \end{bmatrix} = \underbrace{\begin{bmatrix} -b_{ff} & -g_{ff} & -b_{ft} & -g_{ft} \\ g_{ff} & -b_{ff} & g_{ft} & -b_{ft} \\ -b_{tf} & -g_{tf} & -b_{tt} & -g_{tt} \\ g_{tf} & -b_{tf} & g_{tt} & -b_{tt} \end{bmatrix}}_y \cdot \begin{bmatrix} v_f^R \\ v_f^I \\ v_t^R \\ v_t^I \end{bmatrix} \quad (4.8)$$

The parameters of the above can be derived by taking into account the model of a power system branch, as shown in Fig. 3.6. A two-port, four-pole network that is defined by (4.8) is shown in Fig. 4.3 alongside the effect it has to the building of the \mathcal{Y} . This is analogous to the complex case of Fig. 3.7 of chapter 3.

For the different parts of Fig. 3.6 different models can be assumed. Table 4.2 summarizes these different sub-models. The model of the shunts part can be codified using two axes. The first axis concerns the completeness of the model and the second whether it is balanced or not. The letter codes for the table are as follows.

- for Transformer part models: F - full model, R - non-phase shifting model, N - no transformer

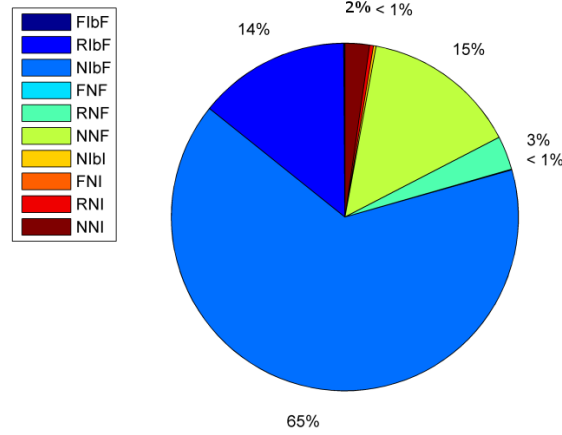


Figure 4.4 – Empirical pie chart of occurrences of branch types for several typical power systems with sizes ranging between 3-15k buses

- for Shunts part models (first axis): F - full model, I - imaginary model, N - no shunts
- for Shunts part models (second axis): u - unbalanced, b - balanced
- for Line part models: F - full model, I - imaginary model

The combination of models for the transformer, the shunts and the line parts yields the overall two-port network model of the branch. The following nomenclature is established: models are to be referred to using the following abbreviation TSsL. T refers to the transformer model, S refers to the completeness axis of the shunts model, s refers to the balanced-ness axis of the shunts model and L refers to the line model. Capital X or small x is used as a wildcard, i.e. means any submodel. E.g. NIbF has a N model for the transformer, an Ib model for the shunts and an F model for the line.

Fig. 4.4 shows the pie chart of the branch types that occur in the power systems of appendix A. The size of the power systems is between 3 and 15226 buses. Among all possible models, there are ten types (FIbF, RIbF, NIbF, FNF, RNF, NNF, NIbI, FNI, RNI and NNI) that appear in systems used in transient simulations.

Analogously, for a shunt element ($g_{sh} + j \cdot b_{sh}$) on bus n the y-parameters are as follows. The one-port network defined by these parameters and its effect on the building of the \mathcal{Y} is shown in Fig. 4.5. This is analogous to the complex case of Fig. 3.8.

$$\begin{bmatrix} -i_n^I \\ i_n^R \end{bmatrix} = \begin{bmatrix} -b_{sh} & -g_{sh} \\ g_{sh} & -b_{sh} \end{bmatrix} \cdot \begin{bmatrix} v_n^R \\ v_n^I \end{bmatrix} \quad (4.9)$$

Table 4.2 – Different models for different parts of the branch model of Fig. 3.6

Transformer part	Shunts part		Line part
Full model: In the most general case no assumptions are made for the transformer parameters, i.e. $x \neq 0$ close to 1 and $\theta \neq 0$. When the transformer is operating at nominal tap setting then $x = 1$. Non-phase shifting model: When the transformer is not phase shifting then $\theta = 0$ and the ratio is real $n = x \in \mathbb{R}$.	Full model: No constraints are set for the parameters of the shunts, g_f, b_f, g_t and b_t . Imaginary model: The conductive parts at the “from” and the “to” sides are neglected ($g_f = g_t = 0$) and the shunt admittances only have a susceptive part.	Unbalanced: This is the general case where the “from” part and the “to” part can have different parameters. Balanced: The shunts are equal for both sides $g_f = g_t$ and $b_f = b_t$.	Full model: No constraints are set for the parameters of the line, r_s and x_s . Imaginary model: In high-voltage transmission lines a common assumption that $x_s \gg r_s$ is made. This leads to $g_s \rightarrow 0$, hence the line has only a susceptive part.
No transformer: Neglecting the transformer means $x = 1$ and $\theta = 0$ and the branch becomes a simple line; the generalized π model of Fig. 3.6 becomes a standard (non-generalized) π model.	No shunts: Neglecting the shunts means $g_f = b_f = g_t = b_t = 0$.		

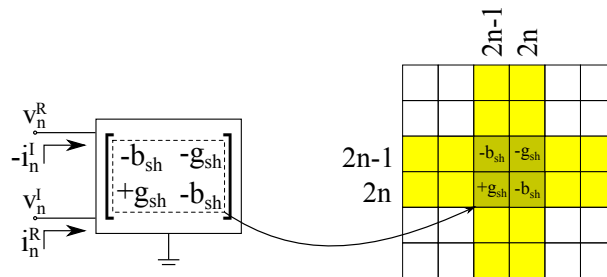


Figure 4.5 – The one-port, two-pole network (2 poles per port) defined for a shunt element by (4.9) and the effect it has on the building of the \mathcal{Y}

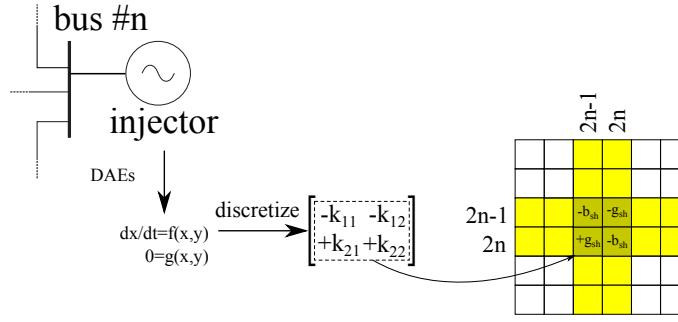


Figure 4.6 – The 2×2 diagonal correction that arises from the dynamic behavior of an injector and the effect it has in the augmenting of \mathcal{V} to $\tilde{\mathcal{V}}$

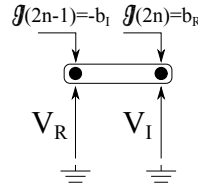


Figure 4.7 – An equivalent of a power system bus that follows the real decomposition, the ordering and the sign convention for the MSC

The \mathcal{V} matrix is generated, by taking into account the effect of all branches and shunt elements. Then it needs to be augmented into $\tilde{\mathcal{V}}$ by taking into account all the diagonal corrections that correspond to injectors. According to observations made earlier in this section, the diagonal correction for an injector represents a relation between the voltage at the bus where the injector is connected at and the current injected into the grid. This is formalized in the following equation. The effect of the diagonal correction corresponding to an injector to the building of matrix $\tilde{\mathcal{V}}$ is illustrated in Fig. 4.6.

$$\begin{bmatrix} -i_n^I \\ i_n^R \end{bmatrix} = \begin{bmatrix} -k_{11} & -k_{12} \\ k_{21} & k_{22} \end{bmatrix} \cdot \begin{bmatrix} v_n^R \\ v_n^I \end{bmatrix} \quad (4.10)$$

To accompany the matrix building procedure described above, a model is also assumed for the nodes of the system. These concern the pseudo-current r.h.s. vector \mathcal{J} and the solution vector \mathcal{V} . Given the real decomposition scheme and the sign convention that has been adopted, a bus of the power system can be represented as a set of two points, for which the currents are injected and the voltages are measured as shown in 4.7. b_I and b_R refer to the b vector of (4.5a).

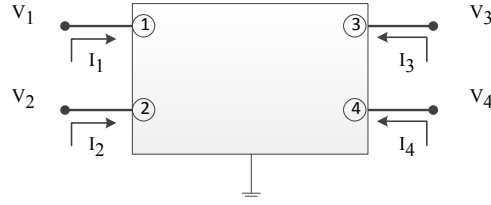


Figure 4.8 – An outline of an electrical two-port network used to map a power system branch

4.2.2 Electronic equivalents

Based on the analysis of the power system components presented in the previous section, their electronic equivalents are presented here.

Electronic Two-Port Networks

For an electronic two-port network to have a conductance parameter matrix such as the one of (4.8) a two-port network four-pole design such as the one of Fig. 4.8 is adopted. The resulting conductance parameters are as follows in (4.11). The nodes of interest in the TPN are the nodes denoted by 1, 2, 3 and 4. The effect of any internal circuitry will be projected to the overall g-parameters of the two-port network, as long as it involves one of these nodes.

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \quad (4.11)$$

The building blocks of the internal topologies of the TPN are *potentiometers* and *voltage controlled current sources (VCCS)*. The effect of a potentiometer to the g-parameters of the TPN is shown in Fig. 4.9a. A VCCS is an active electronic element which can control the current injection at an electrical node based on the voltage difference at other points of the circuit, i.e. the transconductance/transimpedance phenomenon. This explains the fact that a VCCS can have an effect at a single position of the g-parameters matrix, without affecting the symmetrical entry or the corresponding entries of the main diagonal, as shown in Fig. 4.9b.

The y-parameters matrix of each power system branch type has different degrees of freedom and numerical properties, e.g. non-zero pattern, symmetry, etc. Analysis conducted by the author showed that five distinct electronic TPN modifications are sufficient to cover all branch models. These modifications are called TPN Mk1 ... Mk5. Table 4.3 summarizes their ability to represent the power system branch types of Fig. 4.4. TPNs are in decreasing order of complexity.

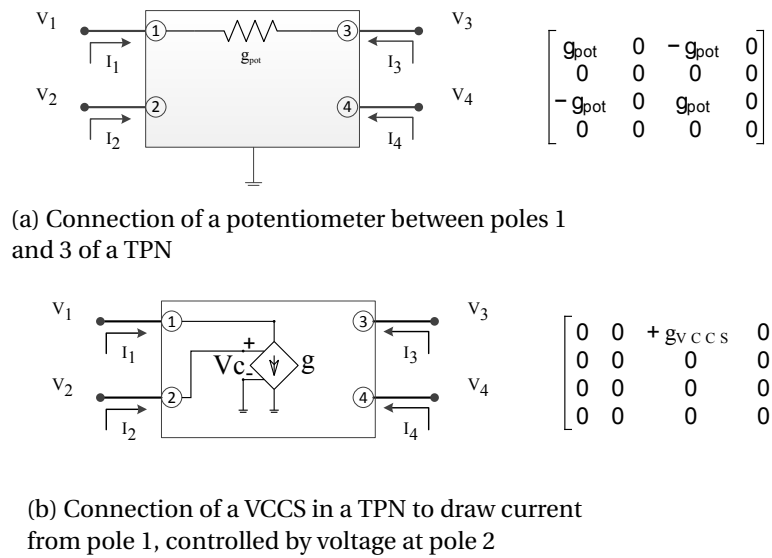


Figure 4.9 – Examples of the effect of the TPN building blocks (potentiometers and VCCS) to the g-parameters matrix of the TPN

Table 4.3 – TPN modifications and their ability to represent branch types

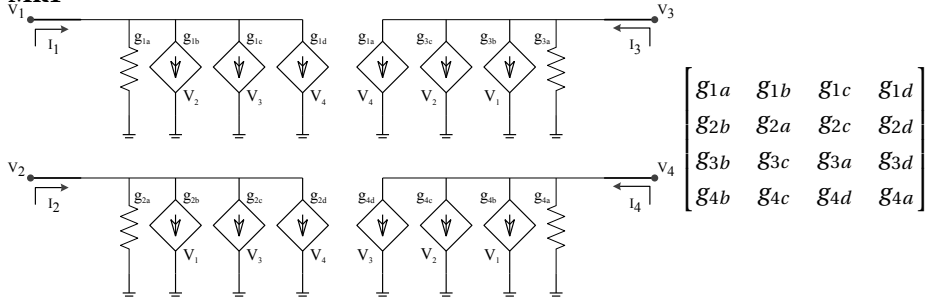
TPN	FIbF	RIbF	NIbF	FNF	RNF	NNF	NIbI	FNI	RNI	NNI
Mk1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Mk2	N	Y	Y	N	Y	Y	Y	Y	Y	Y
Mk3	N	N	N	N	N	Y	Y	Y	Y	Y
Mk4	N	N	N	N	N	N	Y	N	Y	Y
Mk5	N	N	N	N	N	N	N	N	N	Y

It is always desirable to represent branch types with TPNs that are just complex enough and “Y”s in bold show the TPN that fulfills this criteria. TPNs of higher complexity can be used to represent branches of lower complexity. This can be thought of as a procedure of downgrading, as it uses resources in excess of what is really necessary for the representation of the branches. TPN downgrading might be useful in cases when the TPN bank is out of resources of a particular Mk, but there are still resources available in Mks of higher complexity.

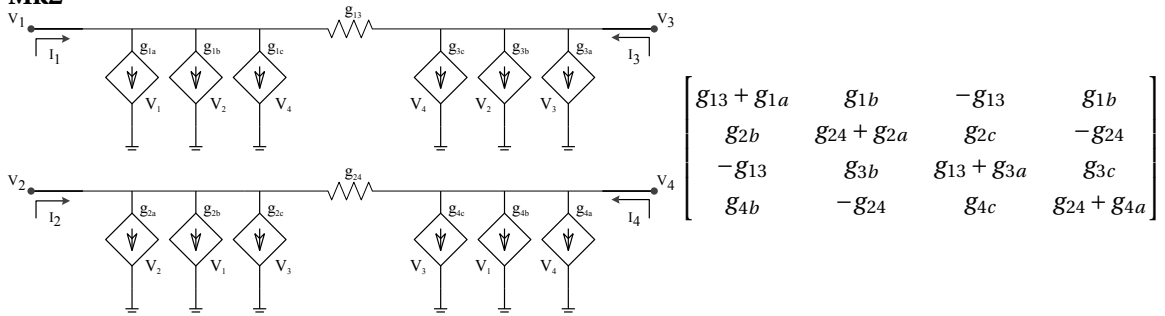
An overview of the architecture of TPN modifications alongside the resulting electronic g-parameters matrix is presented hereunder. The total g-parameters matrices are the combined effect of potentiometers and VCCS in the TPNs. Voltage variables beside the VCCSs denote the controlling voltage of the source.

For TPN Mk3 different dependencies are required for different models that are mapped onto the Mk, e.g. NNF and FNI. Therefore two “versions” of the g-parameters matrix are shown. For the implementation, the issue can be easily addressed with an additional digitally controlled switch that controls the control voltage of the respective VCCS.

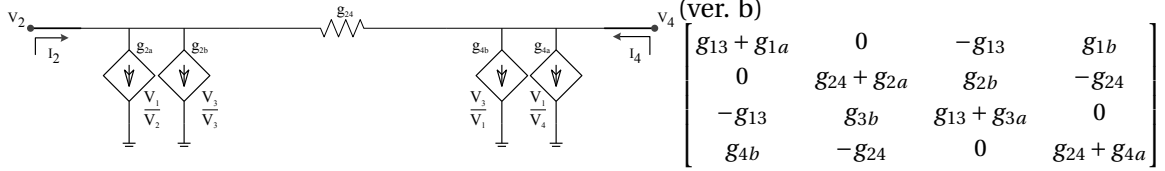
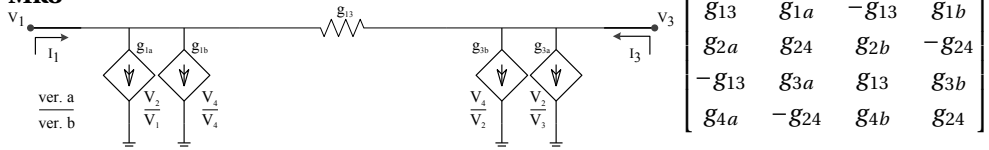
Mk1



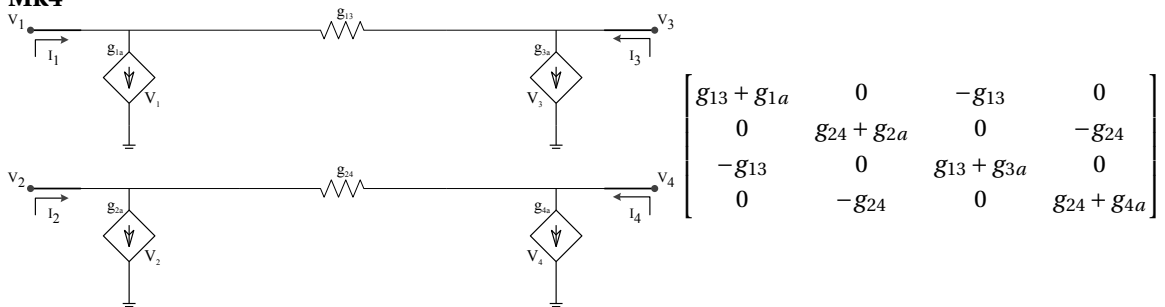
Mk2



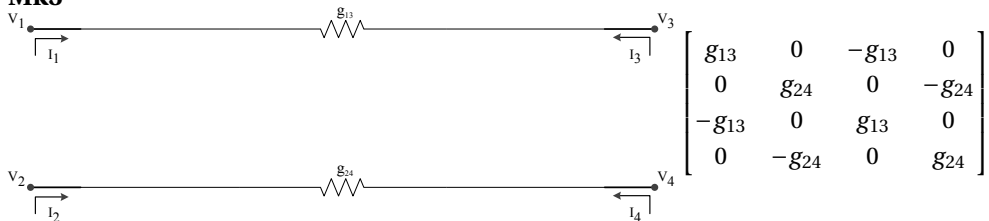
Mk3



Mk4



Mk5



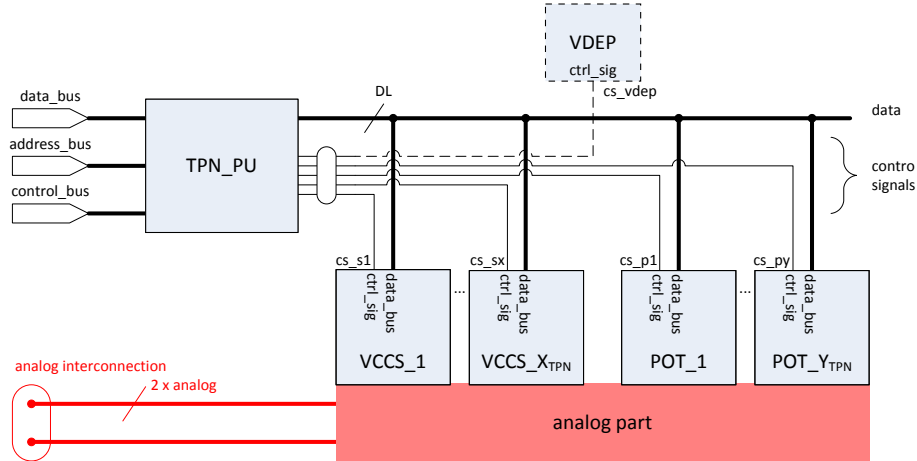


Figure 4.10 – Schematic of the digital part of a TPN

Apart from the analog part, TPNs also have a digital part, illustrated in Fig. 4.10. In the figure, black wires represent digital connections, red wires and coloring represent analog connections and generally analog domain functionality, bold wires stand for (digital or analog) buses (i.e. more than one connections).

The TPN_PU is the digital processing unit present in the TPN. Depending on its Mk, the TPN may have X_{TPN} number of VCCSs and Y_{TPN} number of potentiometers. The analog functionality of the TPN is depicted as the red square on the bottom of the figure.

The TPN_PU is connected to the rest of the MSC by three digital buses (data_bus, address_bus and control_bus). These are used to deliver commands to the TPN by higher levels of the MSC hierarchy. In turn the TPN_PU generates the internal digital signals that are required to drive its VCCSs and POTs.

Table 4.4 summarizes the complexity of TPN modifications in terms of numbers of VCCSs (X_{TPN}) and potentiometers (Y_{TPN}). The last column of the table refers to the nominal percentage of TPN of each type that should be available in the TPN bank. This percentage comes from the sum of occurrence percentages of the branches (see Fig. 4.4) that the TPN Mk is primarily used to represent (second to last column).

Electronic One-Port Networks

An analogous approach has been followed for shunt elements and diagonal corrections. An one-port network (OPN) structure has been created as shown in Fig. 4.11. Its g-parameters matrix is as follows in (4.12). The nodes of interest in the OPN nodes are 1 and 2. The effect of any internal circuitry will be projected to the overall g-parameters of the two-port network, as long as it involves one of these nodes. The building blocks of OPNs are again potentiometers and VCCSs. Two OPN modifications are proposed. Their architecture alongside their corre-

Table 4.4 – TPN modifications, their complexity and their primary usage for branch type representation

TPN	X_{TPN}	Y_{TPN}	Primarily used for	Nominal percentage in the TPN bank
Mk1	12	4	F1bF, FNF	0.20 %
Mk2	12	2	R1bF, N1bF, RNF	82.40 %
Mk3	8	2	NNF, FNI	14.53 %
Mk4	4	2	N1bI, RNI	0.58 %
Mk5	0	2	NNI	2.29 %

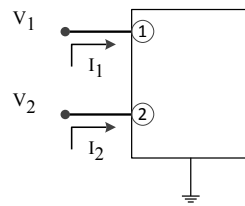
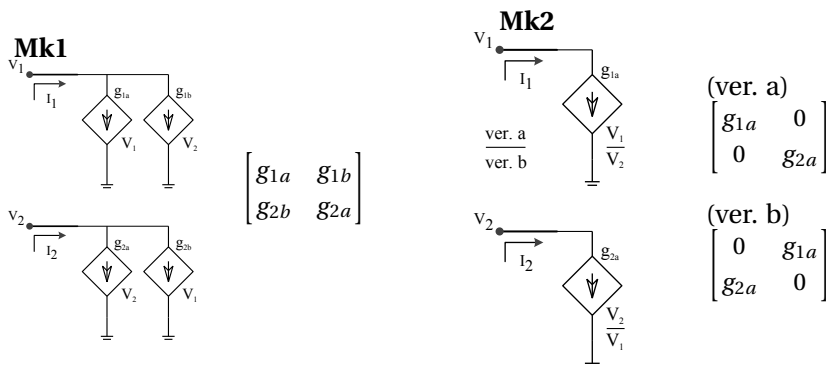


Figure 4.11 – An outline of an electrical one-port network used to map a power system shunt element or a diagonal correction

sponding g-parameter matrices are presented hereunder. OPN Mk1 is used to represent shunt elements or diagonal corrections when they are complex (both real and imaginary parts) while OPN Mk2 can represent only real-only or imaginary-only power system one-ports.

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (4.12)$$



OPNs also have a digital part, illustrated in Fig. 4.12. The figure is analogous to Fig. 4.10 of the TPN case.

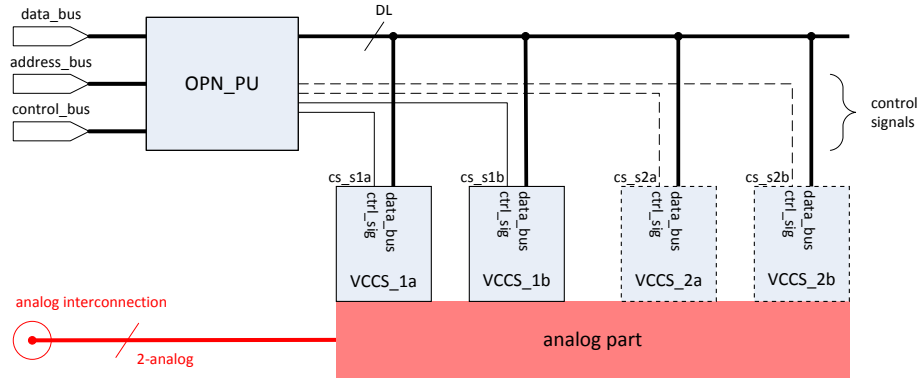


Figure 4.12 – Schematic of the digital part of an OPN

Injection and Measurement Node

The pseudo-current injections and the pseudo-voltage measurements of Fig. 4.7 will be performed by an array of electronic nodes. A synoptical diagram of this injection and measurement node (IMN) is shown in Fig. 4.13. I_{ia} and I_{ib} stand for current injection devices, V_{ia} and V_{ib} for voltage injection devices. The type of the injection to the node is determined by a pair of digitally controlled switches.

I_{ma} and I_{mb} are current measurement devices (e.g. ampere-meters) and hence they are connected in series, while V_{ma} and V_{mb} are voltage measurement devices (e.g. volt-meters) and so that are connected in parallel. All building blocks of the IMN are digitally controlled.

The digital part of the IMN is illustrated in Fig. 4.14. The figure is analogous to Figs. 4.10 and 4.12 of the TPN/OPNs.

4.2.3 Value range profiling

For the sizing of the electronics of the MSC, a value profiling of the corresponding power system elements is necessary. Empirical tests have been conducted on all the power systems of appendix A. The value range of the branch admittances is as follows.

$$g \sim (5 \cdot 10^{-3}, 3 \cdot 10^3) \text{ pu} \quad (4.13)$$

$$b \sim (1 \cdot 10^{-2}, 2 \cdot 10^4) \text{ pu} \quad (4.14)$$

The value range of shunt admittances is as follows.

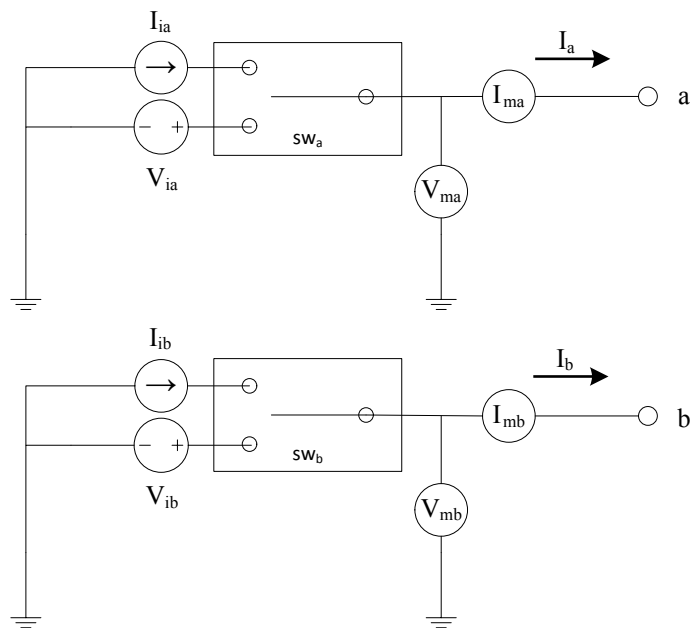


Figure 4.13 – Synoptical diagram of an injection and measurement node of the MSC

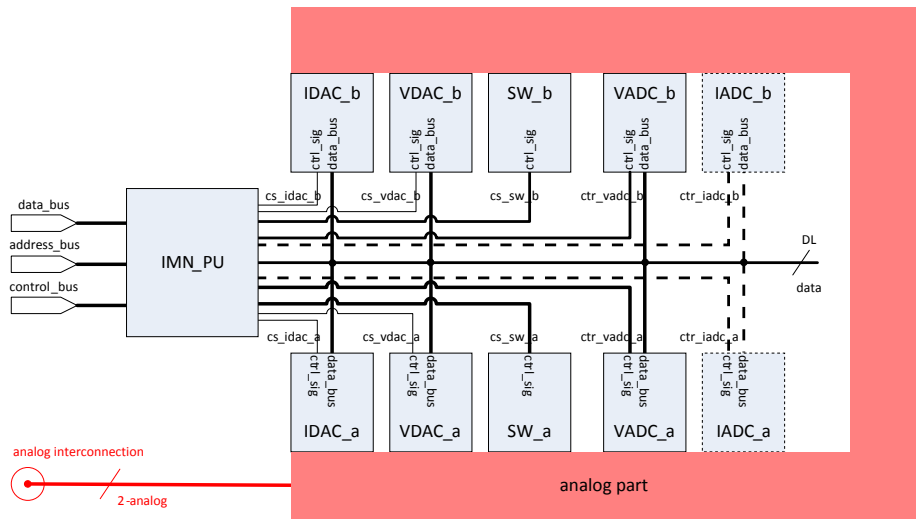


Figure 4.14 – Schematic of the digital part of an IMN

$$g_{sh} \sim (-5 \cdot 10^0, 1 \cdot 10^0) \text{ pu} \quad (4.15)$$

$$b_{sh} \sim (-5 \cdot 10^2, 5 \cdot 10^2) \text{ pu} \quad (4.16)$$

The value range of the diagonal corrections is as follows.

$$(-k_{11}), (k_{22}) \sim (-5 \cdot 10^2, 5 \cdot 10^0) \text{ pu} \quad (4.17)$$

$$(-k_{12}), (k_{21}) \sim (-5 \cdot 10^1, 5 \cdot 10^1) \text{ pu} \quad (4.18)$$

The range of pseudo-current values is as follows.

$$\mathcal{J} \sim (-5 \cdot 10^2, +5 \cdot 10^2) \text{ pu} \quad (4.19)$$

Finally, the range of pseudo-voltage values is as follows.

$$\mathcal{V} \sim (-1.2, +1.2) \text{ pu} \quad (4.20)$$

4.3 MSC architecture

4.3.1 Local cells

A set of a $Y2_{LC}$ IMNs, $Y2_{LC}$ OPNs, and X_{LC} TPNs consist an entity that in the MSC frame is called a *local cell* (LC), shown schematically in Fig. 4.15. A LC is a number of electrical nodes with higher degree of connectivity between them. This is so in order to resemble the clustered nature of real world power system grids. In a local cell $Y2_{LC}$ buses can be represented.

The representation of a bus is handled by the components that are inside the gray shaded boxes on the left of the figure. Injections and measurements are handled by IMNs and the representation of a shunt element and/or diagonal corrections by OPNs. The connection of the latter to the electrical node of the IMN is controlled by the switch OPN_{SW} , which is controlled by a digital line cs.

On the top right of the figure, the interconnection part of the local sell is highlighted in yellow. This interconnection infrastructure is managed by a dedicated processing unit LC_CONN_PU . The horizontal double-analog connections represent the electrical nodes of the MSC. The

vertical lines are connected through analog multiplexers (MUX) to TPNs.

The digital processing core of the LC is the local cell processing unit (LC_PU) on the bottom left of the figure. It controls all the components of the LC through three buses, `lc_data_bus`, `lc_address_bus` and `lc_ctrl_bus`.

The MUX entity (shown on Fig. 4.16) is a digitally controlled double analog multiplexer. Each of the analog lines (shown in bold red in the figure) represents two analog signals, as per the convention in the MSC. The multiplexer on the left is controlled by a digital control signal `mux_f` and leads to the “from” (analog) connectivity side of associated TPN. The multiplexer on the right is controlled by a digital control signal `mux_t` and leads to the “to” (analog) connectivity side of the associated TPN. There are as many positions of the analog multiplexers as there are electrical nodes (IMNs) in the LC, i.e. $Y2_{LC}$, plus one extra position for no connection. This mechanism ensures that the TPN is connected to any IMN on the from-side, and to another one on the to-side. Connectivity is limited to the LC, i.e. only IMNs of the LC can be connected to TPNs of the LC.

For the LC, clusters of $Y2_{LC} = 8$ IMN nodes and $X_{LC} = 15$ TPN branches can be envisioned. This would allow the LC to be able to accommodate a neighborhood of a power system that consists of a cluster of 8 densely connected buses. Out of these buses, there are $Y1_{LC}$ that can be connected to buses of neighboring LCs. This is done in the frame of the global MSC structure that is presented in the following subsection.

4.3.2 Global architecture

Fig. 4.17 presents a synoptical view of the global MSC architecture (GB). The proposed MSC is to be connected to a normal PC architecture as a hardware accelerator through a standard PC bus. The GB consists of Y_{GB} LC entities.

The interconnection part of the GB is highlighted in yellow on the right of the figure, and it is similar to the interconnection found in LC, but not exactly the same. On the bottom of the figure there are X_{GB} TPNs. These are used to connect LCs together through the $Y1_{LC}$ analog connections of the latter. The analog connections of LCs are shown as horizontal red lines in the figure. The from and two sides of the TPNs are connected to analog wires, which are shown as vertical red lines in the figure. The horizontal and vertical lines can be interconnected by a set of switches. For each possible connection, one switch is provided. In total there is the following number of switches.

$$\underbrace{Y_{GB} \cdot Y1_{LC}}_{\text{\# of horizontal lines}} \cdot \underbrace{2 \cdot X_{GB}}_{\text{\# of vertical lines}}$$

Each of the switches is driven by a digital control signal. These control signals are multiplexed

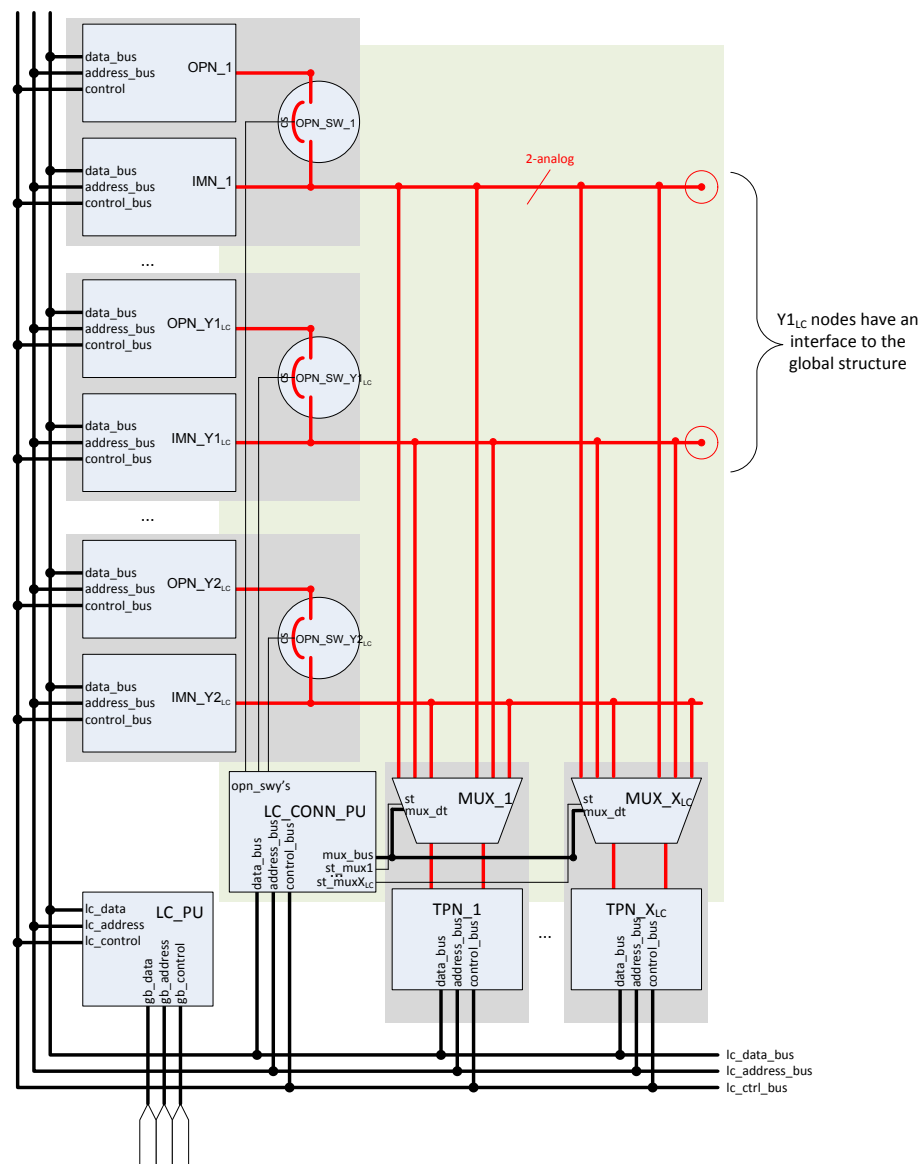


Figure 4.15 – Schematic of a local cell (LC) of the MSC

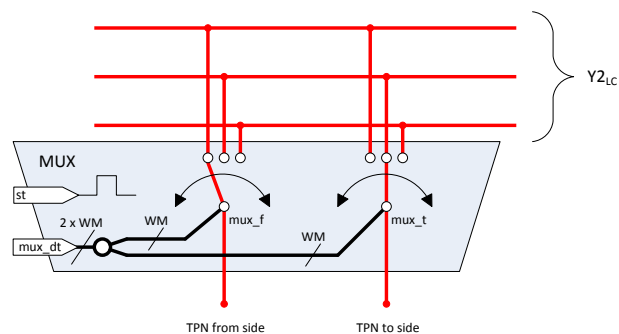


Figure 4.16 – Connectivity detail and simplified schematic of a TPN connection multiplexer (MUX)

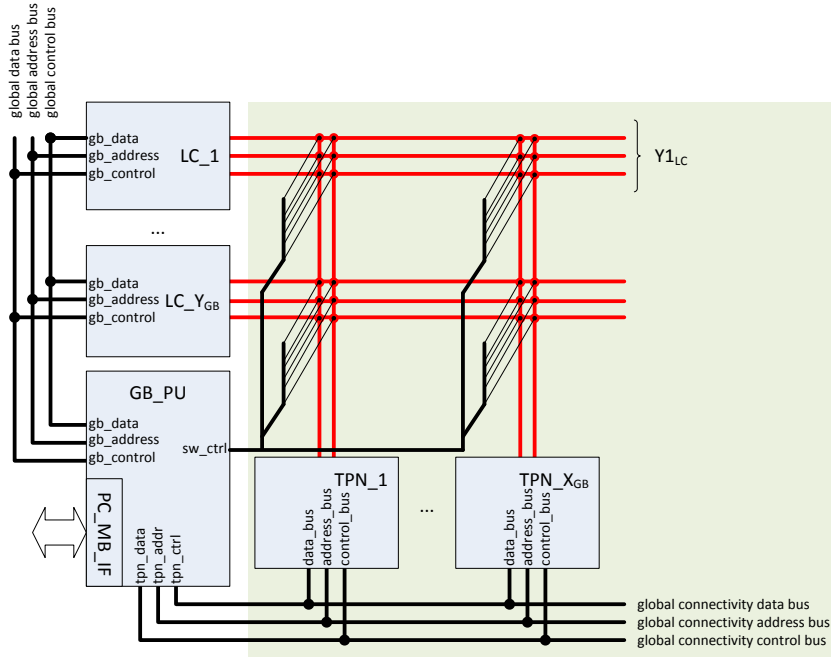


Figure 4.17 – Global view of the digital architecture of the MSC

into GB_PU and fed to a series of cascaded multiplexers up to the point of the single bit wires that control the switches. This is so, as to avoid a potentially prohibitive number of digital lines exiting from the GB_PU.

The reason that this interconnection scheme has been selected for the global architecture of the MSC has to do with parasitic capacitances. Indeed in the interconnection scheme that is used in the LCs there are analog lines that connect each of the TPNs with every IMN. Given the limited size of the LC, this is not problematic. In the GB case however, such a scheme would result in a prohibitive number of total analog connections, which in turn would lead to a prohibitive amount of (capacitive) parasitics. The latter would have an adverse effect on the performance of the MSC, as it will be explained in a later section.

The total number of LCs that are included in the GB MSC scheme depends on the required capacity of the platform. For example assuming LCs with a capacity of 8 nodes, then 10 of them would be required to have a total MSC capacity of 80 buses. The number of TPNs in the GB affects the available connectivity between the LC clusters.

All $Y1_{LC}$, $Y2_{LC}$, X_{LC} , Y_{GB} and X_{GB} are parameters to be decided on depending on the requirements from an actual MSC implementation.

4.3.3 Topological mapping and value mapping

The procedure to associate power system branches to electronic TPNs (edges), and buses and shunts to electronic IMNs and OPNs (nodes) is the *topological mapping* described in section 3.1 of chapter 3. This is achieved through the interconnection fabric of the local cells and the global structure as detailed in the previous subsection. The topological mapping is necessary only once for a given topology.

After a power system branch has been mapped to an electronic TPN the latter has to be configured in a way so that (4.11) is equivalent to (4.8). For a shunt element or diagonal correction mapped on an OPN, the latter has to be configured so that (4.12) is equivalent to the corresponding power system shunt or diagonal correction quantity. Finally, the electronic injections by an IMN associated to a power system bus, have to correspond to the real-world injections of the bus in the power system. This procedure is the *value mapping* of section 3.1 of chapter 3. The multiplicative mapping ratios ρ_Y , ρ_I and ρ_V defined in that section are again relevant. The result is that an electronic equivalent system with is created in electronics.

$$\begin{array}{ccc}
 \mathcal{I} & = & \overbrace{(\mathcal{Y} + \mathcal{K})}^{\Psi} \cdot \mathcal{V} \\
 \uparrow \rho_I & & \uparrow \rho_Y \quad \uparrow \rho_V \\
 I & = & \underbrace{(G + G_K)}_{\Gamma} \cdot V
 \end{array} \tag{4.21}$$

The matrix G is the electronic equivalent of the topology admittance matrix \mathcal{Y} . G is augmented with the extra terms G_K that correspond to the diagonal corrections \mathcal{K} . The resulting total matrix Γ is the electronic equivalent of the power system matrix Ψ . The current injections and measurements (\mathcal{I} and \mathcal{V}) are handled by the IMNs (I and V).

The matrix topology matrix G (made up by TPNs and OPNs corresponding to shunts) is not expected to change often, while G_K (made up by OPNs corresponding to diagonal corrections) may change often. Hence the corresponding OPNs have to modified accordingly. The injection and measurement vectors I and V are the main input and output variables of the MSC and are expected to change in every invocation.

4.3.4 Mathematical operations

In the MSC, voltages and currents can be both injected and read using resources of the IMNs. This is an upgrade compared the FPPNS where currents can be injected but not read from the

grid. For a node, either a voltage or a current injection can be performed. Nodes with voltage injections have unknown current injections, and vice versa nodes with current injections have unknown voltages. Unknown electrical variables are induced by the physics of the system.

Similar to the FPPNS, when only voltages are injected, V is known in (4.21), I is the unknown vector and the operation performed by the analog part of the MSC corresponds to a *matrix-vector multiplication*. When only currents are injected, I is known (rhs vector), V is the unknown vector, and the operation performed corresponds to a *linear system solving*. Mixed voltage and current injections result in the operation described in (3.25).

4.3.5 Inaccuracies and effect in the linear operations

An inaccuracy analysis of the MSC can be performed based on section 3.3 of chapter 3. The following model is assumed for all reconfigurable devices (potentiometers, VCCs, current DACs and voltage DACs) in the system. In the following X may stand for a conductance G of a TPN or OPN, or an injected voltage V or a current I of an IMN.

$$X = \frac{s \cdot t}{2^M} \cdot X_{FS} \quad (4.22)$$

Where X is a physical electronic quantity, s represents the sign, $t \in \mathbb{Z}^+$ is the tap setting, M is the bit resolution and $X_{FS} \in \mathbb{R}^+$ is the full-scale value of the reconfigurable element. The tap setting can take integer values in the range $t \in [0, 2^M)$. The sign s of the electrical quantity can be controlled using one bit s_b , e.g. by a polarity reversal switches at the outputs of DACs. This extra sign bit is not counted in M . The pair $[s_b \ t]$ can be thought of as a signed magnitude representation of the $(M+1)$ -length word of the real signed value of the component. When a specific value X_{req} is requested from the reconfigurable element, s_b and t have to be calculated so that the resulting X from (4.22) best approximates X_{req} . The sign bit is simply set as $s_b \leftarrow \text{sign}(X_{req})$. The tap setting can be calculated from (4.22).

Inaccuracies come from the fact that the real full-scale value X_{FS} is a priori unavailable. Instead knowledge about it is known in the form of a random distribution $\mathbf{X}_{FS} \sim \mathcal{N}(\hat{X}_{FS}, \sigma_{X_{FS}}^2)$. An analysis similar to (3.35)-(3.43) can be followed for the analysis of the inaccuracy that is introduced.

For the measurement devices in the system (voltage and current ADCs) the following simplified model is assumed .

$$s_b \leftarrow \text{sign}(X_{in}) \quad (4.23a)$$

$$D^{\mathbb{R}} = \frac{|X_{in}|}{X_{FS}} \cdot 2^M \quad (4.23b)$$

$$D = \begin{cases} 0 & \text{if } D^{\mathbb{R}} \leq 0 \\ \lfloor D^{\mathbb{R}} + 0.5 \rfloor & \text{otherwise} \\ 2^M - 1 & \text{if } D^{\mathbb{R}} \geq 2^M - 1 \end{cases} \quad (4.23c)$$

$$X = \frac{s \cdot D \cdot X_{FS}}{2^M} \quad (4.23d)$$

The analog input to the ADC is X_{in} and the digital output is the signed magnitude pair $[s_b \ D]$. s_b is the sign bit which is true when $s = +1$. D is the magnitude word of length M . $D^{\mathbb{R}}$ is the magnitude word, if it was allowed to assume non-integer values; it is quantized and saturated to D in (4.23c). X_{FS} determines the range of the ADC, since it is the full scale quantity that can be accepted at its input $|X_{in}| \in [0, X_{FS}]$. The value that is actually “read” by the ADC, and that is communicated to digital parts of the system is given by X . Inaccuracy for (4.23) can be investigated in the lines of (3.61)-(3.65a) of the FPPNS ADCs.

Inaccuracies of individual components contribute to the overall inaccuracy of operations performed by the MSC. Inaccuracy is manifested in different stages, similar to table 3.2 in chapter 3 for the FPPNS.

$$\tilde{\Gamma} = \Gamma + E_{\Gamma} \quad (4.24)$$

$$\tilde{I} = I + \epsilon_I \quad (4.25)$$

$$\tilde{V} = V + \epsilon_{V1} \quad (4.26)$$

$$\tilde{V}' = \tilde{V} + \epsilon_{V2} \quad (4.27)$$

$$(4.28)$$

E_{Γ} inaccuracies in (4.24) are owed to the combined effect of inaccuracies in G and the diagonal correction matrix G_K . The two yields the total inaccuracy in Γ .

$$\left. \begin{aligned} \tilde{G} &= G + E_G \\ \tilde{G}_K &= G_K + E_{G_K} \end{aligned} \right\} \quad \tilde{\Gamma} = \Gamma + E_{\Gamma}, \text{ where } E_{\Gamma} \equiv E_G + E_{G_K} \quad (4.29)$$

Table 4.5 – Interface between the RAMSES software and the MSC platform

Variable	Description	Data Type	I/O type
t	time	1×1 double	I
updtDCorr	flag to denote update of the diag. corrections	1×1 bool	I
dCorr	diagonal corrections	$4 \times M$ double	I
iInj	flags to denote whether node has a current injection	$1 \times 2N$ bool	I
b	bus pseudo-current injection (rhs)	$1 \times 2N$ double	I
x	bus pseudo-voltage (solution)	$1 \times 2N$ double	O

4.3.6 Interface between the MSC and the RAMSES flow

Details about the actual internal memory structure of the GB_PU will not be given since this will depend greatly on the actual final implementation. All the necessary data that are needed for the functioning of the MSC are provided to the GB_PU through PC_MB_IF by the main CPU of the host PC. Then the GP_PU parses the data and generates the words necessary to drive all MSC components through the bus sets (gb_data, gb_address, gb_control) and (tpn_data, tpn_addr, tpn_ctrl).

During the initialization of the use of the MSC the power system data, the mapping data, and the scenario data, are transferred to GB_PU. First, power system and mapping information is parsed and the topology is initialized on the MSC. This involves configure the connectivity switches, configuring the TPN values and configuring the OPN values. The respective commands are issued by the GB_PU to the LC_PUs that are involved.

The scenario file is also parsed, so that the MSC knows when (at which time step) topological changes and faults are to be effectuated in the simulation process. For example, if at some point a three phase fault is to be applied at a bus of the system, in the electronic world this translates as a shunt connection of an OPN to ground at the electrical node that corresponds to the faulted bus. The parameters of the OPN are to be determined by nature of the fault.

In the simulation process, utilization of the MSC is made in the RAMSES flow, as shown in Fig. 4.2. Notice that since a Newton scheme is used for each time step, many internal (Newton) iterations may be required for convergence. The MSC is invoked at every internal iteration. The interface between the MSC and the RAMSES software executed in the conventional CPU(s) of the platform is show in table 4.5. Column *I/O type* is as seen from the GB_PU side.

The time variable t allows the MSC to keep awareness of the simulation. If two subsequent invocations are with the same t , this means that the new invocation concerns another internal Newton iteration for the same time step.

The diagonal corrections vector dCorr correspond to the (linearized) dynamic behavior characteristics of power system elements that are connected to predefined buses. The indexes of

the bus locations of these diagonal corrections are provided by RAMSES at the initialization step of the simulation, as part of the topology of the system. The update of the diagonal corrections is performed only when the `updtDCorr` flag is true. Normally, this happens only for a few iteration in the course of a RAMSES simulation.

The `b` variable contains the vector of the injections required by the current internal Newton iteration of the current time step in RAMSES. The type of the injections is determined by the `iInj` flag vector. The latter has true entries in positions where current injections are to be performed, and false entries at voltage injections. After the $I = \Gamma \cdot V$ solution, the voltage output is written by the MSC into the `V` output vector.

4.3.7 Timing

The sequence of actions in an MSC invocation is as follows.

1. The CPU of the PC writes the interface variables to the GB_PU via the PC_MB_IF.
2. The MSC performs all auxiliary tasks (e.g. internal memory io) and computes the values to be dispatched as commands to the LCs.
3. If topological changes are prevised, the GB_PU issues the topology reconfiguration commands to the LCs.
4. If the `updtDCorr` flag is true, the GB_PU issues OPN (re)configuration commands to the LCs.
5. The MSC issues to LCs the commands to perform the updated DAC injections.
6. Wait for the analog electronics to settle.
7. The MSC issue the read ADC commands to the LCs.
8. The GB_PU sends the solution (`x` vector) back to the CPU of the PC via the PC_MB_IF.

The timing of the above steps depends heavily on the actual implementation. Hereunder, there is an effort to gain some insight on the issue.

The time to complete steps (1) and (8), t_1 and t_8 respectively, depends on the speed of the PC_MB_IF connection. Based on the amount of data to be communicated, these times can be accurately estimated. Time t_2 of step (2) will depend on the actual operations performed by the GB_PU as well as the actual implementation of the latter, and only a rough estimate of it can be given. Times of steps (3), (4), (5), and (7), t_3 , t_4 , t_5 and t_7 can be accurately estimated.

The waiting time of (6), t_6 is the time required for the analog electronics to reach their next steady-state after the DAC injections have been effectuated. This is similar to t_{rd} in the FPPNS implementation. The RC phenomena of Fig. 3.26 occur also in the MSC. For this a simplified RC model similar to (3.75) can be assumed.

4.4 Numerical results

An emulator of the MSC concept has been created in MATLAB. This emulator provides the complete functionality of the MSC in software. It is fully parameterizable in terms of the characteristics (accuracy, bit resolution, etc.) of the electronics that have been modeled. The results presented in this section have been obtained with this emulator.

4.4.1 Selection of mapping ratios

The first design choice is the selection of the mapping ratios. There are three ratios with two degrees of freedom, as per (3.21). The ratio selection is important as the physical limits of the electronics have to be respected, e.g. no mapped conductance must exceed the maximum achievable conductance of the reconfigurable potentiometers and the VCCSs. For the section the following assumptions have been made. These limits are arbitrary and they will depend on the actual electronic realization.

$$\max(g_{el}) = 1/1000 \text{ S} \quad (4.30)$$

$$\max(v_{el}) = 10 \text{ V} \quad (4.31)$$

$$\max(i_{el}) = 100 \text{ mA} \quad (4.32)$$

Full use of the electronic conductance range is made when the highest value of power system admittance value $\max(y_{ps})$ is mapped to the maximum achievable electronic conductance. So a formula for the admittance ratio is as follows.

$$\rho_Y = \max(g_{el}) / \max(y_{ps}) \quad (4.33)$$

The $\max(y_{ps})$ value can be found easily by a search in the conductances of the given topology.

Similarly, it makes sense to associate the maximum power system pseudo-voltage value to the actual maximum electronic voltage.

$$\rho_V = \max(v_{el}) / \max(v_{ps}) \quad (4.34)$$

According to (4.20), a good approximation for the maximum expected pseudo-voltage values is $\max(v_{ps}) = 1 \text{ V}$.

Finally, the current mapping ratio is calculated through Ohm's law of (3.21).

$$\rho_I = \rho_Y \cdot \rho_V \quad (4.35)$$

This is the scheme that has been used in what follows.

4.4.2 Linear system solving

The first way to demonstrate the validity of the MSC concept is to show that it actually solves the problem it is supposed to solve. The 11-bus system of appendix A is used as a test case. Due to the random nature of the true full-scale conductance values, the tests presented in this section have been repeatedly executed so as to avoid probabilistic artifacts.

Grid inaccuracies

Branches and shunt elements of the power systems are mapped onto TPNs and OPNs as per the previous sections. The topology of the power system is recreated in electronics by “virtually” closing the corresponding digital switches in the MATLAB emulator. As a result the $\tilde{\Gamma}$ matrix is created. This $\tilde{\Gamma}$ matrix already takes into account the inaccuracies of the electronics and the quantization, as explained in (4.24).

In fig. 4.18 the relative error of Γ is plotted with respect to the bit resolution of the potentiometers and the VCCSs, as well as with respect to their assumed accuracy. The vertical axis corresponds to the quantity $\|E_{\Gamma}\|_{\infty} / \|\Gamma\|_{\infty}$. The horizontal axis corresponds to different bit resolutions M_G for the reconfigurable conductances. Different lines correspond to different relative accuracies $tol_{G_{FS}}$ (in percent) for the full scale value G_{FS} of the potentiometers. The corresponding standard deviation is calculated as $\sigma_{G_{FS}} = tol_{G_{FS}}/3$.

The effect of the relative accuracy is way more pronounced than the effect of the bit resolution. There is a trend that the error increases for lower resolutions ($M_G \leq 9$), especially for low relative accuracies $tol_{R_{FS}} > 10\%$. The use of an extremely high resolution for the potentiometers is not deemed justifiable as additional potential benefits fail to appear for resolutions above $M_G \geq 13$.

A similar result can be obtained for the inverse of the matrix Γ^{-1} , as shown in Fig. 4.19. The dip for the relative error norm for $M_G = 7$ is coincidental. Again, for $M_G > 10$ and pronouncedly for $M_G \geq 12$ there is a residual error that cannot be further reduced by increasing the bit resolution of the reconfigurable electronics.

Based on the above, a design guideline of $M_G \geq 12$ bits is proposed for the reconfigurable conductances in the MSC.

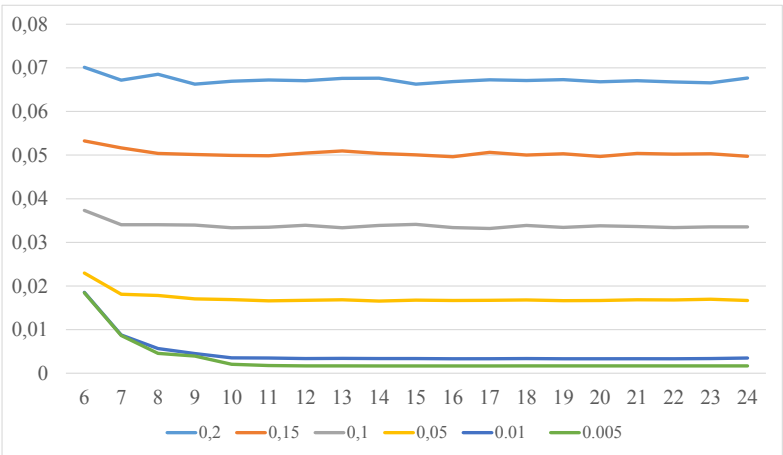


Figure 4.18 – Relative ∞ -norm errors that are introduced in the electronic equivalent of the Γ matrix

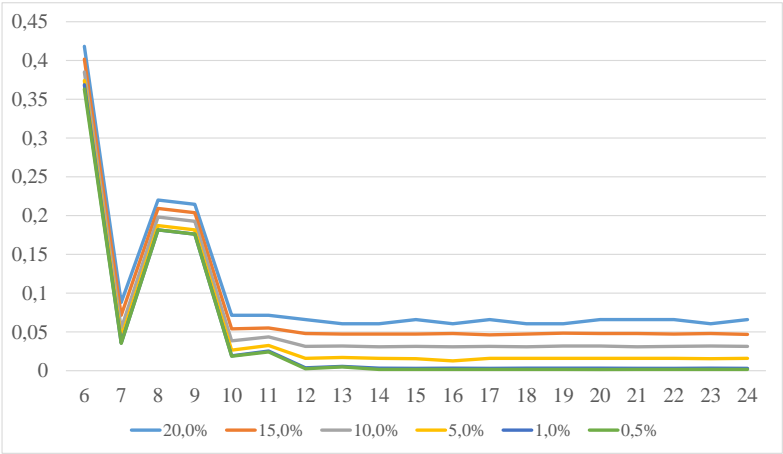


Figure 4.19 – Relative ∞ -norm errors that are introduced in the electronic equivalent of the Γ^{-1} matrix

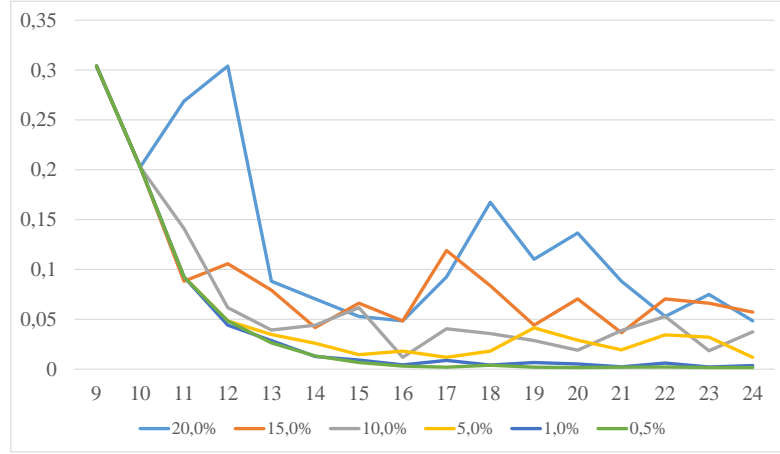


Figure 4.20 – Relative ∞ -norm errors that are introduced in the electronic equivalent of the \mathcal{I} vector

Current injection inaccuracies

A similar study can be performed for the currents that are injected by IMNs of the MSC. Their inaccuracy is mainly controlled by two variables, the bit resolution M_I and the full scale relative accuracy $tol_{I_{FS}}$. The current mapping ratio is as per (4.35). A typical \mathcal{I} injection has been assumed (taken from a real-world scenario) and it has been mapped on the MSC. Results are summarized in Fig. 4.20. The vertical axis corresponds to the quantity $\|\epsilon_I\|_\infty / \|I\|_\infty$, i.e. the relative error to the current injection that is introduced by MSC inaccuracies. The horizontal axis corresponds to different bit resolutions M_I . Different lines correspond to different relative accuracies $tol_{I_{FS}}$.

From the figure it is seen that the increasing the bit resolution of the DACs is beneficial for the overall accuracy. After a point on $M \geq 15$ the residual inaccuracy is mostly due to analog inaccuracy and cannot be decreased more.

Here an important remark should be made regarding the mapping ratios. The results of Fig. 4.20 are obtained by using the ratios of (4.33)-(4.35). In them, the full range of admittances and voltages is exploited by dynamically scaling the ratio in (4.33) and (4.34). The remaining current ratio is calculated by Ohm's law. Instead, similar dynamic scaling could be applied to the current ratio.

$$\rho_I = \max(i_{el}) / \max(i_{ps}) \quad (4.36)$$

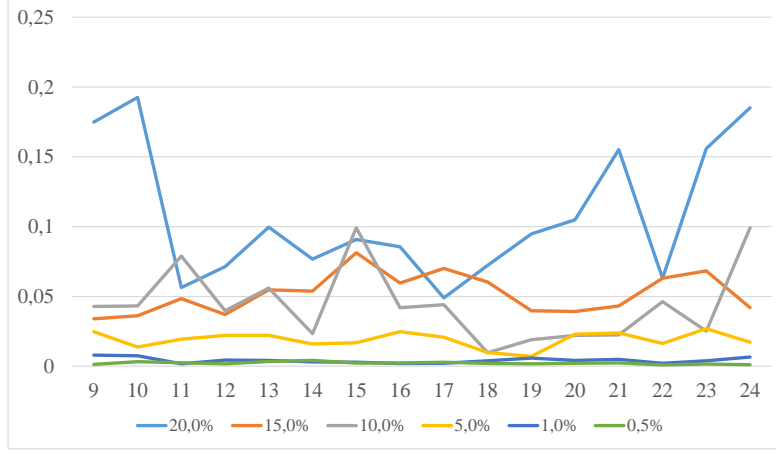


Figure 4.21 – Relative ∞ -norm errors that are introduced in the electronic equivalent of the \mathcal{I} vector; dynamic scaling of the current mapping ratio ρ_I is followed as per (4.36)

Then the voltage ratio has to be calculated by the Ohms law.

$$\rho_V = \rho_Y / \rho_I \quad (4.37)$$

The benefit of this is that the entire range of the current DACs in IMNs can be exploited. This becomes evident in Fig. 4.21. The figure presents the same results as 4.20, but using the dynamic current scaling of (4.36)-(4.37) instead. It can be seen that the error is way less dependent on the resolution, compared to the previous case.

Based on the above, a design guideline of $M_I \geq 16$ bits is proposed for the voltage DACs of the MSC.

Inaccuracies induced to the voltages due to grid and current injection inaccuracies

The error ϵ_{V1} in (4.26) is naturally affected by the assumed E_T and ϵ_I inaccuracies. In turn the latter depend on the assumed inaccuracy of the reconfigurable conductances and current DACs respectively, as shown hereinabove in this section. Fig. 4.22 clarifies this connection. The x-axis is the relative full-scale inaccuracy of the conductances $tol_{G_{FS}}$, the y-axis is the relative full-scale inaccuracy of the I-DACs $tol_{I_{FS}}$. The z-axis corresponds to the quantity $\|\epsilon_{V1}\|_{\infty} / \|V\|_{\infty}$.

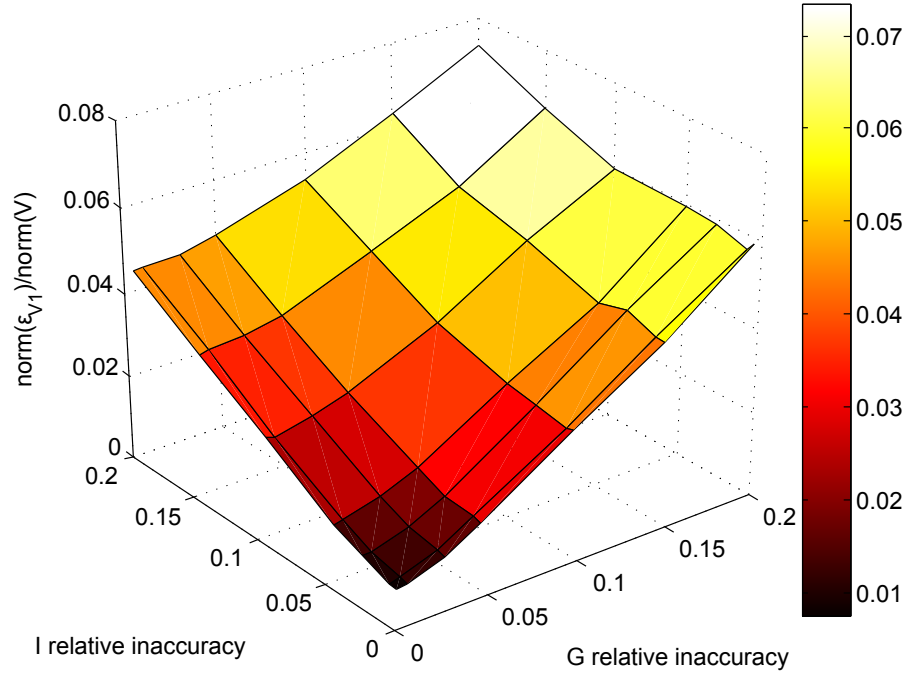


Figure 4.22 – Relative ∞ -norm errors that are introduced to the voltage solution \tilde{V} due to conductance and current inaccuracies of the MSC

From the figure it can be seen that ϵ_{V1} is less sensitive to I inaccuracies than it is to G inaccuracies. This insensitivity is more pronounced for more inaccurate Γ grids (i.e. higher G inaccuracy).

Voltage measurement inaccuracies

The last inaccuracy introduced to the final solution is the one of the voltage DACs of the IMNs ϵ_{V2} . For the purpose of this study no analog inaccuracy has been included in the model of the ADC. Hence the entire ϵ_{V2} inaccuracy in the MATLAB model is due to quantization. Fig. 4.23 summarizes the results of $\|\epsilon_{V2}\|_{\infty} / \|V\|_{\infty}$.

A knee in the graph appears for $M_{V_{ADC}} \approx 11$. A much higher resolution does not offer much in the reduction of the overall error. For the rest of this work a design guideline of $M_{V_{ADC}} \geq 14$ bits is proposed.

The tests presented in this subsection refer to an 11-bus system mapped on the MSC. The graphs presented in the section illustrate the trends on accuracy sufficiently well, since the same observations have been made for larger systems.

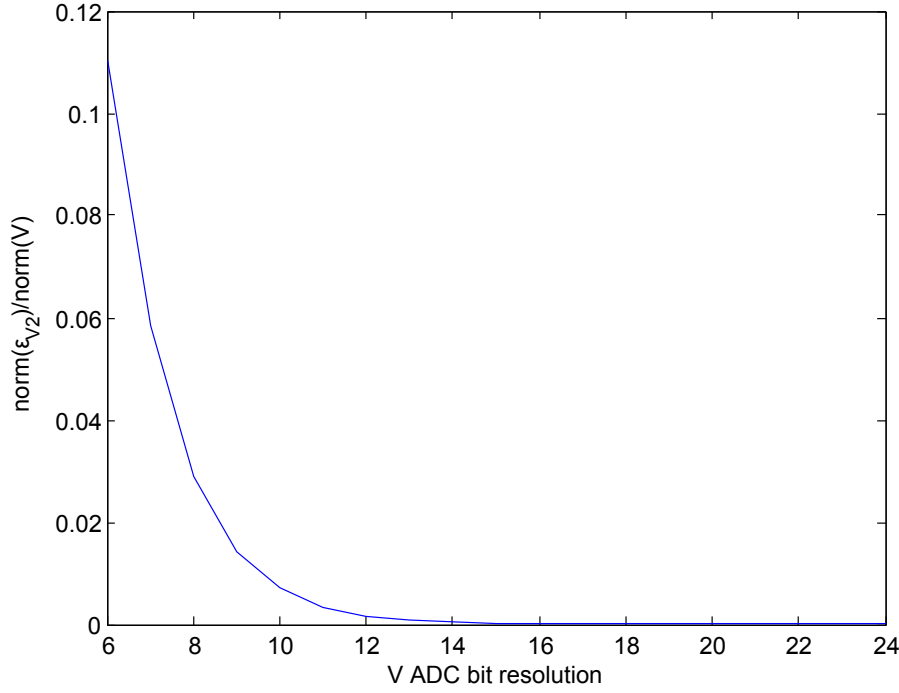


Figure 4.23 – Relative ∞ -norm errors introduced to the voltage solution \tilde{V}' due to the quantization of the voltage ADC of the IMNs

Timing

Regarding the timing of the analog operation of the MSC, the model of Fig. 3.26 is assumed. In order to investigate the phenomenon a SPICE model of the MSC for the 11-bus system has been created. All electrical parameters (conductances, parasitic capacitances, etc.) of the SPICE model are modifiable at will. Fig. 4.24 shows the overview of the topology. For each power system bus, there are two electrical nodes, “a” and “b”. The 4-pole nature of TPNs, and the 2-pole nature of OPNs are also evident. Changes in the IMN current injections are modeled using pulse current sources, seen on the bottom of the figure. The current sources complete the change in their output in a time t_{itr} (output slew).

A series of realistic current injection changes have been simulated in the SPICE model in order to investigate the RC phenomenon. All changes have been taken from realistic cases that occur in the simulation of transient in real power system scenarios. The same output slew rate t_{itr} has been assumed for all current DACs and the same capacitance has been assumed in every electrical node C . For putting the study into context, Fig. 4.25 provides a thumb-rule diagram for the order of magnitude of parasitic capacitances on different electronic design paradigms.

Figs. 4.26 show voltage transient results for $t_{itr} = 10 \text{ ns}$. The horizontal axes are time in s , the vertical axes are voltage in V. For each voltage trajectory, a “o” marker denotes the time instant at which the trajectory has reached the 63.2% of the final steady state value, a “+” marker denotes the time instant at which the 90% of the steady-state value has been reached, a “*”

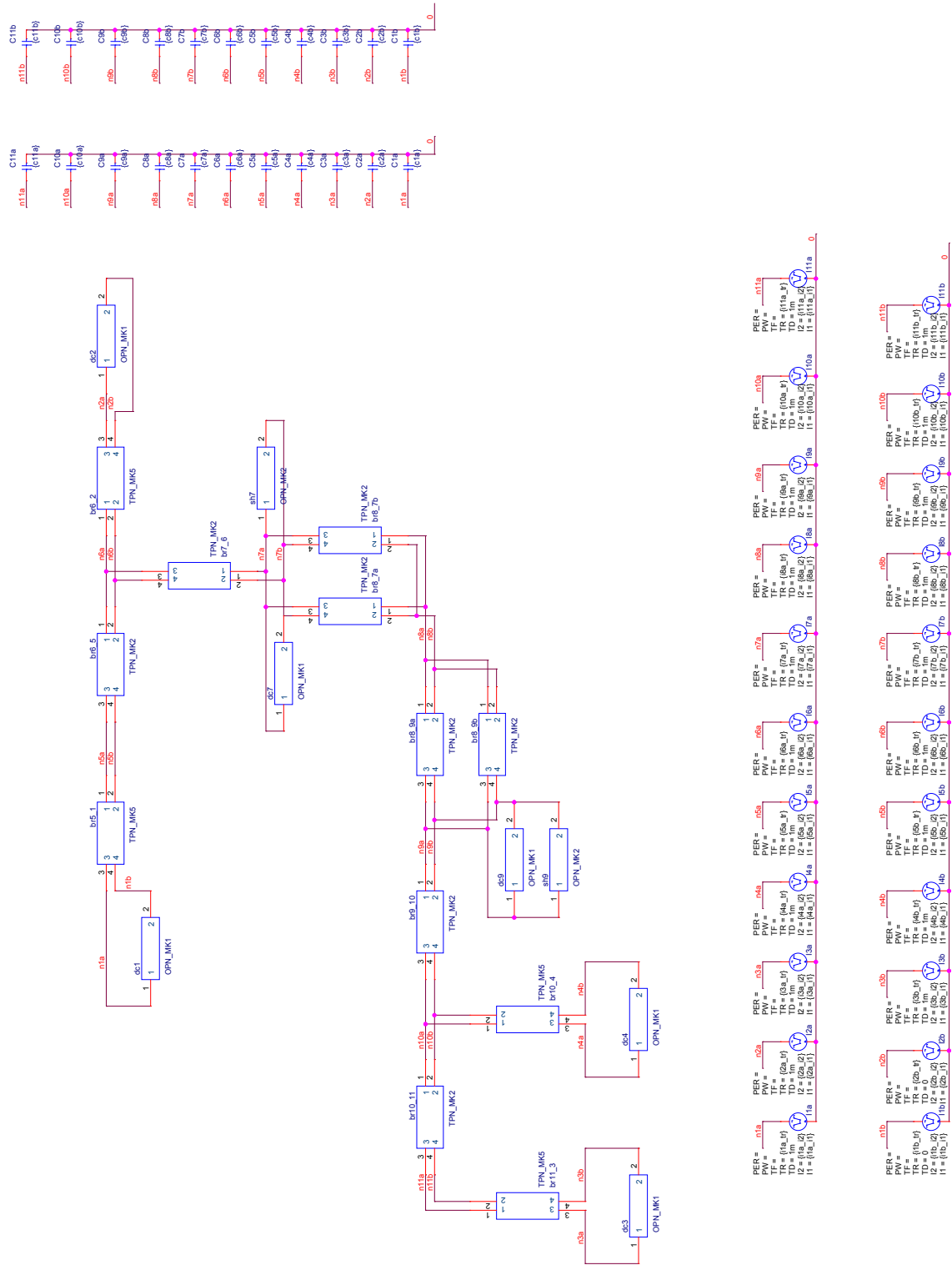


Figure 4.24 – The schematic of the electronic equivalent of an 11-bus system created in OrCAD Capture

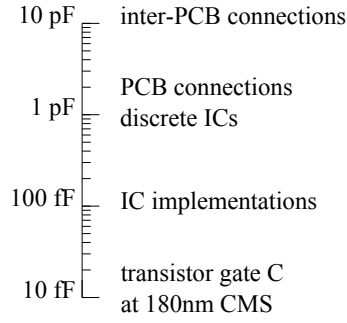


Figure 4.25 – Thumb-rule diagram for the order of magnitude of parasitic capacitances on different electronic design paradigms

marker denotes the time instant at which the 95% of the steady-state value has been reached, and a “x” marker denotes the time instant at which the 99% of the steady-state value has been reached.

As expected different node voltages have different delays. Most of trajectories follow an exponential trend without overshoot, hence the 1-pole RC model of section 3.3.1 of chapter 3 can provide an honest general approximation of the behavior. An interesting phenomenon is observed in Fig. 4.26d. Due to the fast responsiveness of the analog grid, the limit is the speed of the DACs. Indeed, all voltage curves follow a ramp 10 ns transition, which is exactly the t_{itr} .

DACs that are one order of magnitude faster $t_{itr} = 1 ns$ are assumed for the next set of tests. Figs. 4.27 summarize the results. The responses are quantitatively similar. The same “saturation” effect is observed in Fig. 4.27c, where the voltage responses simply follow the 1 ns ramp of the current injections. By retaining the assumption of uniform parasitic capacitances of $C = 10 fF$, faster responses can be obtained by having faster current DACs. Figs. 4.28 summarize the related results. For $t_{itr} = 200 ps$ the slew rate of the DAC starts to be comparable with the speed of an inverter in 180nm CMOS process.

Table 4.6 summarizes the results of the study. Empirical figures have been included for the times that the voltages reach the 63.2%, the 90%, the 95% and the 99% of the steady-state value (named $t_{0.63}$, $t_{0.90}$, $t_{0.95}$ and $t_{0.99}$ respectively).

The general trends that have been observed in the study of this subsection, can be assumed for systems of different size and topology. However the exact figures of the timings are strongly dependent on the exact implementation of the MSC. Additionally they are affected by the actual values of the reconfigurable conductances (potentiometers and VCCS). Due to the latter, the RC phenomena are different for different power systems, since this results into different conductances being mapped onto the electronics. Moreover, even for the same topology, different ρ_Y mapping ratio results in different electronic conductances through the value mapping and hence affects the RC timings. For these reasons, the figures presented hereinabove, should be taken as general rules of thumb for the design process and not as strict metrics.

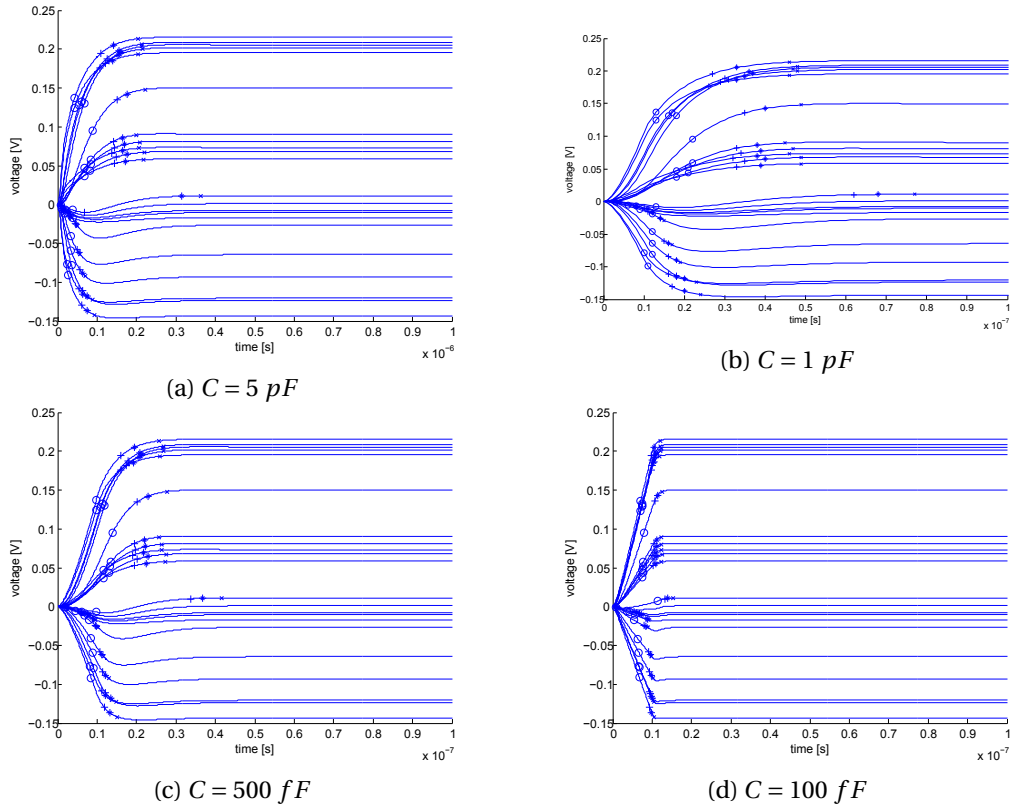


Figure 4.26 – Transients of the node of voltages of the MSC-mapping of the 11-bus system using a current source slew rate of $t_{itr} = 10 \text{ ns}$

Table 4.6 – Empirical settling times for different parasitic capacitances and current sources slew rates

t_{itr}	C	$t_{0.63}$	$t_{0.90}$	$t_{0.95}$	$t_{0.99}$
10 ns	5 pF	47ns	88ns	140ns	202ns
10 ns	1 pF	14ns	26ns	33ns	46ns
10 ns	500 fF	10ns	16ns	19ns	26ns
10 ns	100 fF	7ns	9.8ns	10.5ns	12ns
1 ns	100 fF	1.36ns	2.59ns	3.24ns	4.49ns
1 ns	50 fF	0.98ns	1.60ns	1.93ns	2.57ns
1 ns	10 fF	710ps	980ps	1.05ns	1.20ns
500 ps	10 fF	400ps	545ps	610ps	750ps
200 ps	10 fF	195ps	320ps	386ps	515ps

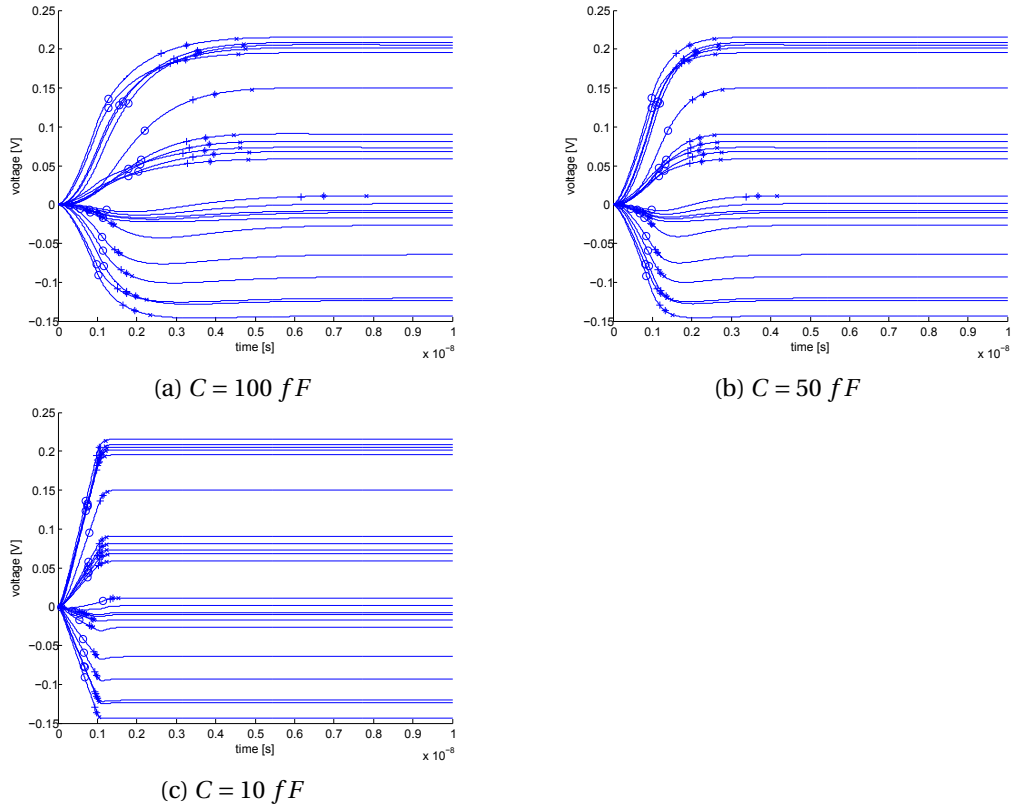


Figure 4.27 – Transients of the node of voltages of the MSC-mapping of the 11-bus system using a current source slew rate of $t_{itr} = 1 \text{ ns}$

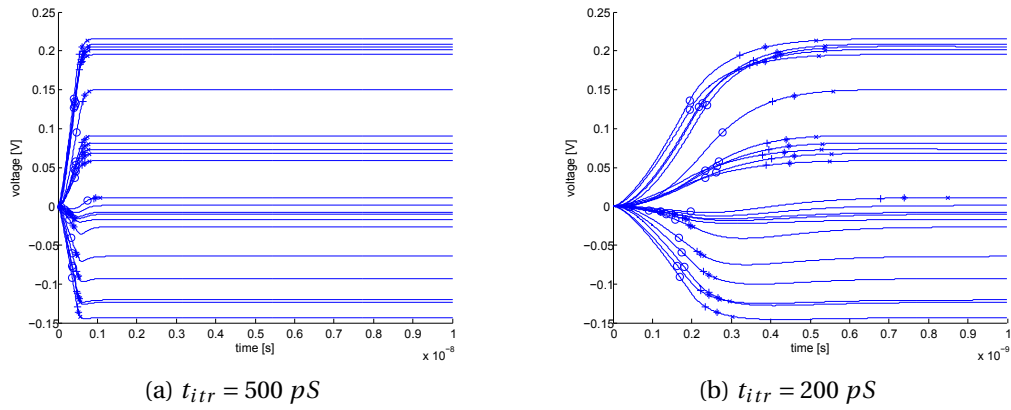


Figure 4.28 – Transients of the node of voltages of the MSC-mapping of the 11-bus system assuming uniform parasitic capacitances of $C = 10 \text{ fF}$

Table 4.7 – Free design parameters of the MSC and their effect in the operation

Free design parameter	Description	Effect	Rule of thumb design rule
M_G	Bit resolution of the reconfigurable conductances	Affects overall accuracy through E_I of (4.24) and ϵ_{V1} of (4.26)	$M_G \geq 12$
M_I	Bit resolution of the current DACs of IMNs	Affects overall accuracy through ϵ_I of (4.25) and ϵ_{V1} of (4.26)	$M_I \geq 16$
M_V	Bit resolution of the voltage ADCs of IMNs	Affects overall accuracy through ϵ_{V2} of (4.27)	$M_I \geq 12$
tol_{GFS}	Analog inaccuracy of the reconfigurable conductances	Affects overall accuracy through E_I of (4.24) and ϵ_{V1} of (4.26)	see Fig. 4.22
tol_{IFS}	Analog inaccuracy of the current DACs of IMNs	Affects overall accuracy through ϵ_I of (4.25) and ϵ_{V1} of (4.26)	see Fig. 4.22
C	Assumed uniform parasitic capacitance of electrical nodes of the MSC	Affects the time delay due to RC phenomena	see table 4.6
t_{itr}	Output slew rate of the current DACs	Affects the time delay due to RC phenomena	see table 4.6

4.4.3 Integration into the RAMSES flow

An arrangement similar to the one of Fig. 4.1 has been used to test the proposed architecture. The RAMSES engine (ver. 3.13) resided on a conventional modern desktop computer (Intel Core i7 2x2.80GHz, 8.0 GB RAM, Windows 7 64b). An emulator of the proposed MSC was created in MATLAB. The flow of the RAMSES simulator was interfaced with the MSC emulator according to Fig. 4.2. Calling of the MSC functionality was done using the software interface of Table 4.5.

The free design parameters of the MSC and their effect in the operation of it are summarized in table 4.7. When integrated into RAMSES, all these parameters affect the functioning of the MSC *as a component of the RAMSES flow*. This effect is examined in this section.

A three-cycle (60 ms) three-phase fault is applied on bus #7 of the 11-bus system of appendix A at time $t_{\text{fault}} = 0.5$ s. The resulting voltage on the faulted bus is as shown in Fig. 4.29. This result has been obtained by using the RAMSES software using a variable time step $h = 1 - 10$ ms and the BDF numerical integration algorithm.

Imperfect linear system solving in the internal (Newton) iterations cannot lead to inaccuracy of the solution for a given time step by the Newton scheme. This is because the Newton scheme stops when the zero of the non-linear function is found with a specific tolerance anyway. Imperfect solution of the linear sub-problem can affect the scheme in two ways.

- It can result in a higher number of Newton iterations for a given time step.

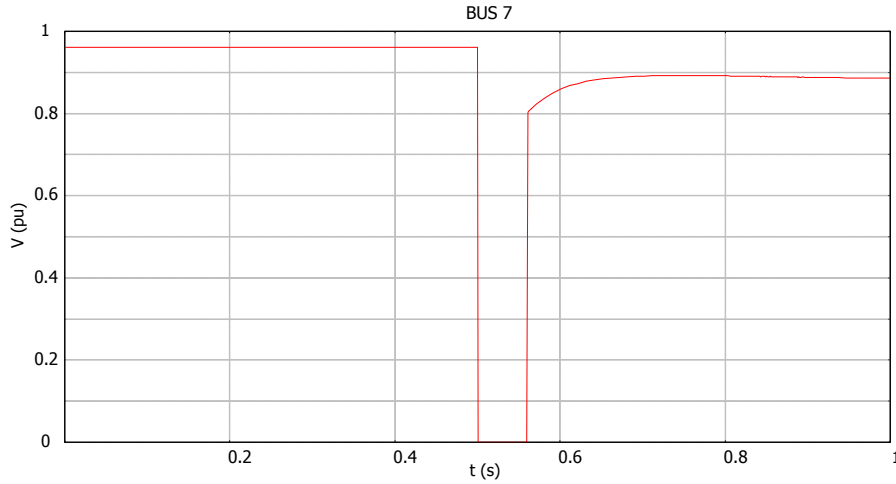


Figure 4.29 – Voltage of bus #7 of the 11-bus system after a 60 ms 3- ϕ fault on the same bus

- It can result in a lower steps selected by the variable time step mechanism.

Both the above have a negative effect in the overall number of total invocations of the linear system solving module. This increase has a potentially detrimental effect in the overall speed of the simulation. This will depend on the speed of linear system solving: if the speedup in the module alone is significant enough, then the higher number of operations can be counterbalanced and justified.

Validation of the design guidelines

A set of thumb-rule design guidelines have been put forth in previous section concerning the bit resolution of the conductances, the current DACs, and the voltage ADCs. A series of tests are conducted to verify the validity of these guidelines by integrating the MSC into RAMSES. Table 4.8 shows the related results. Six different design configuration have been created (A-F). The design parameters for each test case are shown in the respective rows of the table. The simulation that results in Fig. 4.29 was executed for each case. The number of the resulting total time steps for the time window $t = 0 \dots 1.0$ s are shown in column N_t . The last column N_{calls} shows the total number of MSC invocations for the same time window.

Case A has unrealistically good values for all design parameters (e.g. resolutions of 24 bits, no analog inaccuracy). The resulting N_t and N_{calls} values can be taken as reference for the MSC. In all other cases, there is one parameter, in bold typeface, which mildly violates the design guidelines of table 4.7. The adverse effect of this violation is clearly seen on the last column.

Fig. 4.30 visualizes the phenomenon across time. The horizontal axis corresponds to simulated time and the vertical axis shows the number of times the MSC was invoked for the solution of the non-linear system at every time step. This corresponds to the number of (internal) Newton iterations for each time step. In most cases the number of N_{calls} significantly increases while

Table 4.8 – Test cases to validate the design guidelines of table 4.7

test case	M_G	M_I	M_V	$tol_{G_{FS}}$	$tol_{I_{FS}}$	N_t	N_{calls}
A	24	24	24	0%	0%	106	164
B	6	24	24	0%	0%	106	169
C	24	15	24	0%	0%	106	334
D	24	14	24	0%	0%	106	784
E	24	24	13	0%	0%	106	286
F	24	24	12	0%	0%	2 (divergence)	46

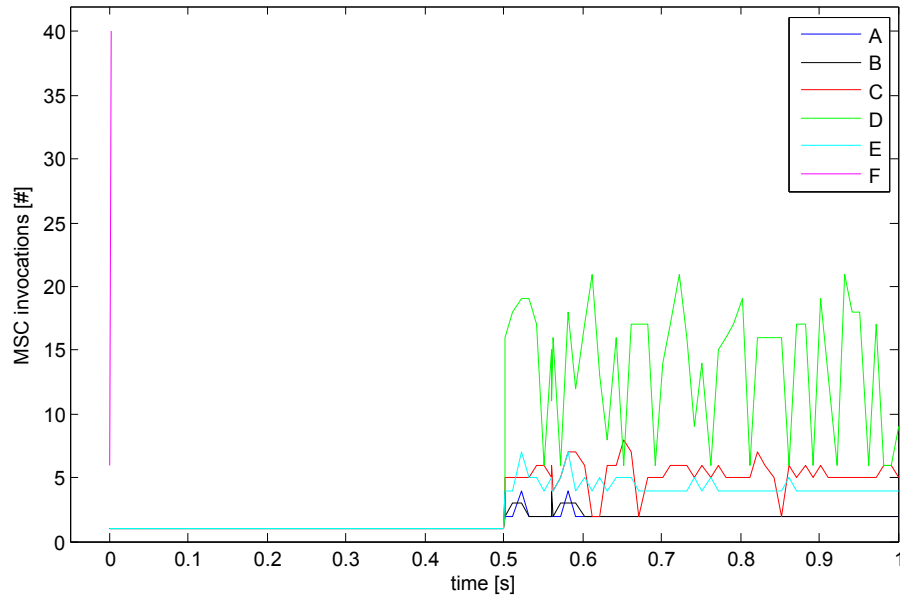


Figure 4.30 – MSC invocations for each time instant for the transient scenario on the 11-bus system

in one particular case (F), the simulation fails to converge, since the maximum number of Newton iterations (40) is reached in the second time step. From the above it can be concluded that a potential MSC implementation should respect the guidelines discussed previously.

Exploration of a reasonable design space

A thorough analysis of the design space has been conducted for different bit resolutions and analog inaccuracies. The limits of this design space are a combination of the design guidelines proposed and validated in previous sections and implementation realism, i.e. resolutions only up to an implementable limit have been considered. The design space is as follows.

- $M_I = \{16, 18, 20\}$
- $M_V = \{16, 18, 20\}$

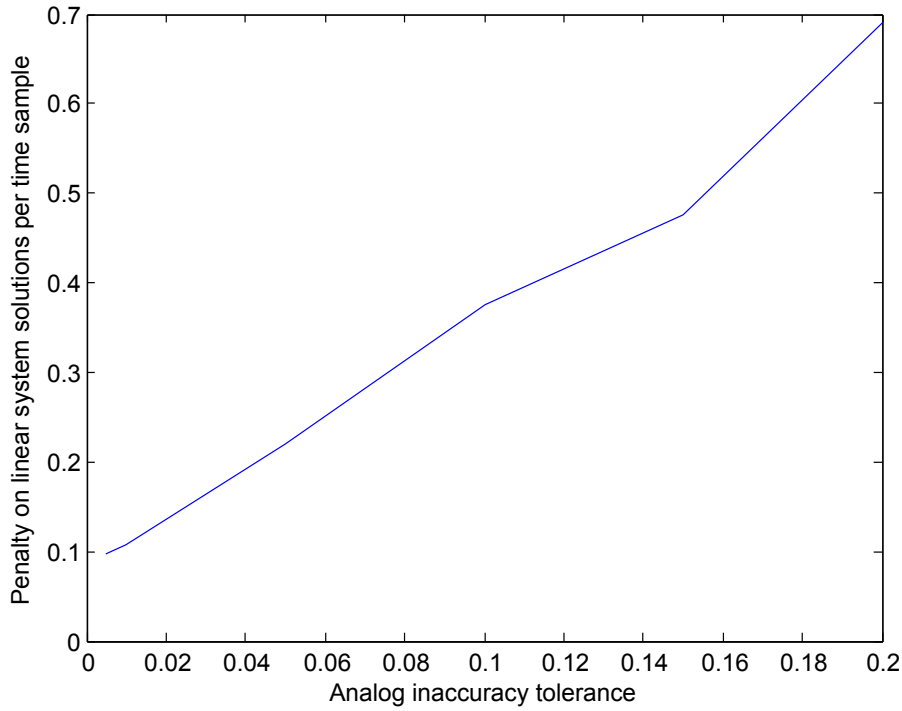


Figure 4.31 – Average penalty on the number of Newton (internal) iterations for one time (external) iteration

- $M_G = \{12, 14, 16, 18\}$
- $tol = tol_{GFS} = tol_{IFS} = \{0.5\%, 1\%, 5\%, 10\%, 15\%, 20\%\}$

The same scenario of the previous subsection has been considered. Multiple runs have been conducted in order to cancel out of probabilistic outlier results.

Fig. 4.31 shows the penalty that is incurred on the number of times the linear system is solved, for different accuracy levels (horizontal axis). The software solution is taken as a reference. Results are averaged for all other free design parameters, M_I , M_V , M_G . For example, a value of +0.4 in the vertical axis, means that for the corresponding analog inaccuracy tolerance in the horizontal axis, on average the Newton scheme needs 0.4 more iterations to converge; this results is averaged over any other parameter variation.

The study concerns the time window $t = 0.5 - 1$ s of the response shown in Fig. 4.29. This choice has to do with the fact that complex dynamics manifest in the network right after the three phase fault is applied. The resulting non-linear systems require more than one Newton iterations to converge. Hence the effect of the inaccurate MSC linear system solution is better demonstrated.

Tables 4.9, 4.10, and 4.11 show the effect of analog inaccuracy with respect to the bit resolution of current injections, voltage measurements, and reconfigurable conductances respectively.

Table 4.9 – Effect of analog inaccuracy to the average penalty on Newton iterations with respect to the bit resolution of the current injections

M_I	tol						ave	inc. gain
	0.5%	1%	5%	10%	15%	20%		
16	0.267	0.261	0.466	0.742	0.881	1.258	0.646	
18	0.012	0.029	0.097	0.194	0.278	0.437	0.175	-0.471
20	0.012	0.029	0.094	0.188	0.272	0.378	0.162	-0.012

Table 4.10 – Effect of analog inaccuracy to the average penalty on Newton iterations with respect to the bit resolution of the voltage measurements

M_V	tol						ave	inc. gain
	0.5%	1%	5%	10%	15%	20%		
16	0.142	0.150	0.261	0.397	0.508	0.715	0.362	
18	0.012	0.088	0.193	0.380	0.454	0.645	0.308	-0.054
20	0.012	0.082	0.204	0.349	0.469	0.713	0.313	+0.005

For each table the respective bit resolution and the analog accuracy are varied while all other design parameters are held constant. Numbers correspond to the average penalty on the number of Newton (internal) iterations for one time (external) iteration, similar to Fig. 4.31. The last column of the tables, presents the gain by increasing (moving down the table) the bit resolution of the respective element. A minus sign correspond to a *decrease* of the penalty, and hence to a “gain” from a performance point of view.

From the tables the following observations can be drawn.

- The most influential parameter in the accuracy of the MSC is the resolution of the current injectors M_I .
- For every element gains in the increase of the bit resolution fade out after a certain level.

Table 4.11 – Effect of analog inaccuracy to the average penalty on Newton iterations with respect to the bit resolution the reconfigurable conductances

M_G	tol						ave	inc. gain
	0.5%	1%	5%	10%	15%	20%		
12	0.070	0.109	0.235	0.387	0.455	0.650	0.318	
14	0.088	0.103	0.214	0.356	0.457	0.784	0.334	+0.016
16	0.136	0.107	0.202	0.372	0.494	0.704	0.336	+0.002
18	0.095	0.107	0.226	0.385	0.502	0.626	0.323	-0.012
20	0.042	0.067	0.205	0.340	0.386	0.564	0.268	-0.055

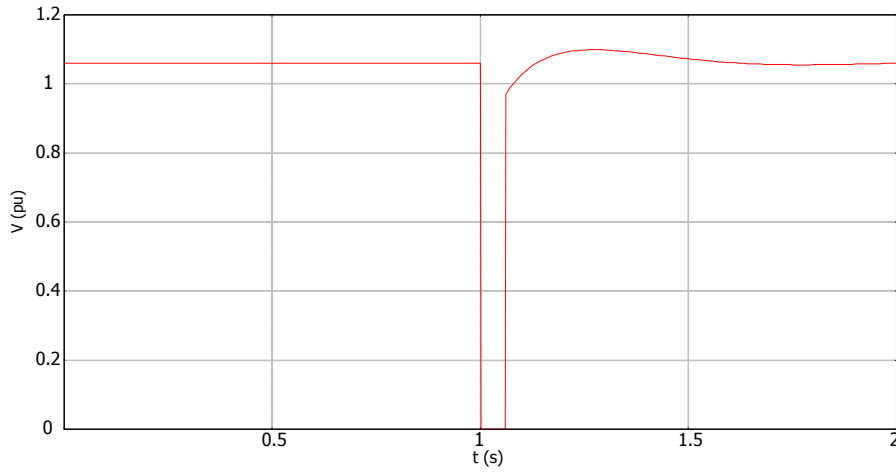


Figure 4.32 – Voltage of bus #4072 of the 77-bus system after a 60 ms 3- ϕ fault on the same bus

- The bit resolution of the reconfigurable conductances M_G seems to only have a minor influence on the overall accuracy. Given the “implementation cost” of having a higher M_G this observation can yield significant resources savings in a potential design.
- An increase in analog inaccuracy has an almost direct impact on the performance penalty - see Fig. 4.31. A quasi-linear sensitivity of 0.03 iteration increase for one % of analog inaccuracy increase has been observed.

A proposed design choice and validation on a larger system

The actual design choice for all design parameters will depend on their *implementation cost*. The latter can be anything like more, larger, more complex, more expensive electronics. As a realistic validation case the following has been selected.

- $M_I = 18$
- $M_V = 18$
- $M_G = 14$

The system is used to simulate a scenario on a much larger test case, the 77-bus system. A three-cycle (60 ms) three-phase fault is applied on bus #4072. Fig. 4.32 shows the voltage trajectory of the faulted bus. The fault is applied at $t = 1.0$ s and the time window of the simulation is $t = 0.0 - 2.0$ s.

The MSC configuration proposed above has been tested in the same simulation for two accuracy levels, $tol_{G_{FS}} = tol_{I_{FS}} = 5\%$ and $tol_{G_{FS}} = tol_{I_{FS}} = 1\%$. Fig. 4.33 presents the number of linear system solution operations across different time steps for the full-software simulator,

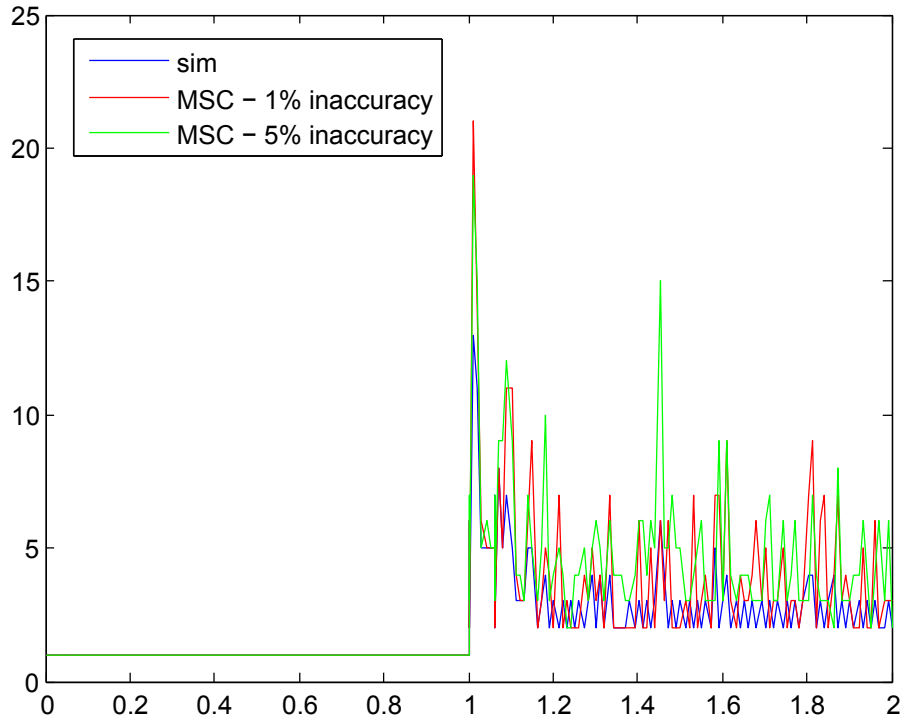


Figure 4.33 – MSC invocations for each time instant for the transient scenario on the 77-bus system

and for the MSC using the two accuracy levels. There are $N_{Is} = 406$ linear system solutions in the fully software run. There are $N_{MSC} = 615$ MSC invocations for the 5% accuracy case and $N_{MSC} = 538$ for the 1% case.

The time profiling of the software solution as given by RAMSES is as follows.

```

a: Local system building factorization and Schur complement 0.0386s (23.47 %)
  b: Reduced system eliminations & factorization (network) 0.0062s ( 3.76 %)
    c: Reduced system solution (network) 0.0056s ( 3.39 %)
      d: Injector evaluation and solution 0.0611s (37.17 %)
        e: Convergence checks at end of each N-I 0.0077s ( 4.67 %)
          f: Discrete events and other at end of time step 0.0039s ( 2.35 %)
            g: Discrete controller computation (DCTL) 0.0006s ( 0.35 %)
              h: Time step initialization 0.0316s (19.24 %)
                i: Remaining time 0.0092s ( 5.61 %)
                  Total elapsed time 0.1643s

```

The factorization (LU) and the solution of the factorized system of (4.5) is done in (b) and (c) respectively. An estimate of 30% of the time spent in (a) concerns the \tilde{D} in (4.5). This estimate comes from an internal profiling of RAMSES. Hence, an MSC invocation corresponds to the times of (b) and (c) and part of the time of (a). In this example this is $t_{sw} = 23.38 \text{ ms}$.

The time profiling of an MSC invocation is as explained in section 4.3.6.

$$t_{MSC} = \underbrace{t_1 + t_8}_{t_{comm}:communication} + \underbrace{t_2 + t_3 + t_4 + t_5 + t_7}_{t_{iop}:internal\ MSC\ ops.} + \underbrace{t_6}_{t_{RC}:RC\ wait} \quad (4.38)$$

The grouping of times shown above is natural in the sense that the three groups depend on different causes. The communication time depends on the speed B_{comm} of the PC_MB_IF as well as in the total amount of data that has to be transmitted D_{comm} . The bandwidth of a PCI Express 3.0 16x is assumed, $B_{comm} = 126$ Gbit/s. The total number of data that needs to be communicated at each MSC invocation can be calculated from table 4.5. $N = 77$ is the number of buses, $M = 42$ is the number of bus injectors for the example in question, and the size of double precision floating point format is 64 bits.

$$D_{comm} = (1 + 4 \cdot M + 4 \cdot N) \cdot sizeof(double) + (1 + 2 \cdot N) \cdot sizeof(bool) = 30683 \text{ bits} \quad (4.39)$$

The above give an estimate for the communication time.

$$t_{comm} = \frac{D_{comm}}{B_{comm}} \approx 400 \cdot ns \quad (4.40)$$

The exact time for the internal MSC operations cannot be determined unless the exact architecture and program flow of the digital part of the computer is known. In any case it holds

$$t_{iop} = N_{op} \cdot t_{clk} = \frac{N_{op}}{f_{clk}}. \quad (4.41)$$

Where N_{op} is the number of operations and f_{clk} is the clock cycle of the digital part of the MSC. Naturally, N_{op} is proportional to the size of the system, since values concerning the elements of the system are moved across the MSC digital hierarchy (see Figs. 4.17 and 4.15). For the sake of quantifying t_{iop} , it is assumed that

$$N_{op} \approx 3 \cdot 2 \cdot N + 4 \cdot M + \underbrace{50}_{auxiliary} = 680. \quad (4.42)$$

This rough assumption is in line with table 4.5. An auxiliary slack of 50 has been added to account of tasks that are necessary to the digital functioning of the MSC. Additionally, a digital clock frequency of $f_{clk} = 1 \text{ GHz}$ is assumed. With current technology, higher frequencies can easily be achieved in speed-optimized digital ICs. With these assumptions t_{iop} is as per (4.41).

$$t_{iop} \approx 680 \text{ ns} \quad (4.43)$$

For the waiting time, table 4.6 is pertinent. The waiting time has to be selected according to the t_{itr} and C parameters of the final implementation. Given that figures on the table only show an order of magnitude and not an exact result, a sufficient multiplicative security margin should be taken. For the sake of quantification, in this example $t_{itr} = 1 \text{ ns}$ and $C = 100 \text{ fF}$ are assumed. Then the waiting time is set to be times ten the $t_{0.95}$ figure of the table.

$$t_{RC} = \underbrace{10}_{\text{security margin}} \cdot 3.24 \text{ ns} \approx 35 \text{ ns} \quad (4.44)$$

The total time of an MSC invocation would then be as follows.

$$t_{MSC} = t_{comm} + t_{iop} + t_{RC} \approx 1.2 \text{ us} \quad (4.45)$$

This figure can be multiplied by the total number of MSC invocations in each case to yield an estimate for the total time spent for the MSC acceleration.

$$\text{For the 5\% accuracy case: } t_{hw} = N_{MSC} \cdot t_{MSC} = 615 \cdot 1.2 \text{ us} = 738 \text{ us} \quad (4.46)$$

$$\text{For the 1\% accuracy case: } t_{hw} = N_{MSC} \cdot t_{MSC} = 538 \cdot 1.2 \text{ us} = 646 \text{ us} \quad (4.47)$$

These figure have to be compared with the time that is spent in equivalent tasks in the full-software solution $t_{sw} = 23.38 \text{ ms}$. For the linear algebra part only, this implies the following speedup.

$$S_{LA} = \frac{t_{sw}}{t_{hw}} = \frac{23.38 \text{ ms}}{646...738 \text{ us}} \approx 31.7...36.2 \quad (4.48)$$

In this particular example, the MSC is two orders of magnitude faster than conventional software. If incorporated into the total timing of the solver, this would result in the following speedup.

$$S_{tot} = \frac{T_{old}}{T_{new}} \approx \frac{164.3 \text{ ms}}{141.6 \text{ ms}} \approx 1.16 \quad (4.49)$$

As said, the figures that are showcased here can be exactly known only after an implementation of the MSC. It is up to the final designer of the system to make all engineering choices that best fit his implementation needs. As shown hereinabove, by making some (moderate) reasonable assumptions, very significant performance benefits can be expected from the architecture. The benefit in the overall solution time, is naturally limited by the share of linear system solving in the total time. It may then be beneficial to design power system algorithms that intentionally perform more linear system solving operations, given that the latter are computationally cheaper through MSC.

4.5 Conclusions

In this section, a mixed-signal computer, the MSC, has been presented that is dedicated to linear algebra operations. To the knowledge of the author the MSC is the most advanced and versatile linear-algebra enabled computer that has been proposed. This work, is also the most thorough presentation of the (linear algebra) computing properties of analog electronics, again, to the knowledge of the author. In this chapter all the design-analysis tools for a future actual implementation of the MSC are contained.

Naturally, the architecture as well as its analysis largely draws upon concepts presented in chapter 3. Structural reconfigurability is greatly enhanced mainly due to the intelligent interconnection scheme of Figs. 4.15-4.17. Much wider value flexibility is offered due to the use of VCCSs in the TPNs and OPNs and the communication bottleneck has been lifted by connecting the MSC directly to the motherboard of the host PC through PC_MB_IF - see scheme in Fig. 4.1. It can therefore be concluded that the design goals have been met since the virtues of the original have been retained, while all of the limitations identified in section 3.5.2 have been overcome.

5 Conclusions

It is out of the scope of this work to question the established computing paradigm that has been shaping the world for the past sixty years. Digital computing is here to stay for a reason. This thesis is merely a look-back to analog computing *given* the advances in modern electronics. It has been quite some time that such a retrospective research has been conducted. To the knowledge of the author, this is the most thorough study on the properties of analog computing for linear algebra purposes to this day. Due to the affinity of analog electronic circuits with power system topologies, analog computing has been applied to problems in the power system domain.

The realization presented in chapter 3, termed the Field Programmable Power System Network (FPPNS), is the most advanced computer in the world in this aspect. Performance-wise tremendous benefits have been noted. Benefits come from both the analog part that handles the linear algebra computations and from the dedicated digital part that handles the solution of the differential algebraic equations of the simulation formulation.

Regarding the analog part, the most important lesson learned from the development and the use of the current prototype is that by definition analog computing cannot match the unrivaled reconfigurability and functional completeness of digital architectures. To put it differently, an analog computing setup is only good for what it has been *exactly* pre-configured for.

A further problem is that virtually all modern computing platform are either completely digital, or at least digital-computer-oriented. Hence for an analog computer to be able to coexist in such a digital ecosystem, bidirectional analog-to-digital electronic interfaces are required. Unsurprisingly enough the latter can be equally as complex to build and operate as the analog computing core itself.

The future computer of chapter 4, termed Mixed-Signal Computer (MSC), solved FPPNS limitations in a conceptual level. Three drastic changes in the design philosophy were key to that.

- The analog computer was brought closer to the digital host PC - see Fig. 4.1. Then the analog computer can operate as a dedicated accelerator while general purpose tasks are left to the host PC.
- Active electronics (e.g. VCCSs) have been used in the design of analog processing units. This is more “expensive” from an implementation point of view, but leads to much less computing constraints for the platform.
- An elaborate interconnection fabric has been proposed - see Figs. 4.15-4.17. This largely reduces most topological limitations, and hence greatly enlarges the set of compatible matrices.

The principle of *coherency between a computing platform and the problem* has been considered in every step of the hardware design process. The exact requirements were drawn from the RAMSES power system simulator software which has been used as an application to host MSC-acceleration. Exact evaluation of the MSC would require an implementation, which was out of the scope of this work. Instead, all the *tools* necessary for such an implementation have been provided in this work. Exemplary sample results show very significant performance benefits for the linear algebra operations, which translate in gains in the overall performance of the analysis.

5.1 Future work

In this section future extensions to this work are proposed. It is separated in two subsections, one dedicated to the already existing prototype of chapter 3 and another dedicated to the concept of chapter 4.

5.1.1 On the existing solver

Upgrade the communication system. In the current prototype a USB 2.0 connections is used. By upgrading it to a more advanced version, e.g. USB 3.0 or 3.1, significant gains could be made in the communication time, which is currently a major performance bottleneck.

Implement other power system applications. This could include power flow, quasi-steady state simulation, etc. For every case, the algorithm will have to be designed according to the capabilities of the FPPNS. In cases where repeated linear system solutions are required, e.g. in many voltage stability analysis algorithms, significant performance improvements can be attained.

Re-implement the system using higher scale integration. In the current prototype discrete IC components have been used on multiple PCBs. By moving towards higher integration, many benefits can be expected, e.g. lower size, lower consumption, and less parasitics. Especially

interesting would be the integration of the analog part of the FPPNS into ICs.

Implement an implicit integration algorithm. A implicit scheme would normally result in a non-linear system since it involves future values of the dynamic variables. The resulting linear system can be solved with a Newton-scheme, similar to the one used in RAMSES. The Jacobian involved in the Newton scheme is very much related to the admittance matrix of the topology, which can be handled in the analog part of the FPPNS. Such a solution scheme is quite different from the current partitioned scheme that uses explicit integration - see Fig. 3.5. Hence, significant changes would be required on the digital side of the FPPNS as well.

Integrate the FPPNS into the software framework of an existing power system simulation package. The software API that wraps the hardware accelerator functionality already exists in the `elab-tsaot`. Minor changes in the flow of the host software as well in the FPPNS API might be required. On the software side, the simplifying assumptions required by the FPPNS should be taken into account. This integration procedure, is similar to the what is proposed in chapter 4, i.e. the integration of the MSC into the RAMSES flow.

5.1.2 On the conceptual future solver

Implement the MSC. All the tools that are required for the analysis of a potential MSC implementation have been introduced in chapter 4. The implementation architecture and underlying technology will determine all the final parameters such as the parasitic capacitances, hence the design can be made accordingly. The proposal of the author is to first implement a Local Cell with relatively small node capacity. Then if functionality of the LC prototype is validated, the LC can be reproduced and the global architecture can be extended as per Fig. 4.17.

Use the MSC outside the power system domain. Since the core of the MSC functionality is linear algebra operations, it can be used as a linear algebra accelerated, even outside the power system domain. Theoretically, arbitrary matrices can be mapped on the reconfigurable analog fabric and the required operation can be performed through IMN injections and measurements. This might create some complications, since the analog part of the MSC is tailored for power system domain operations and it has been designed by taking into account the analogy of table 2.3. There are many case where similar analogies can be found, e.g. in cases where the physical system underlying the matrix is a flow network. In the general case, this cannot be guaranteed and hence the matrix might not be mappable to the MSC.

Extend the MSC functionality to solve (4.5c). As seen in the time profiling of the RAMSES flow in section 4.4.3 a significant amount of time is spend in “Injector evaluation and solution” (approx. 37%). This concerns the factorization and solution of the linear systems of (4.5c). The

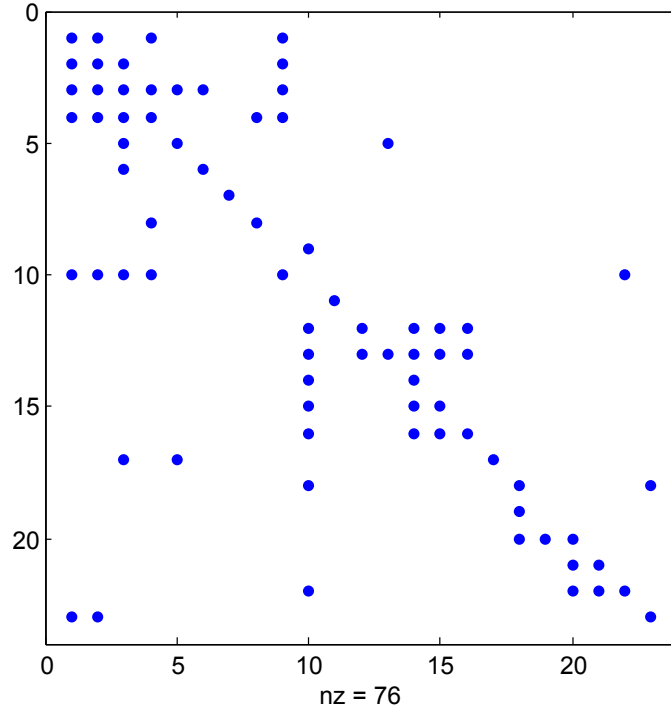


Figure 5.1 – Most generic sparsity pattern of the A_i matrix of injectors

latter refer to the dynamic injectors to the grid and its controllers. The sparsity pattern of the A_i matrix for a synchronous machine is shown in Fig. 5.1. This is a result that comes from the differential equations that govern the non-linear behavior of the injector. For the same type of equipment, e.g. synchronous machine, wind turbine, etc., the size of the matrix is always the same. The same holds for the non-zero pattern. This is due to the ordering of the equations and the dynamic variables that is the same in RAMSES across injectors of the same type.

It can be envisioned to create analog electronic entities, similar to TPNs and OPNs that map the matrices A_i of the injector linear systems. Since the non-zero pattern and the size is fixed, a corresponding electronic topology can be derived. A bank of such electronic entities can exist in the MSC and each injector can be mapped at one of them. At the corresponding point of the flow of RAMSES (see Fig. 4.2) the solution of the (4.5c) systems can be performed in full parallel on the MSC.

Modify RAMSES to exploit more the linear algebra prowess of the MSC. For every change in the \tilde{D} matrix, mathematically its $L \cdot U$ factors also change. However given the intensity of matrix decompositions, what is usually done is to

- update the \tilde{D} matrix as less often as possible, and also
- refactorize it as less often as possible.

This is referred to as a (very) dishonest Newton scheme. The penalty incurred by this strategy is that sometimes the convergence of the Newton solution is slowed down, and more iterations are needed.

This dishonest procedure is not required for the MSC since the “refactorization” of the matrix is not necessary. A change in the matrix translates into a closing or an opening of an electronic switch, or the reconfiguration of the value of one of the electronic components. This procedure is almost instantaneous. Hence it makes sense to update the \tilde{D} matrix as often as it would be deemed beneficial to reducing the total number of Newton (internal) and possible time (external) iterations.

In the same line, an entirely different flow for the host power system algorithm can be considered. The more it relies on linear algebra operations, the more the benefit from MSC acceleration.

5.2 Final discussion

A very good question would be the following: What is the outlook of analog computing in general, and in the linear algebra and power system domains in particular? The question becomes more important/interesting if the modern barriers in digital computers are considered, e.g. fatigue of the performance power increase law and power dissipation considerations.

It is the opinion of the author that the answer to these questions is related to three things,

- the existence of an *analogy* between the problem to be solved, and a real-life physical phenomenon that could potentially be reproduced on the analog computer
- the required “general purpose-ness”, and
- the level of required accuracy.

The first point determines whether analog computing is altogether available for a given problem. For the second point, the more hard the accuracy requirements are in a domain, the less pertinent analog computing is for it. Lastly for the third point, the more general purpose a computer needs to be, the less space there is for analog in its design.

Combining the above, it is here postulated that a future for analog computers may exist as *hardware accelerators for operations for which accuracy requirements are soft*. This is similar to the MSC that has been proposed in chapter 4 of this work. The power system domain is particularly promising for such architectures since there exists an analogy between the topologies of an analog circuit and a power system. It is up to advances of electronics to provide the level of required analog accuracy and low parasitic effects so as to make such a system a viable alternative to digital (co-)processors.

A Test power system topologies

An overview of the systems that have been as test cases throughout this work is presented in table A.1. These systems have all been regularly used for different studies. The concern systems of different complexity and simplifying assumptions. For example, the systems rc3, lanz5, fabre18, fabre36, padiyar39e and fabre59 all have the assumption $G \approx 0$. This assumption is valid for rougher studies in the transmission level. The systems fabre36 and fabre59 have been generated by modifying ieee30 and ieee57 respectively, by adding some branches of almost zero-impedance (node-duplication). The system padiyar39e is a version of padiyar39 with conductances neglected. These systems also have zero shunt components, no phase-shifting transformers. As a result their resulting $Y \equiv B$ matrices are actually singular, since for every one i of their rows it holds $\sum_{j=1}^N Y[i, j] = 0$.

For the aforementioned systems, the properties have been identified for the $N \times N$ complex admittance Y matrix. Table A.2 presents the respective results. In all cases, all matrices are structurally symmetric as discussed earlier and a corresponding column is omitted. Naturally for larger systems, more demanding property conditions are harder to be met, e.g. symmetry conditions are not met for the systems poland2383, poland2736, poland2746, poland3375 and peg15226 due to the presence of phase-shifting transformers.

Appendix A. Test power system topologies

Table A.1 – General overview of systems to be examined

Name-Size	Branch count	Shunt count	nnz	Sparsity
rc3 [299]	3	0	9	100%
gs4 [300]	4	0	12	75%
lanz5 [301]	5	0	15	60%
powell15 [37]	7	0	19	76%
ww6 [302]	11	0	28	77.8%
chow9 [303]	9	0	27	33.3%
wsc9 [304]	9	4	27	33.3%
kundur11 [305]	12	2	31	25.6%
ieee14 [306]	20	1	54	27.6%
fabre18 ¹	24	0	66	20.4%
ieee24 [307]	38	1	92	16.0%
alsac30 [308]	41	2	112	12.4%
ieee30 [306]	41	2	112	12.4%
fabre36 ²	46	0	128	9.9%
ne39 [309]	46	0	131	8.6%
padiyar39 [33]	46	0	131	8.6%
padiyar39e ³	46	0	131	8.6%
ieee57 [306]	80	3	213	6.6%
fabre59 ⁴	70	0	199	5.7%
nordic77 [310]	105	11	253	4.3%
le1101 [311]	110	0	315	3.1%
ieee118 [306]	186	14	476	3.4%
ieee300 [306]	411	29	1118	1.2%
poland2383 [303]	2896	0	8155	0.14%
poland2736 [303]	3269	1	9262	0.12%
poland2746 [303]	3279	0	9292	0.12%
poland3012 [303]	3572	9	10144	0.11%
poland3120 [303]	3693	9	10488	0.11%
poland3374 [303]	4161	9	11510	0.10%
peg15226 [40]	21492	1373	55060	0.02%

¹ fabre18 is a modified version of ieee14

² fabre36 is a modified version of ieee30

³ padiyar39e is a modified version of padiyar39

⁴ fabre59 is a modified version of ieee57

Table A.2 – General linear algebraic properties of the admittance matrix Y of power system test cases

Size	Symmetric	Skew-Hermitian	Normal	Real/Imaginary	Weakly diag. near-dominant	Irreducibly diag. near-dominant	Strictly diag. near-dominant	Complex line represent.
rc3	✓	✓	✓	imag	✓	✓	✓	✓
gs4	✓	-	-	-	✓	✓	✓	✓
lanz5	✓	✓	✓	imag	✓	✓	-	✓
powell15	✓	-	-	-	✓	✓	✓	✓
ww6	✓	-	-	-	✓	✓	✓	✓
chow9	✓	-	-	-	✓	✓	-	✓
wsc9	✓	-	-	-	-	-	-	✓
kundur11	✓	-	-	-	✓	✓	-	✓
ieee14	✓	-	-	-	✓	✓	-	✓
fabre18	✓	✓	✓	imag	✓	✓	-	✓
ieee24	✓	-	-	-	-	-	-	✓
alsac30	✓	-	-	-	✓	✓	-	✓
ieee30	✓	-	-	-	✓	✓	-	✓
fabre36	✓	✓	✓	imag	✓	✓	-	✓
ne39	✓	-	-	-	✓	✓	-	✓
padiyar39	✓	-	-	-	✓	✓	-	✓
padiyar39e	✓	✓	✓	imag	✓	✓	-	✓
ieee57	✓	-	-	-	✓	✓	-	✓
fabre59	✓	✓	✓	imag	✓	✓	-	✓
nordic77	✓	-	-	-	✓	✓	-	✓
le1101	✓	-	-	-	-	-	-	✓
ieee118	✓	-	-	-	-	-	-	✓
ieee300	✓	-	-	-	-	-	-	✓
poland2383	-	-	-	-	-	-	-	-
poland2736	-	-	-	-	-	-	-	-
poland2746	-	-	-	-	-	-	-	-
poland3012	✓	-	-	-	-	-	-	✓
poland3120	✓	-	-	-	-	-	-	✓
poland3374	-	-	-	-	-	-	-	-
peg15226	-	-	-	-	-	-	-	-

B Common modeling assumptions in power systems

In this appendix the mathematical background of some common power system modeling assumptions is presented.

Nomenclature

The following nomenclature conventions are adopted.

- Small letters are used to denote real or complex scalars, e.g. $x = 1.1$.
- Bold small letters are used to denote phasors, e.g. $\mathbf{x} = x \cdot e^{j\cdot\phi}$
- Capital letters are used for matrices and vectors of (real or complex scalars), e.g. $X_{123} = [x_1 \ x_2 \ x_3]^T$. Vectors are always considered as column vectors.
- A bar over a letter is used to denote quantities in the per unit system, e.g. $\bar{x} = x/x_{\text{base}}$.
- Combinations of the above are possible, e.g. a bold capital letter with a bar is used for a vector of phasors in the per unit system $\bar{\mathbf{X}}_{123} = [\bar{\mathbf{x}}_1 \ \bar{\mathbf{x}}_2 \ \bar{\mathbf{x}}_3]^T = [\bar{x}_1 \cdot e^{j\cdot\phi_1} \ \bar{x}_2 \cdot e^{j\cdot\phi_2} \ \bar{x}_3 \cdot e^{j\cdot\phi_3}]^T = [(x_1/x_{\text{base}}) \cdot e^{j\cdot\phi_1} \ (x_2/x_{\text{base}}) \cdot e^{j\cdot\phi_2} \ (x_3/x_{\text{base}}) \cdot e^{j\cdot\phi_3}]^T$.

B.1 Phasor representation

Let $x(t)$ be a time-domain waveform. According to the generalized averaging method [312], in the interval $\tau \in (t - T, t]$ the waveform can be approximated as follows.

$$x(\tau) = \sum_{k=-\infty}^{\infty} \langle x \rangle_k(t) \cdot e^{j\cdot k\omega_s\tau} \quad (\text{B.1})$$

Appendix B. Common modeling assumptions in power systems

Where $\omega_s = 2\pi/T$ is the sampling frequency and $\langle x \rangle_k(t)$ are the Short-Time Fourier series coefficients, given by

$$\langle x \rangle_k(t) = \frac{1}{T} \int_{t-T}^t x(\tau) \cdot e^{-j \cdot k \omega_s \tau} d\tau. \quad (\text{B.2})$$

The above is in essence a time-frequency representation of the original signal, as the signal in (B.1) is written as a sum of frequency-related ($k \cdot \omega_s$), time-varying coefficients. In power system literature these coefficients are called *dynamic phasors* and they are commonly used to obtain state-space models in which the phasors are the state variables [313, 314].

In the general case, when $x(t)$ is complex, the following holds.

$$\langle x \rangle_{-k}(t) = (\langle x^* \rangle_k(t))^* \quad (\text{B.3})$$

For real waveforms this simplifies further to

$$\langle x \rangle_{-k}(t) = (\langle x \rangle_k(t))^*. \quad (\text{B.4})$$

Then (B.1) becomes

$$x(\tau) = \langle x \rangle_0(t) + \sum_{k=1}^{\infty} \left[\langle x \rangle_k(t) \cdot e^{j \cdot k \omega_s \tau} + \left(\langle x \rangle_k(t) \cdot e^{j \cdot k \omega_s \tau} \right)^* \right] \quad (\text{B.5})$$

$$= \langle x \rangle_0(t) + 2 \cdot \sum_{k=1}^{\infty} \text{Re} \left\{ \langle x \rangle_k(t) \cdot e^{j \cdot k \omega_s \tau} \right\}. \quad (\text{B.6})$$

We assume a sinusoidal waveform.

$$x(t) = |x|(t) \cdot \cos(\omega(t) \cdot t + \phi(t)) \quad (\text{B.7})$$

Another common assumption is that the parameters of the waveform above are time-invariant.

$$|x|(t) = |x| \text{ const}$$

$$\omega(t) = \omega \text{ const}$$

$$\phi(t) = \phi \text{ const}$$

Then, for $\omega_s \equiv \omega$ it can be verified that

$$\langle x \rangle_1(t) = |x| \cdot e^{j \cdot \phi} \text{ const} \quad (\text{B.8})$$

$$\langle x \rangle_{k \neq 1}(t) = 0 \text{ const} \quad (\text{B.9})$$

It is this first coefficient that is used as the *phasor* of the signal, $\mathbf{x} := \langle x \rangle_1(t)$. Under normal circumstances a single phasor is enough to describe a signal of the real system. This phasor is the first coefficient of the Short-Time Fourier Series expansion of the signal [313].

The phasor representation of a signal is slowly varying compared to the signal itself given that the bandwidth of the latter is (usually much) smaller than its carrier frequency; for more on that and on the relation between a signal, its Hilbert transform and its phasor see also [315].

Phasors denote quantities that rotate in the counterclockwise direction with the nominal frequency of the system, e.g. 50 Hz for Europe. A common convention for the magnitude of the phasor is to represent the RMS maximum value of the positive half-cycle of the sinusoid [316]. An example of a phasor is presented hereunder.

$$x(t) = \sqrt{2} \cdot x \cdot \cos(\omega \cdot t + \phi_x) \rightarrow \mathbf{x} = x \cdot e^{j \cdot \phi_x}$$

The phasor form above is called the exponential form and it is equivalent to the rectangular form and the polar form, shown hereunder.

$$\mathbf{x} = x \cdot e^{j \cdot \phi_x} = \text{Re}\{\mathbf{x}\} + j \cdot \text{Im}\{\mathbf{x}\} = x \angle \phi_x \quad (\text{B.10})$$

B.2 Balance in electrical quantities

An arbitrary three-phase voltage vector is given by the following.

$$\mathbf{V}_{abc} = \begin{bmatrix} \mathbf{v}_a \\ \mathbf{v}_b \\ \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} v_a \cdot e^{j\phi_a} \\ v_b \cdot e^{j\phi_b} \\ v_c \cdot e^{j\phi_c} \end{bmatrix} \quad (\text{B.11})$$

According to the symmetrical components theory introduced by Fortesque in his seminal paper [317] a three-phase system can be resolved into three balanced systems. First the Fortescue operator α is defined, which denotes a positive phase shift of 120° .

$$\alpha := e^{j \cdot \frac{2\pi}{3}} \quad (\text{B.12})$$

Also

$$\alpha^2 = e^{j \cdot 2 \cdot \frac{2\pi}{3}} = -\alpha \text{ for a shift of } 240^\circ, \quad (\text{B.13})$$

$$\alpha^3 = e^{j \cdot 3 \cdot \frac{2\pi}{3}} = 1 \text{ for a shift of } 360^\circ. \quad (\text{B.14})$$

A transformation matrix is then defined as follows.

$$T_{0+-}^{abc} := \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \quad (\text{B.15})$$

Left multiplication with T_{0+-}^{abc} represents a change of the base of the vectors in \mathbb{C}^3 . So, T_{0+-}^{abc} can be thought of as a change of basis matrix from the symmetrical components to the three-phase domain. In base transformation matrices the subscript is the starting frame of reference and the superscript is the destination frame of reference.

Left multiplying with the inverse of the transformation matrix performs exactly the opposite

transformation, from the three-phase to the symmetrical components domain.

$$T_{abc}^{0+-} = T_{0+-}^{abc^{-1}} = \frac{1}{3} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \quad (\text{B.16})$$

The following also hold.

$$\text{for the transpose } T_{0+-}^{abc^T} = T_{0+-}^{abc} \quad (\text{B.17})$$

$$\text{for the complex conjugate } T_{0+-}^{abc*} = 3 \cdot T_{0+-}^{abc^{-1}} \quad (\text{B.18})$$

Transforming the arbitrary voltage vector of phasors the symmetrical components domain gives the following.

$$\mathbf{V}_{abc} = T_{0+-}^{abc} \cdot \mathbf{V}_{0+-} = T_{0+-}^{abc} \cdot \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_+ \\ \mathbf{v}_- \end{bmatrix} = \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_0 \\ \mathbf{v}_0 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_+ \\ \alpha^2 \cdot \mathbf{v}_+ \\ \alpha \cdot \mathbf{v}_+ \end{bmatrix} + \begin{bmatrix} \mathbf{v}_- \\ \alpha \cdot \mathbf{v}_- \\ \alpha^2 \cdot \mathbf{v}_- \end{bmatrix} = \mathbf{V}_0 + \mathbf{V}_+ + \mathbf{V}_- \quad (\text{B.19})$$

The *zero (or homopolar) sequence* consists of three phasors equal in magnitude and phase.

$$\mathbf{V}_0 := \begin{bmatrix} \mathbf{v}_{a0} \\ \mathbf{v}_{b0} \\ \mathbf{v}_{c0} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_0 \\ \mathbf{v}_0 \end{bmatrix} \quad (\text{B.20})$$

The *positive (or direct) sequence* consists of a balanced system of three phasors having the same sequence as the original signal. As the phasors are rotating counter-clockwise, \mathbf{v}_{b+} lags 120° behind \mathbf{v}_{a+} , and \mathbf{v}_{c+} lags 120° behind \mathbf{v}_{b+} (and 240° behind \mathbf{v}_{a+}). So the sequence is $a \rightarrow b \rightarrow c$.

$$\mathbf{V}_+ := \begin{bmatrix} \mathbf{v}_{a+} \\ \mathbf{v}_{b+} \\ \mathbf{v}_{c+} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_+ \\ \alpha^2 \cdot \mathbf{v}_+ \\ \alpha \cdot \mathbf{v}_+ \end{bmatrix} \quad (\text{B.21})$$

The *negative (or inverse) sequence* consists of a balanced system of three phasors having the

Appendix B. Common modeling assumptions in power systems

opposite sequence from the original signal. As the phasors are rotating counter-clockwise, \mathbf{v}_{c+} lags 120° behind \mathbf{v}_{a+} , and \mathbf{v}_{b+} lags 120° behind \mathbf{v}_{c+} (and 240° behind \mathbf{v}_{a+}). So the sequence is $a \rightarrow c \rightarrow b$.

$$\mathbf{V}_- := \begin{bmatrix} \mathbf{v}_{a-} \\ \mathbf{v}_{b-} \\ \mathbf{v}_{c-} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_- \\ \alpha \cdot \mathbf{v}_- \\ \alpha^2 \cdot \mathbf{v}_- \end{bmatrix} \quad (\text{B.22})$$

The base phasors of the symmetric sequences can be derived from the original signal as follows.

$$\mathbf{V}_{0+-} = T_{abc}^{0+-} \cdot \mathbf{V}_{abc} \quad (\text{B.23})$$

Note that for each of the sequences the three constituent phasors appear always together, e.g. for the positive sequence, \mathbf{v}_{a+} can never exist without \mathbf{v}_{b+} and \mathbf{v}_{c+} [316].

When a balanced operation is considered, amplitudes of the phasors of the three phases are equal, and angles are equidistant around a cycle of a full period 2π .

$$\begin{aligned} v_a &= v_b = v_c \\ \phi_b &= \phi_a - \frac{2\pi}{3} \\ \phi_c &= \phi_a - 2 \cdot \frac{2\pi}{3} \end{aligned} \quad (\text{B.24})$$

The three-phase vector can then be rewritten as follows.

$$\mathbf{V}_{abc} = v_a \cdot e^{j\phi_a} \cdot \begin{bmatrix} 1 \\ -\alpha \\ -\alpha^2 \end{bmatrix} = v_a \cdot e^{j\phi_a} \cdot \begin{bmatrix} 1 \\ \alpha^2 \\ \alpha \end{bmatrix} \quad (\text{B.25})$$

Applying the symmetrical components transformation to it, yields

$$\mathbf{V}_{0+-} = T_{abc}^{0+-} \cdot \mathbf{V}_{abc} = \dots = \begin{bmatrix} 0 \\ v_a \cdot e^{j\phi_a} \\ 0 \end{bmatrix} \Rightarrow \mathbf{v}_+ = \mathbf{v}_a = v_a \cdot e^{j\phi_a}, \mathbf{v}_0 = \mathbf{v}_- = \mathbf{0}. \quad (\text{B.26})$$

That is, under balanced conditions only the direct sequence is present and a single phasor can represent the whole three-phase voltage.

B.3 Symmetry in the network

In this section the effect of the symmetrical components transformation will be examined for the interconnecting building blocks of the network, i.e. the transmission lines. In a general three phase topology, the system is comprised by a set of connected three-port elements. The impedance parameters of a sample such element are given as follows

$$\begin{aligned} \mathbf{V}_{abc} &= Z_{abc} \cdot \mathbf{I}_{abc}, & \mathbf{V}_{abc}, \mathbf{I}_{abc} &\in \mathbb{C}^3 \text{ and } Z_{abc} \in \mathbb{C}^3 \times \mathbb{C}^3 \\ \begin{bmatrix} \mathbf{v}_a \\ \mathbf{v}_b \\ \mathbf{v}_c \end{bmatrix} &= \begin{bmatrix} z_{aa} & z_{ab} & z_{ac} \\ z_{ba} & z_{bb} & z_{bc} \\ z_{ca} & z_{cb} & z_{cc} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{i}_a \\ \mathbf{i}_b \\ \mathbf{i}_c \end{bmatrix} \end{aligned} \quad (\text{B.27})$$

The theory behind the analytical derivation of the elements of the matrix hereinabove, i.e. line parameters estimation, is beyond the scope of this work and the interested reader is referred to relevant bibliography [289, 318, 319]. In the above, the effect of the input in any phase (a , b or c) has an effect on the output of other phases, through the off-diagonal coupling terms of matrices Z_{abc} or Y_{abc} . In other words, all inputs are coupled with all outputs. In order to have decoupled input-output relation the relation matrix needs to be diagonal.

A common assumption in many analyses are that three-port elements in the system are balanced, and thus their impedance parameters have a fully symmetric matrix [318].

$$Z_{abc} = \begin{bmatrix} z & m & m \\ m & z & m \\ m & m & z \end{bmatrix} \quad (\text{B.28})$$

Matrices as the one above are a special case of circulant matrices. As such the eigenvectors

Appendix B. Common modeling assumptions in power systems

and eigenvalues can be easily calculated.

$$\begin{aligned}\lambda_0 &= z + m \cdot \omega_0 + m \cdot \omega_0^2, & u_0 &= [1 \quad \omega_0 \quad \omega_0^2]^T \\ \lambda_1 &= z + m \cdot \omega_1 + m \cdot \omega_1^2, & u_1 &= [1 \quad \omega_1 \quad \omega_1^2]^T \\ \lambda_2 &= z + m \cdot \omega_2 + m \cdot \omega_2^2, & u_2 &= [1 \quad \omega_2 \quad \omega_2^2]^T\end{aligned}\tag{B.29}$$

Where $\omega_k = \exp\left(j \cdot \frac{2\pi k}{3}\right)$ are the n^{th} roots of the unit circle. Finally

$$\begin{aligned}\lambda_0 &= z + 2 \cdot m, & u_0 &= [1 \quad 1 \quad 1]^T \\ \lambda_1 &= z - m, & u_1 &= \left[1 \quad \exp\left(j \cdot 2 \cdot \frac{2\pi}{3}\right) \quad \exp\left(j \cdot \frac{2\pi}{3}\right)\right]^T \\ \lambda_2 &= z - m, & u_2 &= \left[1 \quad \exp\left(j \cdot \frac{2\pi}{3}\right) \quad \exp\left(j \cdot 2 \cdot \frac{2\pi}{3}\right)\right]^T\end{aligned}\tag{B.30}$$

It can be seen that $\exp\left(j \cdot \frac{2\pi}{3}\right) \equiv \alpha$ is nothing but the Fortescue operator.

According to eigendecomposition principles Z_{abc} can be diagonalized as follows.

$$T^{-1} \cdot Z_{abc} \cdot T = Z_{diag} = \begin{bmatrix} \lambda_0 & & \\ & \lambda_2 & \\ & & \lambda_2 \end{bmatrix}\tag{B.31}$$

T is an invertible matrix, the columns of which are the right eigenvectors of Z_{abc} , and the corresponding diagonal entry in Z_{0+-} is the corresponding eigenvalue. It is easily seen that T coincides with the T_{0+-}^{abc} matrix that has been defined in the previous section. Through this observation, it can be seen that when the symmetrical components transformation is applied to balanced three-port impedance networks, it diagonalizes the impedance matrix of the network.

The diagonalized form of Z_{abc} is defined as the diagonal eigenvalue matrix.

$$Z_{0+-} \equiv \begin{bmatrix} z_0 & 0 & 0 \\ 0 & z_+ & 0 \\ 0 & 0 & z_- \end{bmatrix} := Z_{diag} = \begin{bmatrix} z + 2 \cdot m & 0 & 0 \\ 0 & z - m & 0 \\ 0 & 0 & z - m \end{bmatrix}\tag{B.32}$$

This is substituted in (B.27) to yield the decoupled equations.

$$\mathbf{V}_{abc} = T_{0+-}^{abc} \cdot Z_{0+-} \cdot T_{0+-}^{abc^{-1}} \cdot \mathbf{I}_{abc} \Rightarrow \mathbf{V}_{0+-} = Z_{0+-} \cdot \mathbf{I}_{0+-} \Rightarrow \begin{aligned} \mathbf{v}_0 &= z_0 \cdot \mathbf{i}_0 \\ \mathbf{v}_+ &= z_+ \cdot \mathbf{i}_+ \\ \mathbf{v}_- &= z_- \cdot \mathbf{i}_- \end{aligned} \quad (\text{B.33})$$

The last equation is the decoupled impedance parameters of the three port network. The usefulness of symmetrical components is that each sequence can be studied separately. Under balanced operating conditions it has been seen that homopolar and negative sequence phasors are zero, so only the positive sequence equation is relevant. The positive sequence impedance z_+ is what is normally available in power flow studies for the branches of the system.

In the normal non-degenerate case Z_{abc} is invertible $Y_{abc} = Z_{abc}^{-1}$. Then, a similar line can be followed for the admittance version of every equation. The relevant final equation would then be

$$\mathbf{i}_+ = y_+ \cdot \mathbf{v}_+. \quad (\text{B.34})$$

B.4 Power considerations

Let a single phase, line-to-neutral voltage be

$$v(t) = \sqrt{2} \cdot v \cdot \cos(\omega \cdot t + \phi_v) \rightarrow \mathbf{v} = v \cdot e^{j \cdot \phi_v} \quad (\text{B.35})$$

Let a single phase line current be

$$i(t) = \sqrt{2} \cdot i \cdot \cos(\omega \cdot t + \phi_i) \rightarrow \mathbf{i} = i \cdot e^{j \cdot \phi_i} \quad (\text{B.36})$$

Then the instantaneous power entering a single port network is as follows [295].

$$p_i = v(t) \cdot i(t) = \dots = v \cdot i \cdot \cos(\phi_v - \phi_i) \cdot [1 + \cos(2\omega t + 2\phi_i)] - v \cdot i \cdot \sin(\phi_v - \phi_i) \cdot \sin(2\omega t + 2\phi_i) \quad (\text{B.37})$$

Appendix B. Common modeling assumptions in power systems

In the above there are two distinct parts.

- $p_{ip}(t) := v \cdot i \cdot \cos(\phi_v - \phi_i)[1 + \cos(2\omega t + 2\phi_i)]$ is the *instantaneous active power*. This is the amount of power that is irreversibly consumed for the production of real work.
- $p_{iq}(t) := -v \cdot i \cdot \sin(\phi_v - \phi_i) \cdot \sin(2\omega t + 2\phi_i)$ is the *instantaneous reactive power*. This is a byproduct of the alternating nature of the system as it is the result of the component of current that is in quadrature with the voltage. The reactive power supports the transfer of active power over the transmission medium. It is temporarily stored in inductances and capacitances in the network and has a zero average value.

The $2\omega t$ terms in the above are zeroed if averages over one period $2\pi/\omega$. We define the (average) *active* and (peak) *reactive* power as follows.

$$p := v \cdot i \cdot \cos(\phi_v - \phi_i) \quad (\text{B.38})$$

$$q := -v \cdot i \cdot \sin(\phi_v - \phi_i) \quad (\text{B.39})$$

The instantaneous power then can be written as follows.

$$p_i(t) = p \cdot [1 + \cos(2\omega t + 2\phi_i)] + q \cdot \sin(2\omega t + 2\phi_i) \quad (\text{B.40})$$

Active power p is the average amount of work producing power delivered to the element in question over one period of the sinusoidal voltage. Reactive power q is the peak value of the instantaneous reactive power as defined before.

Note that neither p or q are sinusoidal. However we can make use of the phasors that were defined in the beginning of this subsection to define single-phase apparent power as follows.

$$s_{1\Phi} := \mathbf{v} \cdot \mathbf{i}^* = v \cdot i \cdot e^{j(\phi_v - \phi_i)} \quad (\text{B.41})$$

The above is only valid if the magnitude of the phasors \mathbf{v} and \mathbf{i} are the RMS values of the line-to-neutral voltage and line current respectively, as defined in (B.35) and (B.36). Of course (B.41) can also be written in rectangular or polar form, $s_{1\Phi} = p + j \cdot q = s_{1\Phi} \cdot e^{j\phi_s}$.

A similar expression applies to the averaged three-phase apparent power.

$$s_{3\Phi} = \mathbf{V}_{abc}^T \cdot \mathbf{I}_{abc}^* = \mathbf{v}_a \cdot \mathbf{i}_a^* + \mathbf{v}_b \cdot \mathbf{i}_b^* + \mathbf{v}_c \cdot \mathbf{i}_c^* \quad (\text{B.42})$$

For theories on the full three-phase *instantaneous* power the interested reader is referred to [320].

In case the signals are balanced, the above becomes

$$s_{3\Phi} = \mathbf{v}_a \cdot \mathbf{i}_a^* + \mathbf{v}_b \cdot \mathbf{i}_b^* + \mathbf{v}_c \cdot \mathbf{i}_c^* = \mathbf{v}_a \cdot \mathbf{i}_a^* + \alpha^2 \cdot \mathbf{v}_a \cdot (\alpha^2)^* \cdot \mathbf{i}_a^* + \alpha \cdot \mathbf{v}_a \cdot \alpha^* \cdot \mathbf{i}_a^* = 3 \cdot \mathbf{v}_a \cdot \mathbf{i}_a^* = 3 \cdot v_a \cdot i_a \cdot e^{j \cdot (\phi_v - \phi_i)}. \quad (\text{B.43})$$

If the symmetrical components transformation is used

$$s_{3\Phi} = \mathbf{V}_{abc}^T \cdot \mathbf{I}_{abc}^* = (T_{0+-}^{abc} \cdot \mathbf{V}_{0+-})^T \cdot (T_{0+-}^{abc} \cdot \mathbf{I}_{0+-})^* = \dots = 3 \cdot \mathbf{V}_{0+-}^T \cdot \mathbf{I}_{0+-}^*. \quad (\text{B.44})$$

The scalar factor $3 \times$ is produced by the matrix multiplication $T_{0+-}^{abcT} \cdot T_{0+-}^{abc*}$. It is reasonable if the symmetrical components are thought of as three sets of three balanced phasors, in total 9 phasors. In the case of balanced signals, the above becomes

$$s_{3\Phi} = 3 \cdot \mathbf{v}_+ \cdot \mathbf{i}_+^* = 3 \cdot v_+ \cdot i_+ \cdot e^{j \cdot (\phi_v - \phi_i)}, \quad (\text{B.45})$$

which coincides with (B.43), as $v_+ \equiv v_a$ and $i_+ \equiv i_a$.

Another comment on the factor $3 \times$ is that it signifies that the base transformation defined by T_{0+-}^{abc} is not power invariant.

$$T_{0+-}^{abcT} \cdot T_{0+-}^{abc*} = 3 \cdot I_3 \quad (\text{B.46})$$

Where I_3 is the 3×3 identity matrix. In order for the transformation to be power invariant a

Appendix B. Common modeling assumptions in power systems

new transformation matrix would have to be defined as follows [321].

$$T_{0+-}^{abc} = \frac{1}{h} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \quad (\text{B.47})$$

Then the inverse would be

$$T_{0+-}^{abc^{-1}} = \frac{h}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix}, \quad (\text{B.48})$$

and

$$T_{0+-}^{abc^T} \cdot T_{0+-}^{abc*} = \frac{3}{h^2} \cdot I_3. \quad (\text{B.49})$$

For the $h = \sqrt{3}$ the transformation is power invariant. In this case, T_{0+-}^{abc} is also unitary (i.e. the complex analogue of orthonormal), all three column vectors of T_{0+-}^{abc} are of unit length and they are mutually orthogonal.

B.5 Per-unit representation

In order to further simplify the equations to be handled, all system quantities are normalized according to the per unit system (pu). For a quantity x the following holds.

$$\bar{x} [\text{in } pu] = \frac{x [\text{in } SI]}{x_{base} [\text{in } SI]} \quad (\text{B.50})$$

Bases of different quantities can be set arbitrarily and independently, but this affect the form of equations in the pu domain. Two examples of voltage, current and power bases will be considered to illustrate the fact.

Case A

Two of s_{baseA} , v_{baseA} and i_{baseA} are picked arbitrarily and the last of the three is selected to

respect

$$s_{baseA} = 3 \cdot v_{baseA} \cdot i_{baseA}. \quad (B.51)$$

Case B

Two of s_{baseB} , v_{baseB} and i_{baseB} are picked arbitrarily and the last of the three is selected to respect

$$s_{baseB} = v_{baseB} \cdot i_{baseB}. \quad (B.52)$$

Case A could be for example when

- i_{baseA} is set to correspond to the rated RMS current of a line,
- v_{baseA} is set to correspond to the rated RMS line-to-ground voltage of a line, and
- s_{baseA} as calculated by (B.51) would correspond to rated three-phase power.

Case B could be for example when

- i_{baseB} is set to correspond to the rated RMS current of a line,
- v_{baseB} is set to correspond to the rated RMS line-to-ground voltage of a line, and
- s_{baseB} as calculated by (B.52) would correspond to rated single-phase power.

The equations for single- and three-phase power in a balanced system are respectively as follows.

$$s_{1\phi} = \mathbf{v}_\phi \cdot \mathbf{i}_\phi^* \quad (B.53)$$

$$s_{3\phi} = 3 \cdot \mathbf{v}_\phi \cdot \mathbf{i}_\phi^* \quad (B.54)$$

The above are in the SI system. In order to convert them to a pu system, both sides of the equations are divided by the respective s_{base} .

Case A

Appendix B. Common modeling assumptions in power systems

For the single-phase power

$$\begin{aligned}\frac{s_{1\Phi}}{s_{baseA}} &= \frac{\mathbf{v}_\phi \cdot \mathbf{i}_\phi^*}{s_{baseA}} \\ \bar{s}_{1\Phi} &= \frac{1}{3} \cdot \frac{\mathbf{v}_\Phi}{v_{baseA}} \cdot \frac{\mathbf{i}_\Phi^*}{i_{baseA}} \\ \bar{s}_{1\Phi} &= \frac{1}{3} \cdot \bar{\mathbf{v}}_\Phi \cdot \bar{\mathbf{i}}_\Phi^*\end{aligned}\tag{B.55}$$

For the three-phase power

$$\begin{aligned}\frac{s_{3\Phi}}{s_{baseA}} &= 3 \cdot \frac{\mathbf{v}_\phi \cdot \mathbf{i}_\phi^*}{s_{baseA}} \\ \bar{s}_{3\Phi} &= \frac{3}{3} \cdot \frac{\mathbf{v}_\Phi}{v_{baseA}} \cdot \frac{\mathbf{i}_\Phi^*}{i_{baseA}} \\ \bar{s}_{3\Phi} &= \bar{\mathbf{v}}_\Phi \cdot \bar{\mathbf{i}}_\Phi^*\end{aligned}\tag{B.56}$$

Case B

For the single-phase power

$$\begin{aligned}\frac{s_{1\Phi}}{s_{baseB}} &= \frac{\mathbf{v}_\phi \cdot \mathbf{i}_\phi^*}{s_{baseA}} \\ \bar{s}_{1\Phi} &= \frac{\mathbf{v}_\Phi}{v_{baseB}} \cdot \frac{\mathbf{i}_\Phi^*}{i_{baseB}} \\ \bar{s}_{1\Phi} &= \bar{\mathbf{v}}_\Phi \cdot \bar{\mathbf{i}}_\Phi^*\end{aligned}\tag{B.57}$$

For the three-phase power

$$\begin{aligned}\frac{s_{3\Phi}}{s_{baseB}} &= 3 \cdot \frac{\mathbf{v}_\phi \cdot \mathbf{i}_\phi^*}{s_{baseB}} \\ \bar{s}_{3\Phi} &= 3 \cdot \frac{\mathbf{v}_\Phi}{v_{baseB}} \cdot \frac{\mathbf{i}_\Phi^*}{i_{baseB}} \\ \bar{s}_{3\Phi} &= 3 \cdot \bar{\mathbf{v}}_\Phi \cdot \bar{\mathbf{i}}_\Phi^*\end{aligned}\tag{B.58}$$

From the above is it clearly shown that the selection of base quantities affects the form of the equations in the pu domain.

In this work a standard [321, 33] base system is adopted. A common assumption is adopted: first the power base is defined as a system wide 3-phase power base is defined. Then the voltage base is defined as rated line-to-line RMS voltage at the to-bus of the branch. All other relevant bases are derived from them [33].

$$s_{base} = \text{system-wide 3-phase power base, in MVA} \quad (\text{B.59a})$$

$$v_{base} = \text{rated line-to-line RMS bus voltage at the to-bus of the branch, in kV} \quad (\text{B.59b})$$

$$i_{base} = s_{base}/v_{base} \text{ corresponding to } \sqrt{3} \times \text{line RMS current, in A} \quad (\text{B.59c})$$

$$z_{3\Phi base} = v_{3\Phi base}/i_{3\Phi base}, \text{ in } \Omega \quad (\text{B.59d})$$

The voltage base is defined according to the π -equivalent model of a transmission line. It is taken to be the to-bus voltage of the branch, given that in the π -model the line impedance is considered after the transformer (see Fig. 3.6 in chapter 3).

The three-phase power equation can be alternatively read as follows.

$$|s_{3\Phi}| = 3 \cdot v_{\Phi} \cdot i_{\Phi} = (\sqrt{3} \cdot v_{\Phi}) \cdot (\sqrt{3} \cdot i_{\Phi}) = v_{LL} \cdot i_{\sqrt{3}\Phi} \quad (\text{B.60})$$

Where

- $v_{LL} = \sqrt{3} \cdot v_{\Phi}$ is the line-to-line voltage, and
- $i_{\sqrt{3}\Phi} = \sqrt{3} \cdot i_{\Phi}$ is the line current multiplied by $\sqrt{3}$.

To put it simply, a three-phase line carrying 1 pu voltage (line-to-line RMS) and 1 pu current ($\sqrt{3} \times$ line RMS), is transferring 1 pu of power.

These are quantities that are commonly considered in various power system studies, e.g. power flow. Multiplying both sides of (B.34) by $\sqrt{3}$ gives

$$\mathbf{i}_{+\sqrt{3}\Phi} = \mathbf{y} \cdot \mathbf{v}_{+LL}. \quad (\text{B.61})$$

Where $\mathbf{i}_{+\sqrt{3}\Phi}$ is the line RMS current of the direct sequence multiplied by a factor of $\sqrt{3}$, and \mathbf{v}_{+LL} is the line-to-line RMS voltage of the direct sequence. In the same vein, rewriting (B.45)

Appendix B. Common modeling assumptions in power systems

gives

$$s_{3\Phi} = 3 \cdot \mathbf{v}_+ \cdot \mathbf{i}_+^* = \left(\sqrt{3} \cdot \mathbf{v}_+ \right) \cdot \left(\sqrt{3} \cdot \mathbf{i}_\phi \right) \Leftrightarrow s_{3\Phi} = \mathbf{v}_{+LL} \cdot \mathbf{i}_{+\sqrt{3}\Phi}^* \quad (\text{B.62})$$

The pu transformation applied to (B.61) and (B.62) gives the basic core equations that are very commonly used in many power system analyses.

$$\bar{\mathbf{i}}_{+\sqrt{3}\Phi} = \bar{\mathbf{y}} \cdot \bar{\mathbf{v}}_{+LL} \quad (\text{B.63})$$

$$\bar{\mathbf{s}}_{3\Phi} = \bar{\mathbf{v}}_{+LL} \cdot \bar{\mathbf{i}}_{+\sqrt{3}\Phi}^* \quad (\text{B.64})$$

In order to make the pu transformation procedure clear an example for a real three-phase case is provided. Let the three-phase VA base be

$$s_{base} = 100 \text{ kVA}.$$

We consider the base line-to-line RMS voltage to be

$$v_{base} = 11 \text{ kV}.$$

The base value for the quantity $\sqrt{3} \times$ line RMS current is

$$i_{base} = 100/11 = 9.1 \text{ A}.$$

Suppose that in the system there is a three-port network fed with voltage and current that have the following full three-phase time-domain expressions.

$$V_{abc} = \begin{bmatrix} 10e3 \cdot \cos(\omega t + \phi_v) \\ 10e3 \cdot \cos(\omega t + \phi_v - 120^\circ) \\ 10e3 \cdot \cos(\omega t + \phi_v - 240^\circ) \end{bmatrix} \text{ V}$$

$$\mathbf{I}_{abc} = \begin{bmatrix} 5 \cdot \cos(\omega t + \phi_i) \\ 5 \cdot \cos(\omega t + \phi_i - 120^\circ) \\ 5 \cdot \cos(\omega t + \phi_i - 240^\circ) \end{bmatrix} A$$

The phasors of the above are as follows.

$$\mathbf{V}_{abc} = \begin{bmatrix} 7.07e3 \cdot e^{j \cdot \phi_v} \\ 7.07e3 \cdot e^{j \cdot (\phi_v - \frac{2\pi}{3})} \\ 7.07e3 \cdot e^{j \cdot (\phi_v - 2 \cdot \frac{2\pi}{3})} \end{bmatrix} V$$

$$\mathbf{I}_{abc} = \begin{bmatrix} 3.54e3 \cdot e^{j \cdot \phi_i} \\ 3.54e3 \cdot e^{j \cdot (\phi_i - \frac{2\pi}{3})} \\ 3.54e3 \cdot e^{j \cdot (\phi_i - 2 \cdot \frac{2\pi}{3})} \end{bmatrix} A$$

The three-phase power is calculated directly from the SI three-phase phasors as follows.

$$s_{3\Phi} = \mathbf{V}_{abc}^T \cdot \mathbf{I}_{abc}^* = \mathbf{v}_a \cdot \mathbf{i}_a^* + \mathbf{v}_b \cdot \mathbf{i}_b^* + \mathbf{v}_c \cdot \mathbf{i}_c^* = 75 \cdot e^{j \cdot (\phi_v - \phi_i)} \text{ kVA}$$

Alternatively, the per unit system can be used. First, the line-to-line RMS voltage is calculated as

$$v_{LL} = \sqrt{3} \cdot v_{1\Phi} = 12.25e3 \text{ V}.$$

Then, the quantity $\sqrt{3} \times$ line RMS current is calculated.

$$i_{L \times \sqrt{3}} = \sqrt{3} \cdot i_{1\Phi} = 6.13 \text{ A}$$

The above two, converted in pu are

$$\bar{v}_{LL} = \frac{v_{LL}}{v_{base}} = 1.11 \text{ pu},$$
$$\bar{i}_{L \times \sqrt{3}} = \frac{i_{L \times \sqrt{3}}}{i_{base}} = 0.67 \text{ pu}.$$

And the magnitude of the apparent power is

$$\bar{s} = \bar{v} \cdot \bar{i} = 0.75 \text{ pu}$$

If you convert the latter to SI, we see that it coincides with the value that has been directly calculated in SI.

$$s = \bar{s} \cdot s_{base} = 75 \text{ kVA}$$

C Numerical integration

Let an ordinary differential equation be

$$\dot{x} = f(x). \quad (\text{C.1})$$

Suppose an initial value of $x(0) = x_0$. Then the solution of the above along time can be expressed as follows.

$$x(t) = \phi(x_0) \quad (\text{C.2})$$

The point of numerical integration algorithms is to approximate this perfect analytical trajectory by a sequence of points x_k that correspond to respective time instants t_k . For a given point in time t_k a numerical integration algorithm produces a point of the trajectory for time instant $t_{k+1} = t_k + h$, where h is a timestep [33].

C.1 Explicit and implicit methods

Generally, numerical integration algorithms can be defined in two broad categories, *implicit* and *explicit* methods. Implicit methods in order to calculate the next step require knowledge of the variable in the next step itself.

$$\mathbf{x}_{\mathbf{k}+1} = g(\mathbf{x}_{\mathbf{k}+1}, x_k, x_{k-1}, \dots, x_0) \quad (\text{C.3})$$

Unknown quantities (in bold) appear in both sides of the equation, and thus it has to be solved

implicitly.

$$\tilde{g}(x) = x_{k+1} - g(x_{k+1}, x_k, x_{k-1}, \dots, x_0) = 0 \quad (\text{C.4})$$

On the contrary explicit method provide a direct, and hence explicit, solution for the next step, with unknowns appearing only on the left-hand side.

$$\mathbf{x}_{k+1} = g(x_k, x_{k-1}, \dots, x_0) \quad (\text{C.5})$$

Obviously, implicit methods are computationally much more demanding. This comes though with the benefit of having much better numerical properties, in terms of numerical stability. This is to say that they succeed in bounding the propagation of the error in stiff problems. Stiffness is the property exhibited by some differential equations, in which for the solution sequence x_k to be close to the real (perfect) solution $x(t)$, the time step h of numerical integration algorithms has to be very small. Otherwise numerical instability occurs, which means that the error is propagated in an unbounded fashion, rendering the solution useless.

Stiffness is often due to the fact that in the real system there are phenomena with very different time scales. For electromechanical transient simulation in power systems, this is not really the case as the dynamics are confined to the range of the electromechanical time constants of the generators of the network. Usually the latter are concentrated in the region of few hertz, and so transient simulators often benefit from the ease of implementation and the computational simplicity of explicit algorithms. A rule of thumb is that the integration time step should be one-fifth [291] up to one-half [297] the smallest effective time constant in the element models. Typical operator studies are performed with time steps of one-half [297] or one-fourth [322] of the period of the system nominal frequency - 50Hz in Europe, 60 Hz in N. America. In cases where simulators are designed to tackle both transient and longer term phenomena, the set of differential-algebraic equations are stiffer, and implicit algorithms are often favored [323].

C.2 Numerical properties of methods

A deeply specialized analysis of properties of numerical algorithms is out of the scope of this study. The interested reader is referred to [324, 325]. However, for clarity sake a brief view is given in this subsection.

C.2.1 Convergence

A numerical method is convergent if the approximated next step x_{k+1} approaches the exact solution $x(t_k + h)$ as the time step becomes infinitesimally small.

$$\lim_{h \rightarrow 0^+} x_{k+1} = x(t_k + h) \quad (\text{C.6})$$

C.2.2 Order

The order of a method is a measure of how well the difference equation defining the method approximates the differential equation. This can be quantified by the local truncation error.

$$\delta = x_{k+1} - x(t_k + h) \quad (\text{C.7})$$

Where x_{k+1} is given by the formula of the method. A method has order p if the local error is of order $O(h^{p+1})$ as the time step h goes to zero. Intuitively, the higher the order of the method, the smaller the propagation of the error across time steps.

C.2.3 Stability

Stability directly relates to the concept of stiffness. A common method to assess the stability of a method, is to apply it to the standard test equation.

$$\dot{y} = k \cdot y, \quad y_0 = y(0) = 1, \quad k \in \mathbb{C} \quad (\text{C.8})$$

The analytical solution of the above is $y(t) = e^{kt}$. This is a test of how well the method can follow dynamics with time constants dictated by k . When applied on this problem, the solution for the next step of an integration algorithm has the general form

$$y_{k+1} = \mathcal{F}(k \cdot h) \cdot y_k \quad (\text{C.9})$$

Appendix C. Numerical integration

Where $\mathcal{F}(\cdot)$ is determined by the algorithm. By induction the following holds.

$$y_k = \mathcal{F}(k \cdot h)^k \cdot y_0 \quad (\text{C.10})$$

So, for $\lim_{k \rightarrow \infty} y_k = 0$, it has to be $|\mathcal{F}(z)| < 1$, where $z := k \cdot h \in \mathbb{C}$. The latter defines the stability region for the method, in the complex z plane. A smaller stability region normally means that smaller time steps have to be adopted to prevent instability from occurring.

C.3 Truncation error

At each iteration a *local truncation error* is caused to the solution. These errors of each iteration accumulate to a final *global truncation error* for the method. Assume a continuous differential equation.

$$\dot{x} = f(x) \quad (\text{C.11})$$

We would like to approximate the exact solution $x(t)$ with a sequence of values $x^{(k)}$ at discrete time steps, $t = t_1, \dots, t_N = 1 \cdot h, \dots, N \cdot h$, where h is a constant time step. Suppose that we apply a method which results in the following numerical integration formula for each subsequent step.

$$x^{(k+1)} = x^{(k)} + h \cdot A(x^{(k)}, h, f) \quad (\text{C.12})$$

A is called the increment factor and is generally a function of previous approximations of $x(t)$ (only $x^{(k)}$ in the simplified example above), the time step h and the function f . Formally the local truncation error τ_{k+1} is the error that A introduces to $x^{(k+1)}$ if perfect knowledge for the previous value(s) is assumed $x^{(k)} = x(t_k)$.

$$\tau_{k+1} := x(t_{k+1}) - x^{(k+1)} = x(t_{k+1}) - x^{(k)} - h \cdot A(x^{(k)}, h, f) \quad (\text{C.13})$$

In a similar line, the global truncation error ϵ_k is calculated as the accumulation of all the local

truncation errors of all iterations if perfect knowledge for the initial value of x is assumed.

$$\epsilon_k := x(t_k) - x^{(k)} = x(t_k) - \left(x(0) + \sum_{i=0}^{k-1} A(x^{(i)}, h, f) \right) \quad (\text{C.14})$$

In the above a simplistic for $x^{(k+1)}$ has been assumed in (C.12). Expressions for the local and global truncation errors can be derived if more complex integration methods are assumed.

Bibliography

- [1] U.S.-Canada Power System Outage Task Force, “Final Report on the August 14, 2003 Blackout in the United States and Canada,” US Department of Energy and Canadian Ministry of Natural Resources, Technical Report, Apr. 2004. [Online]. Available: <http://www.ferc.gov/industries/electric/indus-act/reliability/blackout/ch1-3.pdf>
- [2] C. W. Taylor, *Power system voltage stability*. McGraw-Hill, 1994.
- [3] N. Watson and J. Arrillaga, *Power systems electromagnetic transients simulation*. IET, 2003, no. 39.
- [4] L. Fabre, G. Lanz, T. Kyriakidis, D. Sallin, I. Nagel, R. Cherkaoui, and M. Kayal, “An Ultra-High Speed Emulator Dedicated to Power System Dynamics Computation Based on a Mixed-Signal Hardware Platform,” *Power Systems, IEEE Transactions on*, vol. 28, no. 4, pp. 4228–4236, 2013.
- [5] D. Fabozzi, A. S. Chieh, B. Haut, and T. Van Cutsem, “Accelerated and Localized Newton Schemes for Faster Dynamic Simulation of Large Power Systems,” *Power Systems, IEEE Transactions on*, vol. 28, no. 4, pp. 4936–4947, 2013.
- [6] P. Aristidou, D. Fabozzi, and T. Van Cutsem, “Dynamic simulation of large-scale power systems using a parallel Schur-complement-based decomposition method,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 10, pp. 2561–2570, 2014.
- [7] A. A. Bazmi and G. Zahedi, “Sustainable energy systems: Role of optimization modeling techniques in power generation and supply—a review,” *Renewable and Sustainable Energy Reviews*, vol. 15, no. 8, pp. 3480–3500, 2011.
- [8] A. A. Bayod-Rújula, “Future development of the electricity systems with distributed generation,” *Energy*, vol. 34, no. 3, pp. 377–383, 2009.
- [9] S. C. Savulescu, *Real-time stability assessment in modern power system control centers*, ser. IEEE Press series on power engineering. Hoboken, N.J.: John Wiley & Sons, Inc., 2009.
- [10] T. Kyriakidis and M. Kayal, “Electronics as a means of facilitating the shift towards the smart grid,” *Network Industries Quarterly*, vol. 14, no. 2 & 3, pp. 12–16, 2012.

Bibliography

- [11] I. E. Agency, *World Energy Outlook 2010*. OECD Publishing, 2010.
- [12] G. J. Schaeffer and R. J. M. Belmans, "Smartgrids; a key step to energy efficient cities of the future," in *Power and Energy Society General Meeting, 2011 IEEE*, 2011, pp. 1–7.
- [13] R. Kumar, P. Datta Ray, and C. Reed, "Smart grid: An electricity market perspective," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, 2011, pp. 1–8.
- [14] A. Massoud and B. Wollenberg, "Toward a smart grid," *IEEE Power & Energy Magazine*, no. September/October 2005, pp. 34–41, 2005.
- [15] R. Metcalfe, "The enernet," 2008.
- [16] M. Hashmi, S. Hanninen, and K. Maki, "Survey of smart grid concepts, architectures, and technological demonstrations worldwide," in *Innovative Smart Grid Technologies (ISGT Latin America), 2011 IEEE PES Conference on*, 2011, pp. 1–7.
- [17] T. Ackermann, G. Andersson, and L. Söder, "Distributed generation: a definition," *Electric Power Systems Research*, vol. 57, no. 3, pp. 195–204, 2001.
- [18] K. El Bakari and W. L. Kling, "Virtual power plants: An answer to increasing distributed generation," in *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, 2010, pp. 1–6.
- [19] G. Pepermans, J. Driesen, D. Haeseldonckx, R. Belmans, and W. D'haeseleer, "Distributed generation: definition, benefits and issues," *Energy Policy*, vol. 33, no. 6, pp. 787–798, 2005.
- [20] J. Pecas Lopes, N. Hatziaargyriou, J. Mutale, P. Djapic, and N. Jenkins, "Integrating distributed generation into electric power systems: A review of drivers, challenges and opportunities," *Electric Power Systems Research*, vol. 77, no. 9, pp. 1189–1203, 2007.
- [21] J. C. Boemer, M. Gibescu, and W. L. Kling, "Dynamic models for transient stability analysis of transmission and distribution systems with distributed generation: An overview," in *PowerTech, 2009 IEEE Bucharest*, 2009, pp. 1–8.
- [22] C. Lo Prete, B. F. Hobbs, C. S. Norman, S. Cano-Andrade, A. Fuentes, M. R. von Spakovsky, and L. Mili, "Sustainability and reliability assessment of microgrids in a regional electricity market," *Energy*, 2011.
- [23] D. Pudjianto, C. Ramsay, and G. Strbac, "Virtual power plant and system integration of distributed energy resources," *Renewable Power Generation, IET*, vol. 1, no. 1, pp. 10–16, 2007.
- [24] R. H. Lasseter, "Microgrids," in *Power Engineering Society Winter Meeting, 2002. IEEE*, vol. 1, 2002, pp. 305–308 vol.1.

-
- [25] C. Marnay, F. J. Robio, and A. S. Siddiqui, "Shape of the microgrid," in *Power Engineering Society Winter Meeting, 2001. IEEE*, vol. 1, 2001, pp. 150–153 vol.1.
- [26] K. Dielmann and A. van der Velden, "Virtual power plants (vpp) - a new perspective for energy generation?" in *Modern Techniques and Technologies, 2003. MTT 2003. Proceedings of the 9th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists*, 2003, pp. 18–20.
- [27] C. Marnay and R. Firestone, "Microgrids: An emerging paradigm for meeting building electricity and heat requirements efficiently and with appropriate energy quality," in *European Council for an Energy Efficient Economy, 2007 Summer study*, Jun. 2007.
- [28] F. Katiraei, R. Iravani, N. Hatziargyriou, and A. Dimeas, "Microgrids management," *Power and Energy Magazine, IEEE*, vol. 6, no. 3, pp. 54–65, 2008.
- [29] B. Kroposki, R. Lasseter, T. Ise, S. Morozumi, S. Papatlianassiou, and N. Hatziargyriou, "Making microgrids work," *Power and Energy Magazine, IEEE*, vol. 6, no. 3, pp. 40–53, 2008.
- [30] M. Sebastian, J. Marti, and P. Lang, "Evolution of dso control centre tool in order to maximize the value of aggregated distributed generation in smart grid," in *SmartGrids for Distribution, 2008. IET-CIRED. CIRED Seminar*, 2008, pp. 1–4.
- [31] H. Happ, "Parallel processing in power systems," *International Journal of Electrical Power & Energy Systems*, vol. 4, no. 1, pp. 37–41, 1982.
- [32] W. F. Tinney and C. E. Hart, "Power Flow Solution by Newton's Method," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-86, no. 11, pp. 1449–1460, 1967.
- [33] K. R. Padiyar, *Power system dynamics: stability and control*. Kent, UK: Anshan Limited, 2004.
- [34] B. Palmer, W. Perkins, Y. Chen, S. Jin, D. Callahan, K. Glass, R. Diao, M. Rice, S. Elbert, M. Vallem *et al.*, "GridpackTM: a framework for developing power grid simulations on high performance computing platforms," in *Proceedings of the Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing*. IEEE Press, 2014, pp. 68–77.
- [35] A. Semlyen and F. de León, "Quasi-newton power flow using partial jacobian updates," *Power Systems, IEEE Transactions on*, vol. 16, no. 3, pp. 332–339, 2001.
- [36] F. Milano, *Power system modelling and scripting*, ser. Power Systems. London: Springer, 2010.
- [37] L. Powell, *Power system load flow analysis*. McGraw-Hill, 2005.

Bibliography

- [38] H. Dag and F. L. Alvarado, "Toward improved uses of the conjugate gradient method for power system applications," *Power Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 1306–1314, 1997.
- [39] S. Abhyankar, B. Smith, and E. Constantinescu, "Evaluation of overlapping restricted additive Schwarz preconditioning for parallel solution of very large power flow problems," in *Proceedings of the 3rd International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. ACM, 2013, p. 5.
- [40] PEGASE Consortium, "Algorithmic requirements for simulation of large network extreme scenarios," PEGASE Consortium, Technical Report DEL_WP4.1/2/3-D4.1, Apr. 2011, cRSA, RTE, TE, TU/e. [Online]. Available: <http://www.fp7-pegase.eu/download.html>
- [41] L. Wang, "Techniques for high performance analysis of transient stability," in *Power and Energy Society General Meeting, 2012 IEEE*, 2012, pp. 1–6.
- [42] Y. Saad, *Iterative methods for sparse linear systems*, 2nd ed. Philadelphia: SIAM, 2003.
- [43] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed., ser. Johns Hopkins studies in the mathematical sciences. Baltimore: Johns Hopkins University Press, 1996.
- [44] T. A. Davis, *Direct methods for sparse linear systems*, ser. Fundamentals of algorithms. Philadelphia: Society for Industrial and Applied Mathematics, 2006.
- [45] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, "Basic Linear Algebra Subprograms for Fortran Usage," *ACM Trans. Math. Softw.*, vol. 5, no. 3, pp. 308–323, 1979.
- [46] J. J. Dongarra, J. D. Croz, S. Hammarling, and R. J. Hanson, "An extended set of FORTRAN basic linear algebra subprograms," *ACM Trans. Math. Softw.*, vol. 14, no. 1, pp. 1–17, 1988.
- [47] J. J. Dongarra, J. D. Croz, S. Hammarling, and I. S. Duff, "A set of level 3 basic linear algebra subprograms," *ACM Trans. Math. Softw.*, vol. 16, no. 1, pp. 1–17, 1990.
- [48] Intel, "Intel Math Kernel Library (Intel MKL)," 2014, [Online; accessed 28-February-2015]. <https://software.intel.com/en-us/intel-mkl>. [Online]. Available: <https://software.intel.com/en-us/intel-mkl>
- [49] AMD, "Amd Core Math Library (ACML)," 2014, [Online; accessed 28-February-2015]. <http://developer.amd.com/tools-and-sdks/cpu-development/cpu-libraries/amd-core-math-library-acml/>. [Online]. Available: <http://developer.amd.com/tools-and-sdks/cpu-development/cpu-libraries/amd-core-math-library-acml/>
- [50] C. Mouton, "Linear Algebra Libraries," INRIA, Technical Report inria-00576469, Mar. 2009. [Online]. Available: <https://hal.inria.fr/inria-00576469>

-
- [51] F. Pruvost, T. Cadeau, P. Laurent-Gengoux, F. Magoules, F.-X. Bouchez, and B. Haut, "Numerical accelerations for power systems transient stability simulations," in *17th Power System Computation Conference*, 2011.
 - [52] S. K. Khaitan, J. D. McCalley, and C. Qiming, "Multifrontal Solver for Online Power System Time-Domain Simulation," *Power Systems, IEEE Transactions on*, vol. 23, no. 4, pp. 1727–1737, 2008.
 - [53] Y.-S. Zhang and H.-D. Chiang, "Fast Newton-FGMRES Solver for Large-Scale Power Flow Study," *Power Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 769–776, 2010.
 - [54] C. Parmer, E. Cotilla-Sanchez, H. K. Thornquist, and P. D. Hines, "Developing a dynamic model of cascading failure for high performance computing using trilinos," in *Proceedings of the first international workshop on High performance computing, networking and analytics for the power grid*. ACM, 2011, pp. 25–34.
 - [55] L. R. de Araujo, D. R. R. Penido, J. L. R. Pereira, and S. Carneiro Jr, "A Sparse Linear Systems Implementation for Electric Power Systems Solution Algorithms," *Journal of Control, Automation and Electrical Systems*, vol. 24, no. 4, pp. 504–512, 2013.
 - [56] S. Pandit, S. Soman, and S. Khaparde, "Design of generic direct sparse linear system solver in C++ for power system analysis," *Power Systems, IEEE Transactions on*, vol. 16, no. 4, pp. 647–652, 2001.
 - [57] C. F. Fu, J. D. McCalley, and J. Tong, "A Numerical Solver Design for Extended-Term Time-Domain Simulation," *Power Systems, IEEE Transactions on*, vol. 28, no. 4, pp. 4926–4935, 2013.
 - [58] A. Khajeh Zadeh, H. Zeynal, K. M. Nor, and M. Shaaban, "Performance Evaluation of SuperLU and PARDISO in Power System Load Flow Calculations," *Electrical Review (Przeglad Elektrotechniczny)*, pp. 290–294, 2011.
 - [59] M. Abdel-Akher, K. M. Nor, and A. A. Rashid, "Improved three-phase power-flow methods using sequence components," *Power Systems, IEEE Transactions on*, vol. 20, no. 3, pp. 1389–1397, 2005.
 - [60] J. Nieplocha, A. Marquez, V. Tipparaju, D. Chavarria-Miranda, R. Guttromson, and H. Huang, "Towards efficient power system state estimators on shared memory computers," in *Power Engineering Society General Meeting, 2006. IEEE*. IEEE, 2006, pp. 5–pp.
 - [61] C. Fu, "High-speed extended-term time-domain simulation for online cascading analysis of power system analysis of power system," PhD Thesis, Iowa State University, 2011.
 - [62] H. Mokhlis and K. Nor, "Object oriented sparse linear solver component for power system analysis," in *Power Engineering Conference, 2003. PECon 2003. Proceedings. National*. IEEE, 2003, pp. 22–26.

Bibliography

- [63] M. Tomim, J. Marti, and L. Wang, "Parallel solution of large power system networks using the Multi-Area Thévenin Equivalents (MATE) algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 9, pp. 497–503, 2009.
- [64] S. Kamel and F. Jurado, "Fast decoupled load flow analysis with SSSC power injection model," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 9, no. 4, pp. 370–374, 2014.
- [65] S. Smith, C. Woodward, L. Min, C. Jing, and A. Del Rosso, "On-line transient stability analysis using high performance computing," in *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*. IEEE, 2014, pp. 1–5.
- [66] C. Woodward, S. Smith, S. Khaitan, L. Min, and J. McCalley, "Advancing the Adoption of High Performance Computing for Time Domain Simulation," Lawrence Livermore National Laboratory (LLNL), Livermore, CA, Technical Report LLNL-TR-636060, 2013.
- [67] W. M. Siever, D. R. Tauritz, A. Miller, M. Crow, B. M. McMillin, and S. Atcitty, "Symbolic reduction for high-speed power system simulation," *Simulation*, vol. 84, no. 6, pp. 297–309, 2008.
- [68] S. Sommer and H. Johannsson, "Real-time thevenin impedance computation," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*. IEEE, 2013, pp. 1–6.
- [69] S. K. Khaitan and J. D. McCalley, "TDPSS: a scalable time domain power system simulator for dynamic security assessment," in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion*:. IEEE, 2012, pp. 323–332.
- [70] S. Khaitan and J. McCalley, "A Class of New Preconditioners for Linear Solvers Used in Power System Time-Domain Simulation," *Power Systems, IEEE Transactions on*, vol. 25, no. 4, pp. 1835–1844, Nov. 2010.
- [71] T. Orfanogianni and R. Bacher, "Using automatic code differentiation in power flow algorithms," *Power Systems, IEEE Transactions on*, vol. 14, no. 1, pp. 138–144, 1999.
- [72] Z. Qin and Y. Hou, "Application of non-linear programming for large-scale AC-DC power flow analysis," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–8.
- [73] H. Johannsson, "Development of early warning methods for electric power systems," Ph.D. dissertation, Technical University of Denmark (DTU), Lyngby, Denmark, Jun. 2011.
- [74] P. Bosshart and R. Bacher, "A domain architecture for solving simultaneous nonlinear network equations [power system simulation]," in *Power Industry Computer Applications, 1997. 20th International Conference on*. IEEE, 1997, pp. 142–148.
- [75] N. Li, "Improving the Execution Time of Large System Simulations," MSc Thesis, Arizona State University, Tempe, AZ, Nov. 2012.

- [76] F. Uriarte and K. Butler-Purry, "Multicore simulation of an AC-radial shipboard power system," in *Power and Energy Society General Meeting, 2010 IEEE*. IEEE, 2010, pp. 1–8.
- [77] Y. Cai, "Synchrophasor-Based Backup Differential Protection Design of Transmission Lines," BSc Thesis, University of Manitoba, Winnipeg, MB, 2014.
- [78] P. H. Divshali, S. Hosseinian, E. Nasr, and B. Vahidi, "Reliable prediction of Hopf bifurcation in power systems," *Electrical Engineering*, vol. 91, no. 2, pp. 61–68, 2009.
- [79] C. E. Portugal, R. B. Prada, and J. E. Pessanha, "Well-organized preconditioner for solving load flow problems by GMRES," in *Utility Exhibition on Power and Energy Systems: Issues & Prospects for Asia (ICUE), 2011 International Conference and*. IEEE, 2011, pp. 1–6.
- [80] J. E. Onoda Pessanha, C. E. Portugal, and O. R. Saavedra, "An Efficient Reordering Technique for Incomplete Factorization GMRES Preconditioning: Application in Load Flow Problems," *International Journal of Emerging Electric Power Systems*, vol. 12, no. 2, 2011.
- [81] R. Armistead and F. Li, "Parallel computing of sparse linear systems using matrix condensation algorithm," in *PowerTech, 2011 IEEE Trondheim*. IEEE, 2011.
- [82] C. G. Petra, O. Schenk, and M. Anitescu, "Real-Time Stochastic Optimization of Complex Energy Systems on High-Performance Computers," *Computing in Science & Engineering*, vol. 16, no. 5, pp. 32–42, 2014.
- [83] J.-C. Tournier, V. Donde, and Z. Li, "Comparison of Direct and Iterative Sparse Linear Solvers for Power System Applications on Parallel Computing Platforms," ABB Inc. Corporate Research, Tech. Rep., 2011.
- [84] S. Abhyankar, A. Flueck, X. Zhang, and H. Zhang, "Development of a parallel three-phase transient stability simulator for power systems," in *Proceedings of the first international workshop on High performance computing, networking and analytics for the power grid*. ACM, 2011, pp. 19–24.
- [85] S. Abhyankar and A. J. Flueck, "Real-time power system dynamics simulation using a parallel block-jacobi preconditioned Newton-GMRES scheme," in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*. IEEE, 2012, pp. 299–305.
- [86] S. Abhyankar, B. Smith, H. Zhang, and A. Flueck, "Using PETSc to develop scalable applications for next-generation power grid," in *Proceedings of the first international workshop on High performance computing, networking and analytics for the power grid*. ACM, 2011, pp. 67–74.
- [87] S. Abhyankar and A. J. Flueck, "Parallel-in-space-and-time scheme for implicitly coupled electromechanical and electromagnetic transients simulation," in *Proceedings of International Conference on Power System Transients (IPST)*, 2013.

Bibliography

- [88] F. Milano, "Dome," 2011, [Online; accessed 28-February-2015]. <http://faraday1.ucd.ie/dome.html>. [Online]. Available: <http://faraday1.ucd.ie/dome.html>
- [89] H. Wang, C. E. Murillo-Sánchez, R. D. Zimmerman, and R. J. Thomas, "On computational issues of market-based optimal power flow," *Power Systems, IEEE Transactions on*, vol. 22, no. 3, pp. 1185–1193, 2007.
- [90] T. Cui and F. Franchetti, "Power system probabilistic and security analysis on commodity high performance computing systems," in *Proceedings of the 3rd International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. ACM, 2013, p. 2.
- [91] T. Cui, R. Yang, G. Hug, and F. Franchetti, "Accelerated AC contingency calculation on commodity multi-core SIMD CPUs," in *PES General Meeting| Conference & Exposition, 2014 IEEE*. IEEE, 2014, pp. 1–5.
- [92] Y. Chen, S. Wang, M. Rice, and Z. Huang, "Sub-second parallel state estimation," Pacific Northwest National Laboratory (PNNL), Richland, WA (US), Tech. Rep., 2014.
- [93] Science & Technology Facilities Council, "A collection of Fortran codes for large scale scientific computation," 2015, [Online; accessed 28-February-2015]. <http://www.hsl.rl.ac.uk/>. [Online]. Available: <http://www.hsl.rl.ac.uk/>
- [94] F. Milano, "A Python-based software tool for power system analysis," in *Power and Energy Society General Meeting (PES), 2013 IEEE*. IEEE, 2013, pp. 1–5.
- [95] S. K. Khaitan and A. Gupta, *High Performance Computing in Power and Energy Systems*. Springer, 2013.
- [96] A. B. Alves, E. N. Asada, and A. Monticelli, "Critical evaluation of direct and iterative methods for solving $Ax=b$ systems in power flow calculations and contingency analysis," *Power Systems, IEEE Transactions on*, vol. 14, no. 2, pp. 702–708, 1999.
- [97] Y. Chen and C. Shen, "A Jacobian-free Newton-GMRES(m) method with adaptive preconditioner and its application for power flow calculations," *Power Systems, IEEE Transactions on*, vol. 21, no. 3, pp. 1096–1103, 2006.
- [98] H. Dag and A. Semlyen, "A new preconditioned conjugate gradient power flow," *Power Systems, IEEE Transactions on*, vol. 18, no. 4, pp. 1248–1255, 2003.
- [99] F. de Leon and A. Sernlyen, "Iterative solvers in the Newton power flow problem: preconditioners, inexact solutions and partial Jacobian updates," *Generation, Transmission and Distribution, IEE Proceedings*, vol. 149, no. 4, pp. 479–484, 2002.
- [100] A. J. Flueck and C. Hsiao-Dong, "Solving the nonlinear power flow equations with an inexact Newton method using GMRES," *Power Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 267–273, 1998.

-
- [101] F. D. Galiana, H. Javidi, and S. McFee, "On the application of a pre-conditioned conjugate gradient algorithm to power network analysis," *Power Systems, IEEE Transactions on*, vol. 9, no. 2, pp. 629–636, 1994.
 - [102] R. Idema, D. J. P. Lahaye, C. Vuik, and L. van der Sluis, "Scalable Newton-Krylov Solver for Very Large Power Flow Problems," *Power Systems, IEEE Transactions on*, vol. 27, no. 1, pp. 390–396, 2012.
 - [103] R. Idema, G. Papaefthymiou, D. Lahaye, C. Vuik, and L. van der Sluis, "Towards Faster Solution of Large Power Flow Problems," *Power Systems, IEEE Transactions on*, vol. 28, no. 4, pp. 4918–4925, 2013.
 - [104] M. A. Pai and H. Dag, "Iterative solver techniques in large scale power system computation," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 4, 1997, pp. 3861–3866 vol.4.
 - [105] A. Semlyen, "Fundamental concepts of a Krylov subspace power flow methodology," *Power Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 1528–1537, 1996.
 - [106] D. J. Tylavsky, A. Bose, F. L. Alvarado, R. Betancour, K. Clements, G. T. Heydt, G. Huang, M. Ilic, M. La Scala, M. A. Pai, C. Pottle, S. Talukdar, J. E. Van Ness, and F. Wu, "Parallel processing in power systems computation," *Power Systems, IEEE Transactions on*, vol. 7, no. 2, pp. 629–638, 1992.
 - [107] P. Luszczek, J. Kurzak, and J. Dongarra, "Looking back at dense linear algebra software," *Journal of Parallel and Distributed Computing*, vol. 74, no. 7, pp. 2548–2560, 2014.
 - [108] D. B. Kirk and W. H. Wen-mei, *Programming massively parallel processors: a hands-on approach*. Newnes, 2012.
 - [109] G. Granelli, M. Montagna, M. La Scala, and F. Torelli, "Relaxation-Newton methods for transient stability analysis on a vector/parallel computer," in *Power Industry Computer Application Conference, 1993. Conference Proceedings*. IEEE, 1993, pp. 387–393.
 - [110] S. H. Fuller and L. I. Millett, "Computing performance: Game over or next level?" *Computer*, vol. 44, no. 1, pp. 31–38, 2011.
 - [111] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and others, "The landscape of parallel computing research: A view from berkeley," Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, Tech. Rep., 2006.
 - [112] A. R. Brodtkorb, C. Dyken, T. R. Hagen, J. M. Hjelmervik, and O. O. Storaasli, "State-of-the-art in heterogeneous computing," *Scientific Programming*, vol. 18, no. 1, pp. 1–33, 2010.

Bibliography

- [113] S. Skalicky, S. Lopez, M. \Lukowiak, J. Letendre, and D. Gasser, "Linear algebra computations in heterogeneous systems," in *Application-Specific Systems, Architectures and Processors (ASAP)*, 2013 IEEE 24th International Conference on. IEEE, 2013, pp. 273–276.
- [114] R. Weber, A. Gothandaraman, R. J. Hinde, and G. D. Peterson, "Comparing hardware accelerators in scientific applications: A case study," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 1, pp. 58–68, 2011.
- [115] S. Kestur, J. D. Davis, and O. Williams, "BLAS Comparison on FPGA, CPU and GPU," in *VLSI (ISVLSI)*, 2010 IEEE Computer Society Annual Symposium on, 2010, pp. 288–293.
- [116] K. D. Underwood and K. S. Hemmert, "Closing the gap: CPU and FPGA trends in sustainable floating-point BLAS performance," in *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on*, 2004, pp. 219–228.
- [117] I. Dzafic and H.-T. Neisius, "Real-time power flow algorithm for shared memory multi-processors for European distribution network types," in *IPEC, 2010 Conference Proceedings*. IEEE, 2010, pp. 152–158.
- [118] J. P. Hulskamp, S. M. Chan, and J. F. Fazio, "Power flow outage studies using an array processor," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-101, no. 1, pp. 254–261, 1982.
- [119] M. Takatoo, S. Abe, T. Bando, K. Hirasawa, M. Goto, T. Kato, and T. Kanke, "Floating vector processor for power system simulation," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-104, no. 12, pp. 3360–3366, 1985.
- [120] G. Granelli, M. Montagna, G. Pasini, and P. Marannino, "Vector computer implementation of power flow outage studies," *Power Systems, IEEE Transactions on*, vol. 7, no. 2, pp. 798–804, 1992.
- [121] C. Aykanat and N. Guven, "Algorithms for efficient vectorization of repeated sparse power system network computations," *Power Systems, IEEE Transactions on*, vol. 10, no. 1, pp. 448–456, 1995.
- [122] Y.-F. Fung, M. F. Ercan, T. Ho, and W. Cheung, "A parallel computation of power system equations," in *Euro-Par 2001 Parallel Processing*. Springer, 2001, pp. 371–374.
- [123] Y. Fung, T. Ho, W. Cheung, and M. Ercan, "Parallel solution for railway power network simulation," in *Communications, Computers and signal Processing, 2001. PACRIM. 2001 IEEE Pacific Rim Conference on*, vol. 2. IEEE, 2001, pp. 413–416.
- [124] Y. Fung, C. Chung, W. Cheung, M. Ercan, M. Singh, and T. K. Ho, "A cost-effective parallel solution for real-time dynamic security assessment," in *ASDCOM 2003-The 6th International Conference on Advances in Power System Control, Operation and Management*, vol. 1. IEEE Computer Society, 2004, pp. 256–261.

- [125] M. F. Ercan, Y.-f. Fung, T.-k. Ho, and W.-l. Cheung, "Parallel linear system solution and its application to railway power network simulation," in *Euro-Par 2003 Parallel Processing*, Springer, 2003, pp. 537–540.
- [126] J. C. Pichel and F. F. Rivera, "Sparse matrix–vector multiplication on the Single-Chip Cloud Computer many-core processor," *Journal of Parallel and Distributed Computing*, vol. 73, no. 12, pp. 1539–1550, 2013.
- [127] A.-J. Yzelman and D. Roose, "High-level strategies for parallel shared-memory sparse matrix-vector multiplication," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 116–125, 2014.
- [128] J. Dongarra, M. Faverge, T. Herault, M. Jacquelin, J. Langou, and Y. Robert, "Hierarchical QR factorization algorithms for multi-core clusters," *Parallel Computing*, vol. 39, no. 4, pp. 212–232, 2013.
- [129] D. Maurer and C. Wieners, "A parallel block LU decomposition method for distributed finite element matrices," *Parallel Computing*, vol. 37, no. 12, pp. 742–758, 2011.
- [130] M. Förster and J. Kraus, "Scalable parallel AMG on ccNUMA machines with OpenMP," *Computer Science-Research and Development*, vol. 26, no. 3-4, pp. 221–228, 2011.
- [131] A. Heinecke, M. Klemm, and H.-J. Bungartz, "From gpgpu to many-core: Nvidia fermi and intel many integrated core architecture," *Computing in Science & Engineering*, vol. 14, no. 2, pp. 78–83, 2012.
- [132] K. Lau, D. J. Tylavsky, and A. Bose, "Coarse grain scheduling in parallel triangular factorization and solution of power system matrices," *Power Systems, IEEE Transactions on*, vol. 6, no. 2, pp. 708–714, 1991.
- [133] D. P. Koester, S. Ranka, and G. C. Fox, "A parallel Gauss-Seidel algorithm for sparse power system matrices," in *Proceedings of the 1994 ACM/IEEE conference on Supercomputing*. IEEE Computer Society Press, 1994, pp. 184–193.
- [134] J.-a. Zhang, N. Zhang, and Y.-l. Jiang, "Parallelizing Methods Analysis for Solving Large Sparse Power Network Equations on Multi-core Processor Platforms," in *Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on*, vol. 1. IEEE, 2011, pp. 192–195.
- [135] Y. Wallach and V. Conrad, "Parallel solutions of load flow problems," *Archiv fur Elektrotechnik*, vol. 57, no. 6, pp. 345–354, 1976.
- [136] H. Taoka, S. Abe, and S. Takeda, "Multiprocessor system for power system analysis," *Annual Review in Automatic Programming*, vol. 11, pp. 101–106, 1981.
- [137] T. Oyama, T. Kitahara, and Y. Serizawa, "Parallel processing for power system analysis using band matrix," in *Power Industry Computer Application Conference, 1989. PICA'89, Conference Papers*. IEEE, 1989, pp. 119–125.

Bibliography

- [138] F. Wang, N. Hadjsaid, and J. Sabonnadiere, "Power system parallel computation by a transputer network," *Electric Power Systems Research*, vol. 52, no. 1, pp. 1–7, 1999.
- [139] H. Dag and G. Soykan, "Power flow using thread programming," in *IEEE PowerTech Trondheim 2011*. IEEE, 2011, pp. 1–5.
- [140] Z. Li, V. Donde, J.-C. Tournier, and F. Yang, "On limitations of traditional multi-core and potential of many-core processing architectures for sparse linear solvers used in large-scale power system applications," in *Power and Energy Society General Meeting, 2011 IEEE*. IEEE, 2011, pp. 1–8.
- [141] Syafii, K. Nor, M. Abdel-Akher, and others, "Parallel three-phase load flow analysis for large scale unbalanced distribution networks," in *Power Engineering, Energy and Electrical Drives (POWERENG), 2011 International Conference on*. IEEE, 2011, pp. 1–6.
- [142] T. Cui and F. Franchetti, "A multi-core high performance computing framework for probabilistic solutions of distribution systems," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–6.
- [143] C. Lemaitre and B. Thomas, "Two applications of parallel processing in power system computation," in *Power Industry Computer Application Conference, 1995. Conference Proceedings., 1995 IEEE*. IEEE, 1995, pp. 62–69.
- [144] J. Fong and C. Pottle, "Parallel processing of power system analysis problems via simple parallel microcomputer structures," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-97, no. 5, pp. 1834–1841, 1978.
- [145] A. Kulkarni, M. Pai, and S. Ghoshal, "Parallel computation of power system dynamics using multistep methods," *International Journal of Electrical Power & Energy Systems*, vol. 14, no. 1, pp. 33–38, 1992.
- [146] G. Vuong, R. Chahine, and S. Behling, "Supercomputing for power system analysis," *Computer Applications in Power, IEEE*, vol. 5, no. 3, pp. 45–49, 1992.
- [147] M. Ten Bruggencate and S. Chalasani, "Parallel implementations of the power system transient stability problem on clusters of workstations," in *Proceedings of the 1995 ACM/IEEE conference on Supercomputing*. ACM, 1995, p. 34.
- [148] Y. Oda and T. Oyama, "Fast calculation using parallel processing and pipeline processing in power system analysis," *Electrical engineering in Japan*, vol. 116, no. 5, pp. 85–96, 1996.
- [149] H. Sasaki, J. Kubokawa, N. Yorino, J. Takehara, and M. Kitagawa, "A parallel computation of state estimation by transputer [power systems]," in *Power System Monitoring and Control, 1991., Third International Conference on*. IET, 1991, pp. 261–263.
- [150] T. D. Han and T. S. Abdelrahman, "hiCUDA: High-level GPGPU programming," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 1, pp. 78–90, 2011.

-
- [151] A. R. Brodtkorb, T. R. Hagen, and M. L. Sætra, "Graphics processing unit (GPU) programming strategies and trends in GPU computing," *Journal of Parallel and Distributed Computing*, vol. 73, no. 1, pp. 4–13, 2013.
- [152] X. Chen, L. Ren, Y. Wang, and H. Yang, "GPU-Accelerated Sparse LU Factorization for Circuit Simulation with Performance Modeling," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 786–795, Mar. 2015.
- [153] M. M. Dehnavi, D. M. Fernandez, J. Gaudiot, and D. D. Giannacopoulos, "Parallel sparse approximate inverse preconditioning on graphic processing units," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 9, pp. 1852–1862, 2013.
- [154] J. Kurzak, P. Luszczek, M. Faverge, and J. Dongarra, "LU factorization with partial pivoting for a multicore system with accelerators," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 8, pp. 1613–1621, 2013.
- [155] R. Guerrieri, T. De Marco, and F. Ries, "Triangular matrix inversion on heterogeneous multicore systems," *IEEE Transactions on Parallel & Distributed Systems*, vol. 23, no. 1, pp. 177–184, 2012.
- [156] V. W. Lee, C. Kim, J. Chhugani, M. Deisher, D. Kim, A. D. Nguyen, N. Satish, M. Smelyanskiy, S. Chennupaty, P. Hammarlund, R. Singhal, and P. Dubey, "Debunking the 100x gpu vs. cpu myth: An evaluation of throughput computing on cpu and gpu," *SIGARCH Comput. Archit. News*, vol. 38, no. 3, pp. 451–460, Jun. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1816038.1816021>
- [157] J. Filipovič, M. Madzin, J. Fousek, and L. Matyska, "Optimizing CUDA Code By Kernel Fusion—Application on BLAS," *arXiv preprint arXiv:1305.1183*, 2013.
- [158] M. Showerman, J. Enos, C. Steffen, S. Treichler, W. Gropp, and W.-m. W. Hwu, "EcoG: A Power-Efficient GPU Cluster Architecture for Scientific Computing," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 83–87, 2011.
- [159] W.-c. Feng, K. Cameron, T. Scogland, and B. Subramaniam, "The Green 500," 2015, [Online; accessed 28-February-2015]. <http://www.green500.org/>. [Online]. Available: <http://www.green500.org/>
- [160] A. Gopal, "DC power flow based contingency analysis using graphics processing units," Ph.D. dissertation, Drexel University, Mar. 2010.
- [161] N. Garcia, "Parallel power flow solutions using a biconjugate gradient algorithm and a Newton method: A GPU-based approach," in *Power and Energy Society General Meeting, 2010 IEEE*. IEEE, 2010, pp. 1–4.
- [162] J. Singh and I. Aruni, "Accelerating power flow studies on graphics processing unit," in *India Conference (INDICON), 2010 Annual IEEE*. IEEE, 2010, pp. 1–5.

- [163] C. Vilacha, J. Moreira, E. Miguez, and A. F. Otero, "Massive Jacobi power flow based on SIMD-processor," in *Environment and Electrical Engineering (EEEIC), 2011 10th International Conference on*. IEEE, 2011, pp. 1–4.
- [164] D. Ablakovic, I. Dzafic, and S. Kecici, "Parallelization of radial three-phase distribution power flow using GPU," in *Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on*. IEEE, 2012, pp. 1–7.
- [165] C. Guo, B. Jiang, H. Yuan, Z. Yang, L. Wang, and S. Ren, "Performance comparisons of parallel power flow solvers on gpu system," in *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012 IEEE 18th International Conference on*. IEEE, 2012, pp. 232–239.
- [166] M. Yang, C. Sun, Z. Li, and D. Cao, "An improved sparse matrix-vector multiplication kernel for solving modified equation in large scale power flow calculation on CUDA," in *Power Electronics and Motion Control Conference (IPEMC), 2012 7th International*, vol. 3. IEEE, 2012, pp. 2028–2031.
- [167] X. Li, F. Li, and J. M. Clark, "Exploration of multifrontal method with GPU in power flow computation," in *Power and Energy Society General Meeting (PES), 2013 IEEE*. IEEE, 2013, pp. 1–5.
- [168] X. Li and F. Li, "GPU-based power flow analysis with Chebyshev preconditioner and conjugate gradient method," *Electric Power Systems Research*, vol. 116, pp. 87–93, 2014.
- [169] J.-C. Tournier, V. Donde, and Z. Li, "Potential of general purpose graphic processing unit for energy management system," in *Parallel Computing in Electrical Engineering (PARELEC), 2011 6th International Symposium on*. IEEE, 2011, pp. 50–55.
- [170] H. Karimipour and V. Dinavahi, "Accelerated parallel WLS state estimation for large-scale power systems on GPU," in *North American Power Symposium (NAPS), 2013*. IEEE, 2013, pp. 1–6.
- [171] V. Jalili-Marandi and V. Dinavahi, "SIMD-Based Large-Scale Transient Stability Simulation on the Graphics Processing Unit," *Power Systems, IEEE Transactions on*, vol. 25, no. 3, pp. 1589–1599, 2010.
- [172] V. Jalili-Marandi, Z. Zhiyin, and V. Dinavahi, "Large-Scale Transient Stability Simulation of Electrical Power Systems on Parallel GPUs," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 7, pp. 1255–1266, 2012.
- [173] Z. Qin and Y. Hou, "A GPU-based Transient Stability Simulation using Runge-Kutta Integration Algorithm," *International Journal of Smart Grid and Clean Energy*, vol. 2, no. 1, pp. 32–39, 2012.
- [174] B. Wen, W. Guo, J. Hu, F. Wang, and J. Ye, "GPU based parallel simulation of transient stability using symplectic Gauss algorithm and preconditioned GMRES method," in *Power Engineering and Automation Conference (PEAM), 2012 IEEE*. IEEE, 2012, pp. 1–5.

-
- [175] J. K. Debnath, W.-K. Fung, A. M. Gole, and S. Filizadeh, "Simulation of large-scale electrical power networks on graphics processing units," in *Electrical Power and Energy Conference (EPEC), 2011 IEEE*. IEEE, 2011, pp. 199–204.
- [176] —, "Electromagnetic transient simulation of large-scale electrical power networks using graphics processing units," in *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*. IEEE, 2012, pp. 1–4.
- [177] S. Cieslik, "GPU Implementation of the Electric Power System Model for Real-Time Simulation of Electromagnetic Transients," in *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering*. Atlantis Press, 2013.
- [178] N. Garcia and R. Olmos, "GPU-accelerated Poincarè map method for harmonic-oriented analyses of power systems," in *Power and Energy Society General Meeting (PES), 2013 IEEE*, Jul. 2013, pp. 1–5.
- [179] Z. Zhou and V. Dinavahi, "Parallel Massive-Thread Electromagnetic Transient Simulation on GPU," *Power Delivery, IEEE Transactions on*, vol. 29, no. 3, pp. 1045–1053, Jun. 2014.
- [180] L. Rakai and W. Rosehart, "GPU-Accelerated Solutions to Optimal Power Flow Problems," in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. IEEE, 2014, pp. 2511–2516.
- [181] A. George, H. Lam, and G. Stitt, "Novo-g: at the forefront of scalable reconfigurable supercomputing," *Computing in Science and Engineering*, vol. 13, no. 1, pp. 82–86, 2011.
- [182] G. R. Morris and K. H. Abed, "Mapping a Jacobi Iterative Solver onto a High-Performance Heterogeneous Computer," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 1, pp. 85–91, 2013.
- [183] T. Nechma and M. Zwolinski, "Parallel sparse matrix solution for circuit simulation on fpgas," *Computers, IEEE Transactions on*, vol. 64, no. 4, pp. 1090–1103, April 2015.
- [184] J. D. Cappello and D. Strenski, "A practical measure of fpga floating point acceleration for high performance computing," in *Application-Specific Systems, Architectures and Processors (ASAP), 2013 IEEE 24th International Conference on*. IEEE, 2013, pp. 160–167.
- [185] R. Martinez-Alonso, K. Mino, and D. Torres-Lucio, "Array Processors Designed with VHDL for Solution of Linear Equation Systems Implemented in a FPGA," in *Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2010*, 2010, pp. 731–736.
- [186] L. Zhuo and V. K. Prasanna, "Scalable and Modular Algorithms for Floating-Point Matrix Multiplication on Reconfigurable Computing Systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, no. 4, pp. 433–448, 2007.
- [187] D. G. Chavarria-Miranda and D. Chassin, "A Hardware-Accelerated Steady-State Power Flow Solver," Pacific Northwest National Laboratory (PNNL), Richland, WA, Tech. Rep., Oct. 2005.

- [188] J. Foertsch, J. Johnson, and P. Nagvajara, "Jacobi load flow accelerator using FPGA," in *Power Symposium, 2005. Proceedings of the 37th Annual North American*, 2005, pp. 448–454.
- [189] J. Johnson, P. Vachranukunkiet, S. Tiwari, P. Nagvajara, and C. Nwankpa, "Performance Analysis of Load Flow Computation using FPGA," in *15th Power Systems Computation Conference (PSCC)*, Liege, Belgium, Aug. 2005.
- [190] X. Xu, S. G. Ziavras, and T.-G. Chang, "An FPGA-Based parallel accelerator for matrix multiplications in the newton-raphson method," in *Embedded and Ubiquitous Computing-EUC 2005*. Springer, 2005, pp. 458–468.
- [191] M. Murach, P. Vachranukunkiet, P. Nagvajara, J. Johnson, and C. Nwankpa, "Optimal reconfigurable hw/sw co-design of load flow and optimal power flow computation," in *Power Engineering Society General Meeting, 2006. IEEE*. IEEE, 2006, pp. 5–pp.
- [192] X. Wang, S. G. Ziavras, C. Nwankpa, J. Johnson, and P. Nagvajara, "Parallel solution of Newton's power flow equations on configurable chips," *International Journal of Electrical Power & Energy Systems*, vol. 29, no. 5, pp. 422–431, 2007.
- [193] J.-H. Byun, A. Ravindran, A. Mukherjee, B. Joshi, and D. Chassin, "Accelerating the Gauss-Seidel power flow solver on a high performance reconfigurable computer," in *Field Programmable Custom Computing Machines, 2009. FCCM'09. 17th IEEE Symposium on*. IEEE, 2009, pp. 227–230.
- [194] K. Cunningham, P. Nagvajara, and J. Johnson, "Reconfigurable multicore architecture for power flow calculation," in *North American Power Symposium (NAPS), 2011*. IEEE, 2011, pp. 1–7.
- [195] M. Matar, M. Abdel-Rahman, and A. Soliman, "Developing an FPGA coprocessor for real-time simulation of power systems," in *Electrical, Electronic and Computer Engineering, 2004. ICEEC '04. 2004 International Conference on*, Sep. 2004, pp. 791–794.
- [196] J. C. Pimentel and H. Le-Huy, "Hardware emulation for real-time power system simulation," in *Industrial Electronics, 2006 IEEE International Symposium on*, vol. 2. IEEE, 2006, pp. 1560–1565.
- [197] J. C. Pimentel, "Implementation of simulation algorithms in FPGA for real time simulation of electrical networks with power electronics devices," in *Reconfigurable Computing and FPGA's, 2006. ReConFig 2006. IEEE International Conference on*. IEEE, 2006, pp. 1–8.
- [198] Y. Chen and V. Dinavahi, "FPGA-based real-time EMTP," *Power Delivery, IEEE Transactions on*, vol. 24, no. 2, pp. 892–902, 2009.
- [199] —, "An iterative real-time nonlinear electromagnetic transient solver on FPGA," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 6, pp. 2547–2555, 2011.

-
- [200] —, “Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation,” *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 2, pp. 1300–1309, 2012.
- [201] —, “Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems,” *IET Generation, Transmission & Distribution*, vol. 7, no. 5, pp. 451–463, 2013.
- [202] M. Matar and R. Iravani, “The Reconfigurable-Hardware Real-Time and Faster-Than-Real-Time Simulator for the Analysis of Electromagnetic Transients in Power Systems,” *Power Delivery, IEEE Transactions on*, vol. 28, no. 2, pp. 619–627, 2013.
- [203] M. Murach, P. Nagvajara, J. Johnson, and C. Nwankpa, “Optimal power flow utilizing FPGA technology,” in *Power Symposium, 2005. Proceedings of the 37th Annual North American*, 2005, pp. 97–101.
- [204] M. Ali, E. Stotzer, F. D. Igual, and R. A. van de Geijn, “Level-3 BLAS on the TI C6678 multi-core DSP,” in *Computer Architecture and High Performance Computing (SBAC-PAD), 2012 IEEE 24th International Symposium on*. IEEE, 2012, pp. 179–186.
- [205] Y. Gao and J. D. Bakos, “Sparse Matrix-Vector Multiply on the Texas Instruments C6678 Digital Signal Processor,” in *Application-Specific Systems, Architectures and Processors (ASAP), 2013 IEEE 24th International Conference on*. IEEE, 2013, pp. 168–174.
- [206] F. D. Igual, M. Ali, A. Friedmann, E. Stotzer, T. Wentz, and R. Van de Geijn, “Unleashing DSPs for General-Purpose HPC,” The University of Texas at Austin, Department of Computer Science, Technical Report TR-12-02, FLAME Working Note #61, Feb. 2012.
- [207] E. Stotzer, A. Jayaraj, M. Ali, A. Friedmann, G. Mitra, A. P. Rendell, and I. Lintault, “OpenMP on the low-power TI Keystone II ARM/DSP System-on-Chip,” in *OpenMP in the Era of Low Power Devices and Accelerators*, ser. Lecture Notes in Computer Science, A. P. Rendell, B. M. Chapman, and M. S. Müller, Eds. Springer Berlin Heidelberg, 2013, vol. 8122, pp. 114–127.
- [208] K. Li, Y. Pan, and S. Q. Zheng, “Fast and processor efficient parallel matrix multiplication algorithms on a linear array with a reconfigurable pipelined bus system,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 9, no. 8, pp. 705–720, 1998.
- [209] D. I. Moldovan, “On the design of algorithms for vlsi systolic arrays,” *Proceedings of the IEEE*, vol. 71, no. 1, pp. 113–120, 1983.
- [210] A. Pedram, R. A. van de Geijn, and A. Gerstlauer, “Codesign tradeoffs for high-performance, low-power linear algebra architectures,” *Computers, IEEE Transactions on*, vol. 61, no. 12, pp. 1724–1736, 2012.
- [211] A. Pedram, A. Gerstlauer, and R. Geijn, “A high-performance, low-power linear algebra core,” in *Application-Specific Systems, Architectures and Processors (ASAP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 35–42.

Bibliography

- [212] C. González-Álvarez, J. B. Sartor, C. Álvarez, D. Jiménez-González, and L. Eeckhout, “Accelerating an application domain with specialized functional units,” *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 10, no. 4, p. 47, 2013.
- [213] M. Gries and K. Keutzer, *Building ASIPs: The Mescal Methodology: The MESCAL Methodology*. New York, NY: Springer Science & Business Media, 2006.
- [214] Y. Leung, *VLSI Systolic Array for Matrix Triangulation in Load Flow Analysis*, ser. MSU-ENGR. Michigan State University. Department of Electrical Engineering and Systems Science, 1983.
- [215] M. Shanblatt and Y.-Y. Leung, “The promise of VLSI for load flow computation enhancement,” *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-103, no. 7, pp. 1714–1719, 1984.
- [216] T. Cornu, P. Ienne, D. Niebur, and M. A. Viredaz, “A systolic accelerator for power system security assessment,” in *Proc. of the Conference on Intelligent System Application to Power Systems (ISAP 94)*, vol. 8, Montpellier, France, Sep. 1994, pp. 431–438.
- [217] B. J. MacLennan, “Grounding analog computers,” *Think*, vol. 2, pp. 48–51, 1993.
- [218] M. Hilbert and P. Lopez, “The world’s technological capacity to store, communicate, and compute information,” *science*, vol. 332, no. 6025, pp. 60–65, 2011.
- [219] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, “Dark silicon and the end of multicore scaling,” *IEEE Micro*, vol. 32, no. 3, pp. 122–134, 2012.
- [220] S. Borkar, “Design perspectives on 22nm CMOS and beyond,” in *Proceedings of the 46th Annual Design Automation Conference*. ACM, 2009, pp. 93–94.
- [221] V. V. Zhirnov, R. K. Cavin, J. A. Hutchby, and G. I. Bourianoff, “Limits to binary logic switch scaling—a gedanken model,” *Proceedings of the IEEE*, vol. 91, no. 11, pp. 1934–1939, 2003.
- [222] H. J. Caulfield and S. Dolev, “Why future supercomputing requires optics,” *Nature Photonics*, vol. 4, no. 5, pp. 261–263, 2010.
- [223] A. M. Childs, “Quantum algorithms: Equation solving by simulation,” *Nature Physics*, vol. 5, no. 12, pp. 861–861, 2009.
- [224] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, H. Chen, X. Peng, S. Kais, and J. Du, “Experimental realization of quantum algorithm for solving linear systems of equations,” *Physical Review A*, vol. 89, no. 2, p. 022313, 2014.
- [225] J. W. Mills, “The nature of the extended analog computer,” *Physica D: Nonlinear Phenomena*, vol. 237, no. 9, pp. 1235–1256, 2008.

-
- [226] S. Suh, A. Basu, C. Schlottmann, P. E. Hasler, and J. R. Barry, "Low-power discrete Fourier transform for OFDM: A programmable analog approach," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 58, no. 2, pp. 290–298, 2011.
- [227] H. T. Siegelmann and E. D. Sontag, "Analog computation via neural networks," in *Theory and Computing Systems, 1993., Proceedings of the 2nd Israel Symposium on the*. IEEE, 1993, pp. 98–107.
- [228] J. Cabessa and H. T. Siegelmann, "Evolving recurrent neural networks are super-turing," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, 2011, pp. 3200–3206.
- [229] NSA, "Why analog computation," NSA, Tech. Rep., 2008, approved for Release by NSA on 01-08-2008, FOIA Case #51551.
- [230] J. M. Bryant, "Analog Computation in the Digital Age," Analog Devices, Inc., Norwood, MA, Tech. Rep. T06126-0-5/06, 2006.
- [231] J. F. Traub, "A continuous model of computation," *arXiv preprint physics/0106045*, 2001.
- [232] S. Sethumadhavan, R. Roberts, and Y. Tsividis, "A case for hybrid discrete-continuous architectures," *Computer Architecture Letters*, vol. 11, no. 1, pp. 1–4, 2012.
- [233] G. A. Korn and T. M. Korn, "Electronic analog and hybrid computers," *New York: McGraw-Hill, 1964*, vol. 1, 1964.
- [234] R. St Amant, A. Yazdanbakhsh, J. Park, B. Thwaites, H. Esmailzadeh, A. Hassibi, L. Ceze, and D. Burger, "General-purpose code acceleration with limited-precision analog computation," in *Proceeding of the 41st annual international symposium on Computer architecture*. IEEE Press, 2014, pp. 505–516.
- [235] Y. Huang and S. Sethumadhavan, "Hybrid Continuous-Discrete Computer: from ISA to Microarchitecture," *ACM Transactions on Embedded Computing Systems*, vol. 1, no. 1, May 2013.
- [236] G. E. Cowan, R. C. Melville, and Y. Tsividis, "A VLSI analog computer/digital computer accelerator," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 1, pp. 42–53, 2006.
- [237] W. J. Karplus, "A new active-passive network simulator for transient field problems," *Proceedings of the IRE*, vol. 49, no. 1, pp. 268–275, 1961.
- [238] —, "A hybrid computer technique for treating nonlinear partial differential equations," *Electronic Computers, IEEE Transactions on*, vol. EC-13, no. 5, pp. 597–605, 1964.
- [239] —, *Analog simulation: solution of field problems*. McGraw-Hill New York, 1958.
- [240] W. Chen and L. P. McNamee, "Iterative solution of large-scale systems by hybrid techniques," *IEEE Transactions on Computers*, vol. 19, no. 10, pp. 879–889, 1970.

Bibliography

- [241] W. J. Karplus and R. A. Russell, "Increasing digital computer efficiency with the aid of error-correcting analog subroutines," *Computers, IEEE Transactions on*, vol. 100, no. 8, pp. 831–837, 1971.
- [242] V. Mishutin, "Concerning hybrid programs for hybrid computers," *Cybernetics and Systems Analysis*, vol. 11, no. 2, pp. 320–328, 1975.
- [243] K. Grandek, "Convergence of a hybrid iterative procedure," *Mathematics and Computers in Simulation*, vol. 22, no. 4, pp. 319–323, 1980.
- [244] J. Mandel and W. L. Miranker, "New techniques for fast hybrid solutions of systems of equations," *International Journal for Numerical Methods in Engineering*, vol. 27, no. 3, pp. 455–467, 1989.
- [245] C. C. Douglas, J. Mandel, and W. L. Miranker, "Fast hybrid solution of algebraic systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 11, no. 6, pp. 1073–1086, 1990.
- [246] A. Cichocki and R. Unbehauen, "Neural networks for solving systems of linear equations and related problems," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 39, no. 2, pp. 124–138, 1992.
- [247] C. Schlottmann, C. Petre, and P. Hasler, "Vector matrix multiplier on field programmable analog array," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1522–1525.
- [248] C. R. Schlottmann and P. E. Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 1, no. 3, pp. 403–411, 2011.
- [249] A. Basu, C. M. Twigg, S. Brink, P. Hasler, C. Petre, S. Ramakrishnan, S. Koziol, and C. Schlottmann, "RASP 2.8: A new generation of floating-gate based field programmable analog array," in *Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE*. IEEE, 2008, pp. 213–216.
- [250] P. C. Krause, T. A. Lipo, and D. P. Carroll, "Applications of analog and hybrid computation in electric power system analysis," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 994–1009, 1974.
- [251] J. D. Strong and G. Hannauer, "A Practical Approach to Analog Computers," *Instruments and Automation*, vol. 35, no. 8, Aug. 1962.
- [252] Y. Tamura, E. Dan, I. Horie, Y. Nakanishi, and S. Yokokawa, "Development of power system simulator for research and education," *Power Systems, IEEE Transactions on*, vol. 5, no. 2, pp. 492–498, 1990.
- [253] M. Goto, T. Kawai, S. Yokokawa, T. Suzuki, and others, "Advanced power system analogue simulator," *Power Systems, IEEE Transactions on*, vol. 5, no. 3, pp. 962–968, 1990.

-
- [254] M. Roitman, E. H. Watanabe, and F. J. Lyra, "Power systems analog simulation enhancement using a new programmable electronic model," *Power Systems, IEEE Transactions on*, vol. 4, no. 1, pp. 286–292, 1989.
- [255] L. H. Michaels, "The AC/hybrid power system simulator and its role in system security," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-91, no. 1, pp. 128–136, 1972.
- [256] R. Fried, R. Cherkaoui, and C. Enz, "Low-power CMOS, analog transient-stability-simulator for a two-machine power-system," *ISCAS '97 - Proceedings of 1997 IEEE International Symposium on Circuits and Systems, Vols I - Iv*, pp. 137–140 2832, 1997.
- [257] R. Fried, R. S. Cherkaoui, C. C. Enz, A. Germond, and E. A. Vittoz, "Approaches for analog VLSI simulation of the transient stability of large power networks," *IEEE Transactions on Circuits and Systems I-Fundamental Theory and Applications*, vol. 46, no. 10, pp. 1249–1263, 1999.
- [258] J. Gu and G. G. Karady, "Application of automatic tuning in transient stability analysis using analog VLSI," *2004 IEEE Power Engineering Society General Meeting, Vols 1 and 2*, pp. 1578–1582 2358, 2004.
- [259] J. Gu, G. G. Karady, and R. Farmer, "Real-time analysis of transient stability using reconfigurable analog VLSI," *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1207–1209, 2003.
- [260] G. G. Karady and J. Gu, "Employing reprogrammable analog VLSI in real-time analysis of transient stability," *Electric Power Systems Research*, vol. 77, no. 8, pp. 980–988, 2007.
- [261] C. Nwankpa, A. Deese, Q. Y. Liu, A. S. Leger, J. Yakaski, and N. Yok, "Power system on a chip (PSoC): Analog emulation for power system applications," *2006 Power Engineering Society General Meeting, Vols 1-9*, pp. 1429–1434 4472, 2006.
- [262] A. S. Deese and C. O. Nwankpa, "Circuit theoretical analysis of reconfigurable analog load emulation circuit," in *IEEE PowerTech Lausanne 2007, Vols 1-5*. IEEE, 2007, pp. 737–742 2235.
- [263] J. C. Jimenez, A. S. Leger, and C. O. Nwankpa, "Reconfigurable Phase Shifting Transformer for Analog Computation," in *Power Tech, 2007 IEEE Lausanne*, 2007, pp. 743–747.
- [264] A. S. Leger and C. O. Nwankpa, "Configuration of a large scale analog emulator for power system analysis," in *IEEE PowerTech Lausanne 2007, Vols 1-5*. IEEE, 2007, pp. 748–753 2235.
- [265] A. S. Leger and C. Nwankpa, "Analog emulation of transmission lines with time-varying phasors," in *Power & Energy Society General Meeting, 2009. PES '09. IEEE*, 2009, pp. 1–7.

Bibliography

- [266] V. Cecchi, A. S. Leger, K. Miu, and C. O. Nwankpa, "Modeling Approach for Transmission Lines in the Presence of Non-Fundamental Frequencies," *IEEE Transactions on Power Delivery*, vol. 24, no. 4, pp. 2328–2335, 2009.
- [267] A. S. Deese and C. O. Nwankpa, "Analogue emulation of power system load behaviour," *Iet Generation Transmission & Distribution*, vol. 3, no. 6, pp. 535–546, 2009.
- [268] A. S. Leger and C. Nwankpa, "OTA-based transmission line model with variable parameters for analog power flow computation," *International Journal of Circuit Theory and Applications*, vol. 38, no. 2, pp. 199–220, 2010.
- [269] A. S. Deese and C. O. Nwankpa, "Application of analog and hybrid computation methods to fast power system security studies," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 5, pp. 1140–1150, 2011.
- [270] —, "Utilization of FPAA technology for emulation of multiscale power system dynamics in smart grids," *Smart Grid, IEEE Transactions on*, vol. 2, no. 4, pp. 606–614, 2011.
- [271] A. S. Leger, A. Deese, J. Yakaski, and C. Nwankpa, "Controllable analog emulator for power system analysis," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 10, pp. 1675–1685, 2011.
- [272] P. Nagvajara, C. Nwankpa, and J. Johnson, "Reconfigurable Hardware Accelerators for Power Transmission System Computation," in *High Performance Computing in Power and Energy Systems*. Springer, 2013, pp. 211–228.
- [273] A. S. Deese and C. O. Nwankpa, "Design and Testing of Custom FPAA Hardware With Improved Scalability for Emulation of Smart Grids," *Smart Grid, IEEE Transactions on*, vol. 5, no. 3, pp. 1369–1378, 2014.
- [274] M. Kayal, R. Cherkaoui, I. Nagel, L. Fabre, F. Emery, and B. Rey, "Toward a power system emulation using analog microelectronics solid state circuits," in *IEEE PowerTech Lausanne 2007, Vols 1-5*. IEEE, 2007, pp. 726–730 2235.
- [275] I. Nagel, D. Seetharamdoo, L. Fabre, R. Cherkaoui, and M. Kayal, "Microelectronic Emulation for Power System Computation," in *Power Systems Computation Conference (PSCC), 16th*, 2008.
- [276] L. Fabre, I. Nagel, R. Cherkaoui, and M. Kayal, "High-speed, mixed-signal emulation for power system dynamic analysis," in *IEEE PowerTech Bucharest 2009, Vols 1-5*. IEEE, 2009, pp. 52–59.
- [277] I. Nagel, L. Fabre, R. Cherkaoui, and M. Kayal, "High-Speed Power System Stability Simulation Using Analog Computation," *Mixdes 2009: Proceedings of the 16th International Conference Mixed Design of Integrated Circuits and Systems*, pp. 560–565 716, 2009.

-
- [278] —, “High-speed power system stability simulation using analog computation: Systematic error analysis,” in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2010 Proceedings of the 17th International Conference*, 2010, pp. 514–518.
 - [279] —, “Microelectronic, high-speed data processing calculators for power system analysis: Comparison,” in *NEWCAS Conference (NEWCAS), 2010 8th IEEE International*, 2010, pp. 137–140.
 - [280] L. Fabre, I. Nagel, C. Meinen, R. Cherkaoui, and M. Kayal, “A mixed-signal platform dedicated to power system dynamic computation,” in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, 2011, pp. 1860–1863.
 - [281] L. Fabre, G. Lanz, I. Nagel, F. L. Conte, R. Cherkaoui, and M. Kayal, “An ultra high-speed, mixed-signal emulator for solving power system dynamic equations,” in *New Circuits and Systems Conference (NEWCAS), 2011 IEEE 9th International*, 2011, pp. 374–377.
 - [282] L. Fabre, D. Sallin, G. Lanz, T. Kyriakidis, I. Nagel, R. Cherkaoui, and M. Kayal, “A 3d architecture platform dedicated to high-speed computation for power system,” in *IEEE PowerTech Grenoble 2013*. IEEE, 2013, pp. 1–6.
 - [283] T. Kyriakidis, G. Lanz, D. Sallin, G. Lilis, L. Fabre, R. Cherkaoui, and M. Kayal, “A mixed-platform framework for Dynamic Stability Assessment,” in *Power and Energy Society General Meeting (PES), 2013 IEEE*, 2013, pp. 1–5.
 - [284] G. Lanz, L. Fabre, G. Lilis, T. Kyriakidis, D. Sallin, R. Cherkaoui, and M. Kayal, “Power network transient stability electronics emulator using mixed-signal calibration,” in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2013 Proceedings of the 20th International Conference*, 2013, pp. 369–373.
 - [285] I. Nagel, L. Fabre, M. Pastre, F. Krummenacher, R. Cherkaoui, and M. Kayal, “High-Speed Power System Transient Stability Simulation Using Highly Dedicated Hardware,” *Power Systems, IEEE Transactions on*, vol. 28, no. 4, pp. 4218–4227, Nov. 2013.
 - [286] G. Lilis, T. Kyriakidis, G. Lanz, R. Cherkaoui, and M. Kayal, “On the effect of Integration Algorithms on Reduced Accuracy Computational Architectures,” in *7th Electrical and Computer Engineering Student Conference (ECESCON)*, Thessaloniki, Greece, Apr. 2014.
 - [287] —, “Pipelined Numerical Integration on Reduced Accuracy Architectures for Power System Transient Simulations,” in *Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, ser. UKSIM ’14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 176–181.
 - [288] B. Stott, “Power system dynamic response calculations,” *Proceedings of the IEEE*, vol. 67, no. 2, pp. 219–241, 1979.
 - [289] J. D. Glover and M. S. Sarma, *Power system analysis and design*, 4th ed. Toronto, ON: Thomson, 2008.

Bibliography

- [290] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.
- [291] H. W. Dommel and N. Sato, “Fast Transient Stability Solutions,” *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-91, no. 4, pp. 1643–1650, 1972.
- [292] F. Lo Conte, “Interface I/O Universelle pour port USB,” MSc Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, Feb. 2006.
- [293] A. L. Shenkman, *Transient analysis of electric power circuits handbook*. Dordrecht: Springer, 2005.
- [294] L. O. Chua and P. Y. Lin, *Computer-aided analysis of electronic circuits: algorithms and computational techniques*. Prentice Hall Professional Technical Reference, 1975.
- [295] C. A. Desoer and E. S. Kuh, *Basic circuit theory*. Tata McGraw-Hill Education, 1984.
- [296] S. Cole and R. Belmans, “Matdyn, a new matlab-based toolbox for power system dynamic simulation,” *Power Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 1129–1136, 2011.
- [297] F. P. de Mello, J. W. Feltes, T. F. Laskowski, and L. J. Oppel, “Simulating fast and slow dynamic effects in power systems,” *Computer Applications in Power, IEEE*, vol. 5, no. 3, pp. 33–38, 1992.
- [298] A. S. Deese and C. O. Nwankpa, “Analogue emulation of power system load behaviour,” *Iet Generation Transmission & Distribution*, vol. 3, no. 6, pp. 535–546, 2009, 458SJ Times Cited:0 Cited References Count:32.
- [299] R. Cherkaoui, “Dynamique des Reseaux,” EPFL, Lausanne, Switzerland, 2010. [Online]. Available: <http://desl-pwrs.epfl.ch/>
- [300] J. J. Grainger and W. D. Stevenson, *Power system analysis*. McGraw-Hill New York, 1994, vol. 31.
- [301] G. Lanz, T. Kyriakidis, R. Cherkaoui, and M. Kayal, “Mixed-signal power system emulator extension to solve unbalanced fault transient stability analysis,” in *18th Power Systems Computation Conference (PSCC)*, Wroclaw, Poland, Aug. 2014.
- [302] A. J. Wood and B. F. Wollenberg, *Power generation, operation, and control*. John Wiley & Sons, 2012.
- [303] R. D. Zimmerman, C. E. Murillo, Sanchez, and R. J. Thomas, “MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education,” *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 12–19, 2011.
- [304] P. M. Anderson and A. A. Fouad, *Power system control and stability*, 1st ed. Ames: Iowa State University Press, 1977.

-
- [305] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*, ser. The EPRI power system engineering series. New York: McGraw-Hill, 1994.
- [306] University of Washington, "Power systems test archive," 1999, [Online; accessed 28-February-2015]. <http://www.ee.washington.edu/research/pstca/>. [Online]. Available: <http://www.ee.washington.edu/research/pstca/>
- [307] P. Subcommittee, "Ieee reliability test system," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-98, no. 6, pp. 2047–2054, Nov 1979.
- [308] O. Alsac and B. Stott, "Optimal load flow with steady-state security," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-93, no. 3, pp. 745–751, May 1974.
- [309] G. W. Bills, "On-line stability analysis study," North American Rockwell Information Systems Co., Anaheim, CA, Technical Report RP 90-1, Dec. 1970.
- [310] T. Van Cutsem and L. Papangelis, "Description, Modeling and Simulation Results of a Test System for Voltage Stability Analysis," Montefiore Institute, Université de Liège, Liege, Belgium, Tech. Rep. ver. 6, Nov. 2013.
- [311] A. Ishchenko, "Dynamics and stability of distribution networks with dispersed generation," PhD Thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, Jan. 2008.
- [312] S. R. Sanders, J. M. Noworolski, X. Z. Liu, and G. C. Verghese, "Generalized averaging method for power conversion circuits," *Power Electronics, IEEE Transactions on*, vol. 6, no. 2, pp. 251–259, 1991.
- [313] A. M. Stankovic and T. Aydin, "Analysis of asymmetrical faults in power systems using dynamic phasors," *Power Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 1062–1068, 2000.
- [314] A. M. Stankovic, S. R. Sanders, and T. Aydin, "Dynamic phasors in modeling and analysis of unbalanced polyphase AC machines," *Energy Conversion, IEEE Transactions on*, vol. 17, no. 1, pp. 107–113, 2002.
- [315] V. Venkatasubramanian, "Tools for dynamic analysis of the general large power system using time-varying phasors," *International Journal of Electrical Power & Energy Systems*, vol. 16, no. 6, pp. 365–376, 1994.
- [316] J. L. Blackburn, *Symmetrical components for power systems engineering*. CRC Press, 1993.
- [317] C. L. Fortescue, "Method of symmetrical co-ordinates applied to the solution of polyphase networks," *American Institute of Electrical Engineers, Transactions of the*, vol. 37, no. 2, pp. 1027–1140, 1918.

Bibliography

- [318] M. E. El-Hawary, *Electrical energy systems*, ser. The power engineering series. Boca Raton: CRC Press, 2000.
- [319] T. A. Short, *Electric power distribution handbook*. CRC press, 2014.
- [320] R. Teodorescu, M. Liserre, and P. Rodriguez, *Grid converters for photovoltaic and wind power systems*. John Wiley & Sons, 2011, vol. 29.
- [321] P. M. Anderson and P. Anderson, *Analysis of faulted power systems*. IEEE press New York, 1995, vol. 445.
- [322] M. Y. Borodulin, “Effect of numerical integration on critical time evaluation in power system stability studies,” in *Power and Energy Society General Meeting (PES), 2013 IEEE*, 2013, pp. 1–5.
- [323] J. Y. Astic, A. Bihain, and M. Jerosolimski, “The mixed Adams-BDF variable step size algorithm to simulate transient and long term phenomena in power systems,” *Power Systems, IEEE Transactions on*, vol. 9, no. 2, pp. 929–935, 1994.
- [324] N. J. Higham, *Accuracy and stability of numerical algorithms*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2002.
- [325] J. C. Butcher, *Numerical methods for ordinary differential equations*, 2nd ed. Chichester, England ; Hoboken, NJ: Wiley, 2008.

Theodoros Kyriakidis

Born on September 06, 1986. Greek Citizen.

Ecole Polytechnique Fédérale de Lausanne
Electronics Laboratory
Lausanne, Vaud, CH-1015

Phone: +41 21 69 34609, +30 6978 175 508
Email: thekyria@gmail.com



Education

Ph.D. Electrical Engineering, Ecole Polytechnique Fédérale de Lausanne, 2015. Thesis: A mixed-signal computer architecture and its application to power system problems

Ph.D. Research visit Montefiore Institute, Université de Liège, Aug 2014 - Jan 2015.

Dipl. Electrical Engineering and Computer Science, Aristotle University of Thessaloniki, 2010. Thesis: GyroScanField - Electromagnetic Field Measuring Instrument with Real-Time Visualization

Employment

Software Developer, Aristotle University of Thessaloniki (AUTH) Jan 2009 - Dec 2009
Development of a book e-distribution system: Database design, support and system upgrade

Network Engineering, Network Operations Center, AUTH Nov 2008 - Jun 2009
Installation, configuration and troubleshooting of network routers and ethernet switches

Electrical Panel Assembler, I&K Engineering & Automation Jun 2005 - Aug 2006
Assembly and spatial design of electrical automation panels

IT support, Information Technology Center, AUTH Nov 2005 - Feb 2006
Hardware and software troubleshooting and support

Skills and Qualifications

Research Skills & Interests

Electronics: reconfigurable computing, analog computing, embedded systems.

Power systems: dynamics, security assessment, optimization, power flow, smart grid architecture.

Mathematics: numerical analysis, (non-)linear systems, linear algebra, ODEs/DAEs, regression analysis.

Computing: (scientific) programming, network engineering, databases, sensor fusion.

Qualifications

FPGAs (h/w & s/w co-design), MCUs, PCBs, C[+]/[#], MATLAB, PHP, [My]SQL, VHDL, Networking - Device Management & conf. (Cisco), Server admin (W/LAMP), CMSs, Latex, Versioning (svn, git).

Languages

Greek (native)	English (prof. - C2)	Spanish (ave. - B2)
German (bas. - C1)	Italian (ave. - B2)	French (ave. - n/a)

Soft Skills & Other Competencies

Leadership, integrity, fast learning ability, discipline, work ethic, adaptability, multi-cultural living and working experience, polyglotism, synthetic thinking, driving licences (A, B).

Affiliations

Member, Technical Chamber of Greece, 2010-present.

Member, IEEE, 2008-present.

Member, IEEE Power & Energy Society, 2010-present.

Member, IEEE PES Swiss Chapter.

Member, IEEE Smart Grid Technical Community.

Member, IEEE Computer Society Member, 2008-2012 & 2014-present.

Member, ACM, 2014-present.

Publications (Selected)

Awards and Honors

Swiss National Science Foundation (SNSF) Doc.Mobility fellowship, 2014.

ECESCON 7 best paper award - 1st in *Energy* section, 4th overall, 2014.

Texas Instruments European Analog Design Contest - 4th prize, 2013.

Nano-Tera Artist Summer School best poster award - 1st prize, 2013.

Scholarship by the Fulbright Foundation, 2010.

Academic Excellence Scholarship by the Greek National Scholarship Foundation (IKY), 2008 & 2004.

Honorary Diploma by the Hellenic Mathematical Society, 2000.