

# Real-Time Hardware-in-the-Loop Emulation of Electric Machines for Electrified Transportation Systems

by

Behzad Jandaghi

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy  
in  
Energy Systems

Department of Electrical and Computer Engineering

University of Alberta

© Behzad Jandaghi, 2018

# Abstract

Design and prototyping of electric power components and systems employing off-line transient simulation tools are inefficient due to, first, time consumption of sequential processors for several cycles of testing, and second, inaccuracy arising from simplifications in the modeling of the entire system. Hardware-in-the-loop (HIL) emulation can massively accelerate the design procedure and provide a highly accurate testing platform for the manufactured prototype devices to interact with the rest of the system model on a simulator in a nondestructive environment before commissioning. The scenario is efficient when the emulated system computations can be executed in real-time with a specified time-step which is small enough to model all the transients of the system. Field programmable gate array (FPGA) proposes an attractive platform for real-time simulation due to reconfigurability, widely paralleled, deeply pipelined architecture and low input/output latencies.

As a key component of the power systems with a wide variety of applications, testing of electric machines in the design and control procedure for the purposes of energy efficiency and performance improvement is becoming increasingly demanding. The complex structure of electric machines makes real-time simulation challenging mainly due to mechanical movement and magnetic nonlinearity.

This thesis addresses challenges and solutions for real-time HIL emulation of electric machines. Comprehensively, all three widely used electric machine models including q-d vector model, magnetic equivalent circuit, and finite el-

ement method models are investigated for real-time simulation of low power rotary induction motor and high power linear induction motor for magnetic levitation application. An evaluation in terms of real-time step-size and accuracy as well as FPGA hardware resource utilization corresponding to each model is provided. The validation of results with off-line transient simulation and finite element tools and experimental measurements demonstrates the accuracy and efficiency of the proposed approaches.

# Preface

This thesis is an original work by Behzad Jandaghi. As detailed in the following, some chapters of this thesis have been published, or under review for publication as scholarly articles in which Prof. Venkata Dinavahi was the supervisory author and has contributed to concepts formations and the manuscript composition.

Chapter 2 of this thesis has been published as:

- B. Jandaghi, and V. Dinavahi, “Hardware-in-the-loop emulation of linear induction motor drive for maglev application”, *IEEE Transactions on Plasma Science*, vol. 44, no. 4, pp. 679-686, Mar. 2016.

Chapter 3 of this thesis has been published as:

- B. Jandaghi, and V. Dinavahi, “Prototyping of nonlinear time-stepped finite element simulation for linear induction machines on parallel reconfigurable hardware”, *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 7711-7720, Apr. 2017.

Chapter 4 of this thesis has been published as:

- B. Jandaghi, and V. Dinavahi, “Real-time FEM computation of nonlinear magnetodynamics of moving structures on FPGA for HIL emulation”, Accepted in *IEEE Transactions on Industrial Electronics*, pp. 1-10, Feb. 2018. (DOI: 10.1109/TIE.2018.2801843)

Chapter 5 of this thesis is under review as:

- B. Jandaghi, and V. Dinavahi, “Real-time HIL emulation of faulted electric machines based on nonlinear MEC model”, Under review in *IEEE Transactions on Energy Conversion*, pp. 1-8, Jan. 2018. (ID: TEC-00018-2018)



*To my beloved family,  
for their love, unlimited support, and encouragement*

# Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Venkata Dinavahi for his support, encouragement, and guidance through my research at the University of Alberta.

It is an honor for me to extend my gratitude to my Ph.D. committee members Prof. Yasser Abdel-Rady I. Mohamed, Prof. Ali Khajehoddin, and Prof. John Salmon, and my external examiner Prof. Narayan Kar for reviewing my thesis and providing thoughtful and constructive comments.

I would like to thank my fellow colleagues at RTX-Lab and also my friends, especially Dr. Nariman Roshandel Tavana, for their feedback, cooperation and of course friendship.

I would like to thank T. Morris from American Maglev Technology, and Prof. K. Davey from the University of Texas at Austin for providing the data of the industrial sample linear induction machine required for this thesis.

I would appreciate the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and Alberta Innovates Technology Futures (AITF).

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Real-Time Hardware-in-the-Loop Emulation . . . . .	2
1.1.1	Real-Time simulation . . . . .	2
1.1.2	Hardware-in-the-Loop . . . . .	3
1.2	FPGA Architecture . . . . .	4
1.3	Electric Machine Models . . . . .	8
1.3.1	Lumped q-d Vector Model . . . . .	8
1.3.2	Magnetic Equivalent Circuit (MEC) Model . . . . .	9
1.3.3	Finite Element Method (FEM) Model . . . . .	9
1.4	Literature Review . . . . .	10
1.5	Thesis Motivation . . . . .	12
1.6	Thesis Objectives . . . . .	13
1.7	Thesis Outline . . . . .	15
<b>2</b>	<b>Real-Time Lumped q-d Model of Linear Induction Motor</b>	<b>17</b>
2.1	Introduction . . . . .	17
2.2	Hardware-in-the-Loop Emulation of LIM Drive . . . . .	19
2.2.1	Hardware Realization of LIM . . . . .	19
2.2.2	Hardware Realization of the Drive System . . . . .	22
2.2.3	FPGA Implementation . . . . .	25
2.3	Results and Discussion . . . . .	28
2.3.1	Real-Time Simulation Results . . . . .	28
2.3.2	Floating-Point vs. Fixed-Point Comparison . . . . .	30
2.3.3	Result Verification . . . . .	33
2.4	Summary . . . . .	35

<b>3</b>	<b>Real-Time Magnetic Equivalent Circuit Model of Faulted Rotary Induction Motor</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Novel MEC-Based RT-TLM Emulation . . . . .	39
3.2.1	Transmission Line Modeling (TLM) Method . . . . .	39
3.2.2	TLM-Based MEC Modeling . . . . .	39
3.2.3	Matrix Re-Ordering, Partially Pre-Calculated LU Decomposition and Reduced Order Sparse Solver . . . . .	44
3.2.4	Parallelism Exploitation . . . . .	47
3.3	MEC-Based RT-TLM Hardware Implementation . . . . .	47
3.4	Real-Time Simulation Results . . . . .	50
3.4.1	Transient Start-Up . . . . .	50
3.4.2	Steady-State . . . . .	53
3.4.3	Hardware Resource . . . . .	58
3.4.4	Timing Analysis . . . . .	58
3.5	Summary . . . . .	59
<b>4</b>	<b>Hardware Acceleration of Finite Element Method Model of Linear Induction Motor</b>	<b>61</b>
4.1	Introduction . . . . .	61
4.2	Finite Element Hardware Acceleration . . . . .	63
4.2.1	Model Formulation and Numerical Solution . . . . .	63
4.2.2	Linear Sparse Solver . . . . .	67
4.2.3	Hardware Emulation . . . . .	69
4.2.4	FPGA Implementation . . . . .	70
4.3	Results, Verification and Discussion . . . . .	73
4.4	Summary . . . . .	81
<b>5</b>	<b>Real-Time Finite Element Method Model of Linear Induction Motor</b>	<b>82</b>
5.1	Introduction . . . . .	82

5.2	Novel RT-TLM Method for Real-Time Finite Element Modeling Emulation . . . . .	84
5.2.1	FEM Element Equivalent Circuit . . . . .	84
5.2.2	Strategies for Enforcing Real-Time Execution . . . . .	87
5.3	FPGA Design Architecture for the Novel RT-TLM FEM Emulation . . . . .	90
5.3.1	Hardware Architecture . . . . .	90
5.3.2	FPGA Implementation . . . . .	93
5.4	Real-Time Hardware Emulation Results . . . . .	95
5.4.1	Real-Time Results . . . . .	95
5.4.2	Results Verification and Accuracy Evaluation . . . . .	95
5.4.3	Resource Utilization and Scalability . . . . .	100
5.4.4	Timing Analysis . . . . .	102
5.5	Summary . . . . .	103
<b>6</b>	<b>Conclusion</b>	<b>105</b>
6.1	Contributions of This Thesis . . . . .	106
6.2	Directions for Future Work . . . . .	108
	<b>Appendix A Device Specifications</b>	<b>132</b>

# List of Tables

2.1	Hardware Resource Utilization . . . . .	33
3.1	Hardware resource utilization . . . . .	58
3.2	Latencies of each state in real-time processing . . . . .	58
4.1	Latencies for Each State of the Finite State Machine of Fig. 4.4	79
4.2	Execution time latencies of the JMAG-Designer <sup>®</sup> and Hardware prototype . . . . .	80
4.3	Hardware Resource Utilization . . . . .	80
5.1	Hardware resource utilization . . . . .	101
5.2	Latencies of each module . . . . .	103
A.1	Specifications of the studied LIM. . . . .	132
A.2	Specifications of the studied rotary induction motor. . . . .	133
A.3	Hardware resource utilization . . . . .	133

# List of Figures

1.1	CHIL and PHIL configurations. . . . .	4
1.2	FPGA hardware architecture. . . . .	6
1.3	FPGA hardware design procedure. . . . .	7
2.1	Taxonomy of literature on linear induction motor. . . . .	18
2.2	Structure of linear induction motor. . . . .	20
2.3	Massively parallel state-space hardware implementation of LIM model. . . . .	23
2.4	LIM rotor field oriented control scheme. . . . .	25
2.5	Massively parallel hardware implementation of rotor field oriented control units. . . . .	26
2.6	Signal integration in the hardware emulation architecture for the LIM drive system (modules scaled by latency). . . . .	29
2.7	Floating-point real-time results of LIM [time: 1 sec/div, speed: 8 m/s/div, current: 1440 A/div, secondary flux: 0.2 Wb/div, propulsion force: 12000 N/div, levitation force: 16000 N/div].	31
2.8	Fixed-point and floating-point accuracy comparison. . . . .	32
2.9	Finite element result of flux and current distribution in JMAG <sup>®</sup> . . . . .	34
2.10	LIM experimental set up, courtesy of American Maglev Technology and Prof. K. Davey, used in [131] at Center of Electromechanics at the University of Texas at Austin. . . . .	34
2.11	Torque vs. machine speed with experimental, real-time and finite element analysis. . . . .	35

3.1	The TLM application procedure on MEC. (a) One tooth of the stator and rotor of an induction motor, (b) MEC model, (c) Transmission line decoupling, (d) MEC-based TLM network and the decoupled nonlinear permeances. . . . .	41
3.2	Proposed RT-TLM MEC matrix LU decomposition procedure.	46
3.3	Proposed MEC-based RT-TLM FPGA implementation state diagram. . . . .	49
3.4	Massively paralleled and deeply pipelined FPGA hardware architecture. (a) S2: reduced-order sparse LU decomposition, (b) S4: sparse forward elimination and backward substitution, (c) S7: next incident magnetic vector potentials. . . . .	51
3.5	Real-time simulator hardware set up. . . . .	52
3.6	Stator winding phase current transients during no load start-up.	52
3.7	Rotor speed transients during no load start-up. . . . .	53
3.8	Steady-state stator current result validation. (a) Case 1: Healthy, (b) Case 2: Stator winding inter-turn short circuit fault, (c) Case 3: Rotor broken bar fault, (d) Case 4: Rotor broken end ring fault, (e) Case 5: Supply 5 <sup>th</sup> and 7 <sup>th</sup> harmonics, (f) Case 6: Unbalanced supply. . . . .	54
3.9	Steady-state torque and current vs. speed from no load to full load. . . . .	56
3.10	FEM results of Jmag-Designer <sup>®</sup> for healthy and broken rotor bar. (a) Magnetic flux density, (b) Current density. . . . .	57
4.1	Linear induction machine (a) structure and boundary conditions (b) triangular element (c) one slot meshing during movement. . . . .	64
4.2	Proposed linear sparse solver scheme. . . . .	68
4.3	Pipelined and paralleled hardware architecture of the system equation formation for the finite element emulator of LIM. . .	71
4.4	Finite state machine (FSM) of the hardware prototype . . . .	72



4.5	Hardware prototype results comparison with JMAG-Designer <sup>®</sup> (a) magnetic flux density vector (b) current density (c) magnetic flux density distribution. . . . .	74
4.6	Nodal magnetic flux density of the hardware prototype and JMAG-Designer <sup>®</sup> in the highest field intensity region shown by scope in Fig. 4.5. . . . .	76
4.7	LIM propulsion and levitation forces vs. slip frequency . . . . .	78
5.1	The TLM application procedure on an eddy current FEM el- ement. (a) FEM element equivalent circuit, (b) Transmission line decoupling, (c) Lattice diagram of the transient incident and reflected potential waves, (d) TLM equivalent circuit, (e) Decoupled nonlinear conductances. . . . .	88
5.2	One pole pitch structure of the linear induction motor case study.	89
5.3	Massively paralleled and deeply pipelined hardware architec- ture. (a) The next incident potential, (b) Forward elimination and (c) Backward substitution. . . . .	91
5.4	Finite state machine of the novel RT-TLM FEM hardware. . .	94
5.5	Novel RT-TLM method real-time oscilloscope results of the FPGA. (a) Primary mover speed 15 m/s/div, (b) Supply fre- quency 50 Hz/div, (c) Propulsion force 15,000 N/div, (d) Levi- tation force 15,000 N/div, (e) Supply phase current 2,500 A/div.	96
5.6	The novel RT-TLM results validation with Jmag-Designer <sup>®</sup> . (a) Propulsion force vs. speed, and (b) Levitation force vs. speed.	97
5.7	The novel RT-TLM method losses, validated by conventional FEM. . . . .	98
5.8	Real-time results validation with experimental, Jmag-Designer <sup>®</sup> , conventional FEM, conventional TLM, and the novel RT-TLM method. . . . .	99
5.9	The novel RT-TLM results. (a) Magnetic field distribution, (b) Current density distribution. . . . .	100

5.10	Floor plan of the mapped novel RT-TLM FEM design on Virtex UltraScale+ XCVU9P FPGA. . . . .	101
5.11	Number of clocks per time-step versus the position number. . .	102

# Chapter 1

## Introduction

Electric machines have been one of the key components of the power systems with a wide range of applications from power generation to consumption, which form 65% of the total energy consumption [1]. In particular, their application in electrified transportation systems from low power electric vehicles (EV) and hybrid electric vehicles (HEV) to high power magnetic levitation (maglev) systems is rapidly growing due to their green operation, fuel efficiency, low maintenance costs, noiseless operation, high torque during acceleration, and regenerative capability during deceleration [2–5]. With the concern of cost efficiency associated with electric vehicles, design and control of the electric machines become more crucial than ever, which highlights the importance of accurate testing and validation of electrical machines and drives [6].

Prototyping an electric machine drive system targets the design and performance evaluation of the controllers, testing protection devices during fault as well as the interaction of the electric machine with the power converter and the grid. The process requires several cycles of testing, expensive equipment, human resources, time for construction, power consumption, and physical space in a destructive environment. Conventionally, off-line simulation tools were used to evaluate the behavior of an electric machine drive system and several well-known softwares are developed including JMAG-Designer<sup>®</sup>, ANSYS Maxwell<sup>®</sup>, COMSOL Multiphysics<sup>®</sup>, MATLAB/Simulink<sup>®</sup>, PSCAD/EMTDC<sup>®</sup>, etc. for different objectives [7, 8]. However, off-line simulations require modeling and simulation of the entire system, which suffer from inac-

curacy of the results due to simplifications in the modeling of each component, and also time consumption of the modeling and simulation execution.

Hardware-in-the-loop (HIL) real-time simulation offers a solution for the aforementioned problem by performing the entire calculations of the system within a time constraint that happens in the real world [9]. However, since generally the sequential processors are not able to satisfy the timing constraint especially with detailed models, parallel processors including FPGAs [10], graphics processing unit (GPU) [11], and multi-core CPUs [12–15] are utilized in real-time simulators. HIL simulation is an efficient and cost-effective approach to predict the behavior of a system prior to manufacturing and commissioning by emulation of the behavior of electric components on a real-time simulator interacting with actual device under test in a safe, flexible, and realistic conditions [16].

## 1.1 Real-Time Hardware-in-the-Loop Emulation

### 1.1.1 Real-Time simulation

Real-time simulation is achieved when the entire system computations are carried out within a specified time-step, which is chosen small enough in such a way that the required accuracy is satisfied and desired transients are captured [17]. Based on Nyquist theorem, at least two data points per cycle are required to reproduce a signal. In other words, the sampling rate should be at least twice the highest frequency component of the signal that needs to be reproduced. Assuming the computation of the entire system for the  $n^{th}$  time-step takes  $T_c(n)$  and the specified real-time step-size is  $T_s$ , the satisfaction of the real-time simulation requires:

$$T_c(n) \leq T_s \quad n = 1, 2, 3, \dots, N \times T_s, \quad (1.1)$$

where  $N$  is the number of time-steps of the simulation. It is worth emphasizing that  $T_c(n)$  may change for each time-step, causing a remaining time between the  $T_c(n)$  and  $T_s$  known as *idle time*. Violating  $T_c(n)$  from  $T_s$  is called *overrun*,

where the real-time simulation is no longer valid.

### **1.1.2 Hardware-in-the-Loop**

The HIL simulation allows testing prototype devices in interaction with the rest of the system on the real-time simulator [18]. Real-time HIL technology is rapidly growing as the superior, reliable and cost-effective testing method due to the following advantages:

- minimizing the error of the testing set up by removing inaccurate modeling of the actual devices
- providing a safe, flexible and nondestructive environment for testing especially during faults
- rapid control prototyping by quickly develop, iterate and test control strategies
- simulation acceleration due to real-time execution especially for population-based designs and automation tests

The efficiency of the HIL simulation is based on the model fidelity and the minimum achievable time-step of the simulated components on the real-time simulator, which are usually in contradiction.

#### **a) Control Hardware-in-the-Loop (CHIL)**

Traditionally, the newly designed and fabricated control systems are tested directly on the actual power device under control in the field or laboratories. Although the scenario has the advantage of high fidelity testing, it is unsafe in the case of fabrication errors and not flexible to evaluate the system performance in a wide range of possible operating conditions including faults.

The CHIL systems offer an excellent alternative by testing the complex control, protection and monitoring systems interfaced with the real-time simulated model of the power devices. Although the interface in CHIL is usually low power (1 to 10V, few mA) for transferring control signals through optical

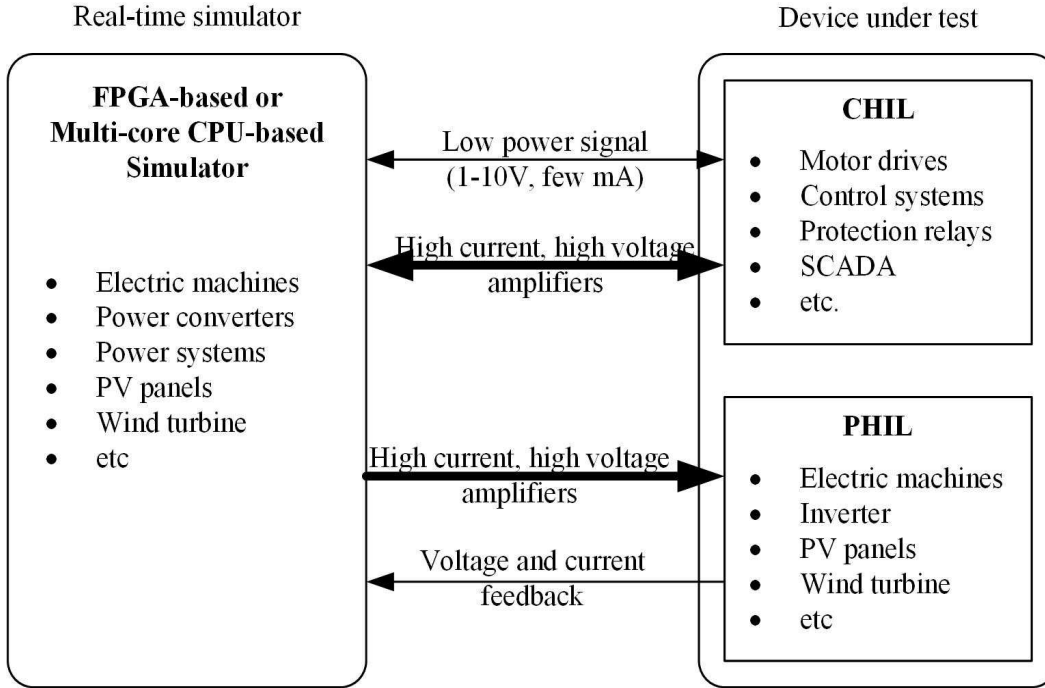


Figure 1.1: CHIL and PHIL configurations.

fibers and communication systems, depending on the case study high power (60 to 200V, 5A to 100A) may be required through amplifiers besides the low power feedback signals.

### b) Power Hardware-in-the-Loop (PHIL)

In the PHIL scenario, the actual device under test is an electric power component including an electric machine or a power converter. The interface incorporates both high power interface through amplifiers and low power measurement sensors and control commands. It can be also used for rapid control prototyping of controllers before fabrication. The CHIL and PHIL configurations are shown in Fig. 1.1.

## 1.2 FPGA Architecture

Most simulation tools employ traditional CPUs, where an arithmetic logic unit (ALU) is able to perform only one operation at a time. The sequentiality of CPUs makes them inappropriate for real-time simulation, since they are not

able to replicate the behavior of a system in real-time. Multi-core CPUs are currently used in a majority of real-time simulators with some degree of parallelism limited by the number of CPU cores [19–21]. In addition, multi-core CPUs have high input/output (I/O) latency, which is a major disadvantage in high-frequency systems such as PWM-based power converter control systems [22].

Recently, FPGAs have gained a lot of attention in real-time simulation, especially for large and complex systems, due to their abundant parallel hardware architecture, fast prototyping, reconfigurability, low cost, and fast I/O interface [23,24]. The generic circuit of FPGAs can be programmed for a wide variety of applications. As shown in Fig. 1.2, FPGAs consist of a 2-D array of computational logic blocks (CLBs), memory elements, DSP blocks, I/O blocks, and switching matrices, which are programmable through the computer aided design (CAD) tools to implement the desired functionality [25,26].

In this work Xilinx<sup>®</sup> Virtex-7 XC7VX485T and Virtex UltraScale+XCVU9P FPGAs are used, and the hardware resource of each is presented in the Appendix A.

A complete process of FPGA prototyping is shown in Fig. 1.3. Starting from the design entry, two methods can be employed. First, the VHSIC Hardware Description Language (VHDL) program can be automatically generated by schematic blocks available in Xilinx System Generator (XSG) as a toolbox of Matlab, which is an intuitive and simple approach for beginners [27]. However, the output VHDL program is generated based on an optimization algorithm, which may lack efficiency and flexibility for the case study. Second, creating the design entry directly by use of handwritten textual programming language, which is more efficient at the cost of complexity and requiring specialty in FPGA design [28].

ISE<sup>®</sup> and its later version, Vivado<sup>®</sup>, are the FPGA design tools developed by Xilinx<sup>®</sup>. Once the VHDL program file created, the tools are used for the behavioral simulation for results verification, synthesis to determine the logic circuits, and implementation of the design including translate, map, place and route to generate the programming file of FPGA through the JTAG cable. The

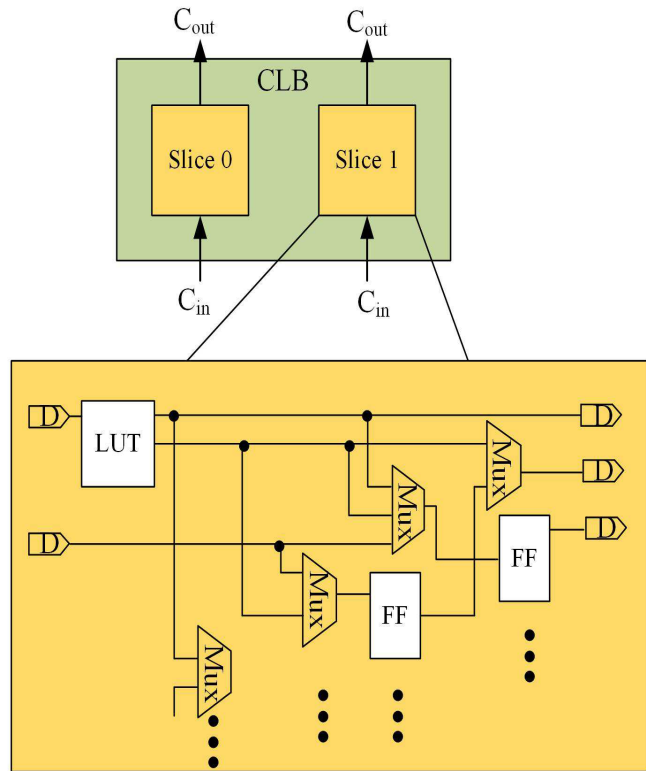
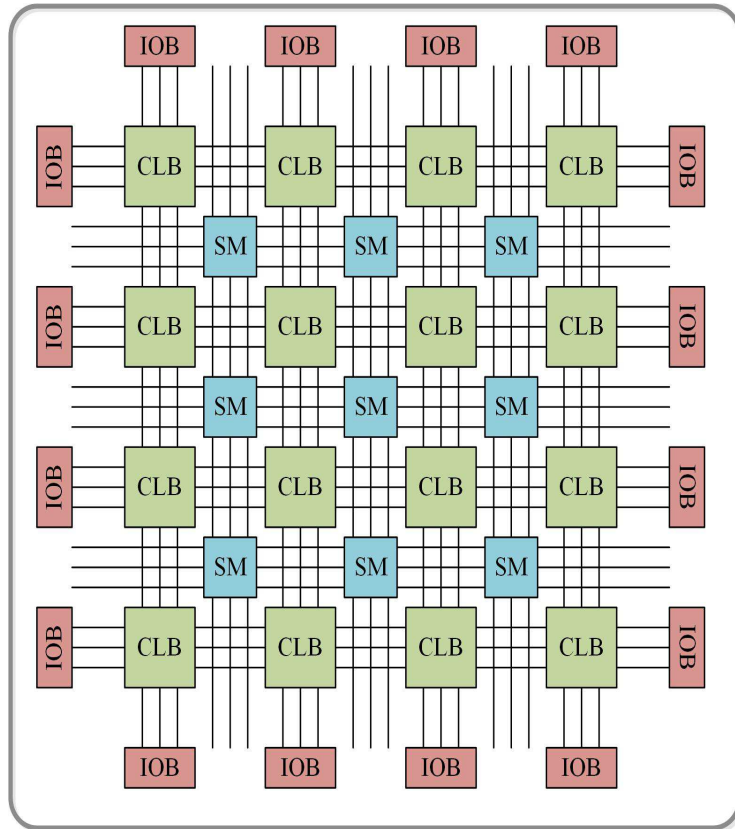


Figure 1.2: FPGA hardware architecture.



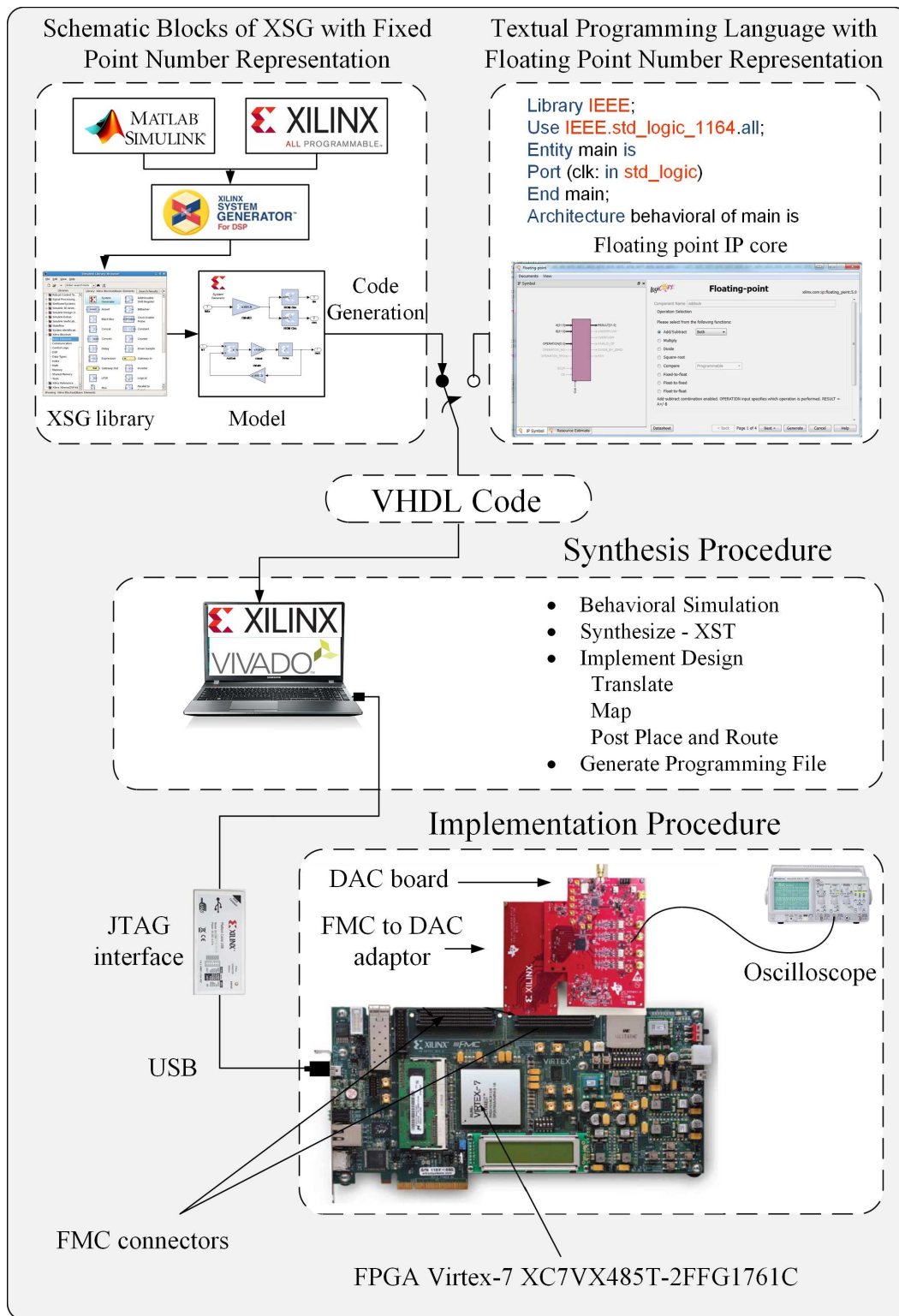


Figure 1.3: FPGA hardware design procedure.

digital to analog converter (DAC) board enables illustrating the digital signals to the analog oscilloscope.

In hardware design, compromising between hardware resource utilization, the simulation speed, and the accuracy is challenging and depends on the case study. Massively parallel implementation of algebraic calculations (single data per logic resource per time-step) implies requiring more resource utilization with the advantage of reaching less time-step and simplicity. However, fully pipelined implementation (multiple data per logic resource per time-step) employs an IP core for several times in each time-step, which implies utilizing less hardware resources while increasing the design complexity as well as having larger time-step. In most implementations for optimal hardware design both parallel and pipelined architectures should be used.

### 1.3 Electric Machine Models

In modeling of electric machines, there is a trade-off between the computational intensity and accuracy. The q-d vector model of an electric machine is simple and mostly used for control purposes [29], the magnetic equivalent circuit (MEC) model has a medium computational burden with a fair accuracy [30–32], and the accuracy offered by the finite element method (FEM) model with an extensive computational burden is more favorable for design purposes [33]. In this section, the three widely used electric machine models targeted for real-time simulation are described.

#### 1.3.1 Lumped q-d Vector Model

The dynamic behavior of an AC machine in phase domain can be represented based on an equivalent circuit model as a function of rotor position. The model represents the magnetic coupling between different parts of the machine through a set of self and mutual inductances of the windings [34]. The q-d model describes the AC parameters of the equivalent circuit model in a quadrature and direct reference frame with DC parameters rotating with an arbitrarily angular velocity mainly for the ease of control.

The model is widely used in electric machine modeling for control purposes, due to its simplicity and low computational burden. However, the model suffers from low accuracy, inability to model space harmonics, and its limitations to study different type of internal machine faults. Although in most cases the inductances are considered linear for simplicity, the nonlinearity of mutual inductances can be considered [35–37]. Despite the model accuracy and limitations, the low computational burden of the q-d model makes it ideal for real-time simulation of electric machines with low time-steps.

### 1.3.2 Magnetic Equivalent Circuit (MEC) Model

MEC model represents the dynamic behavior of an electric machine through a network of lumped nonlinear magnetic permeances of the flux paths and magnetomotive force (MMF) of the windings [38,39]. The model is developed for both 2-D [40,41] and 3-D [42,43] cases.

The model is sometimes used in electric machine modeling for population-based design purposes, where numerous number of iterations are required [44,45]. Since the model is based on the geometrical data, it is able to consider nonlinearity of the iron core, space harmonics, and accommodate internal machine faults. In comparison to the lumped q-d model, MEC has higher accuracy and computational burden. The medium size of the system of nonlinear equations of electric machines makes the model challenging for real-time simulation.

### 1.3.3 Finite Element Method (FEM) Model

Finite element method decomposes the study domain into a finite number of elements (often triangular) and nodes, which is known as meshing. In electromagnetic case studies of FEM, the magnetic vector potential is assumed to linearly change between the nodes of an element. A high-quality mesh employs elements as small as possible with equilateral edges to minimize the assumption error. The magnetic vector potential of the nodes across the study domain is calculated through Maxwell’s partial differential equations with boundary conditions.

The model is widely used for electric machine modeling for design purposes due to its accuracy and ability to consider all phenomena including nonlinearity, space harmonics, and faults. The accuracy of the method is dependent on the number of nodes and elements. The number of nodes determines the size of the system of nonlinear equation. Neither 2-D or 3-D cases are studied for real-time simulation of electric machines due to its high computational intensity, mainly arisen from the linear solver in each nonlinear iteration of the system of equation [46–48].

## 1.4 Literature Review

The FPGA is widely used for real-time simulation of power electronics [49–59] and power systems [60–67]. In the field of electric machine and drives, numerous studies have been conducted with lumped q-d vector model for various types of electric machines including rotary induction machines [68–74], permanent magnet synchronous machines [75–78], synchronous machines [79–81], doubly-fed machines [82], and DC machines [83, 84].

In some studies, the nonlinearity of the iron core is taken into account in the lumped q-d vector model through off-line pre-calculation of nonlinear inductance matrix as a function of rotor position and stator currents [85–95]. The pre-calculated data are stored in a look-up table and are used during q-d-based real-time simulation depending on the operating condition. However, creating the look-up tables to cover a dense grid of the variables range is very time consuming, memory inefficient, and causes incorrect transient behavior due to jump discontinuities. Magnetic nonlinearity of iron core in the lumped q-d model without the use of FEM is considered in [96]. With the lumped q-d model, time-steps in the range of nanoseconds to a few microseconds have been achieved.

FPGA-based real-time simulation of a permanent magnet linear synchronous machine (PMLSM) using analytical space harmonic model is proposed in [97], where the minimum achievable time-step of  $1.8 \mu s$  is reported. The real-time magnetic equivalent circuit model and finite element method model of power

transformers on FPGA are proposed in [98,99] and [100], respectively. However, unlike electric motors, power transformers do not involve mechanical movement.

Physics-based models of electric machines, including MEC and FEM, suffer from the change of matrix equations in every nonlinear iteration, due to saturation of the iron core, and every time-step, due to mechanical movement. As a result, a new linear system of equations needs to be solved in every iteration, which make the models time consuming and very challenging for real-time simulation.

In [101], a new transmission line modeling (TLM) algorithm for the MEC-based solution of induction machines is proposed to decouple the nonlinear magnetic equations. A look-up table is used to solve the decoupled nonlinear elements, where an acceleration rate of 8.7 times is achieved in comparison to the conventional MEC method. Later in [102], MEC-based real-time simulation of one pole pitch of the induction machine with the time-step of  $150\mu s$  has achieved on multi-core CPU-based real-time simulator.

In [103], the FPGA-based real-time simulation of a switched reluctance motor (SRM) using MEC model is performed on FPGA. However, due to the salient magnetic structure of this type of machine, the MEC system of equation size is small and cannot be generalized for other electric machines including induction motors. In [104], the MEC model of an induction motor is used for real-time simulation on FPGA using massively paralleled and deeply pipelined Gauss-Jordan elimination solver and the minimum time-step of  $400\mu s$  is achieved. However, the proposed hardware prototype is not able to accommodate internal machine faults since the study domain is reduced to one pole pitch of an induction motor under anti-periodicity boundary condition assumption.

Up to now, no research has been conducted on real-time finite element method simulation of electric machines on FPGA for HIL application, which is the focus of this thesis.

## 1.5 Thesis Motivation

With rapidly growing demand for electrified transportation systems from low power EVs to high power MagLev systems, the concern of cost efficiency associated with electric machines and the drive systems become more crucial than ever, which highlights the importance of accurate testing and validation.

Since off-line simulation tools with conventional CPU processors cannot emulate the behavior a system in the real world due to the timing constraints, real-time simulators in a HIL scenario are substituted to provide an accurate, flexible, and safe testing platform to evaluate the system efficiency in a wide range of operating conditions and fault scenarios, which cannot be easily and safely created. Recently, as a part of the power system, the real-time HIL simulation of electric machines has become the topic of interest for many industries including electric vehicle companies.

Besides current multi-core CPU-based processors, the massively paralleled hardware architecture of FPGAs make them an attractive option as the processor of real-time simulators for large and complex systems. As the literature indicates, numerous studies have been conducted on electric machine lumped parameter models; however, real-time simulation of electric machine drives based on physics models on FPGA gained less attention and has potential for research.

An electric machine can be considered as a complex multi-physics energy conversion device due to its complicated geometry, nonlinear properties, and mechanical movement. The main challenging part of HIL simulation is to have a high fidelity electric machine model for evaluation of the system performance over a wide range of operating conditions and fault scenarios that may or may not be detected by the protection systems. As the bottleneck in the minimum achievable real-time simulation time-step to interact with fast dynamic power converters, a compromise between detailed modeling and computational burden is targeted.

Focusing on electric machine detailed models, the problem of high computational intensity is a challenge in order to achieve a minimum time-step

as small as possible to capture the transients of the electric machine. The computational intensity of the models is mainly due to the solution of a nonlinear system of equation, which is carried out by a linear solver with the size of number of model nodes in nonlinear iterations. As a result, a different set of linear equations needs to be solved in every nonlinear iteration, due to the nonlinearity of the magnetic materials, and every time-step, due to the mechanical movement. The problem size highly affects the execution time of the linear solver.

Since VHDL programing requires expertise in FPGA circuit design to assign each operation signals to a particular hardware module, it is difficult for beginners in comparison with conventional off-line tools. To overcome this problem, FPGA software developers provided other tools such as Vivado HLS, which automatically converts C/C++ to VHDL program, and Xilinx system generator (XSG), which employs schematic blocks in Matlab/Simulink to generate VHDL program. However, the design output efficiency is dependent on the general purpose optimization algorithm of the tools. The textual programming language gives the flexibility to customize FPGA hardware design for a particular case study. So, it is required to address the implementation details.

## 1.6 Thesis Objectives

This thesis is dedicated to investigate and resolve the aforementioned problems in real-time HIL simulation of electric machines and drives. Accordingly, the thesis objectives are listed as follows:

- The mathematical representation of the three widely used electric machine models, i.e. lumped q-d, MEC and FEM models, should be developed and the optimal FPGA hardware realization scheme should be proposed for real-time simulation applicable in HIL testing.
- In HIL platform, due to timing constraints, a compromise between the model accuracy and computational burden should be made. So, the real-time models should be compared in terms of model fidelity, minimum

achievable time-step, and FPGA hardware resource utilization. Furthermore, the ability of each model in considering physical phenomenon including nonlinearity of the iron core and accommodation of various types of internal machine faults should be evaluated in a wide range of operating conditions.

- Targeting to achieve the minimum time-step with a specified model, the potential for reducing the number of iterations, problem size reduction, increasing pre-calculations, and utilizing algorithms to facilitate parallelism on FPGA-based real-time simulator should be investigated.
- As the core of electric machine simulation with detailed MEC and FEM models, the solution of a different system of linear equations in every nonlinear iteration, due to nonlinear magnetic materials, and time-step, due to mechanical movement, is a bottleneck in achieving the minimum time-step. The application of TLM method in solution of the nonlinear systems to decouple nonlinear elements from the linear system shows its efficiency in reducing the execution time on conventional CPU processors. However, the efficiency of TLM method on FPGA hardware to highlight the parallelism properties of the method, mainly in solution of the decoupled nonlinear elements, should be evaluated.
- As the most time consuming part of the MEC and FEM solutions, the linear solver consists of a sparse LU decomposition followed by forward elimination and backward substitution. Considering the hardware architecture of FPGAs and the limitation of resources, efficient methods should be proposed for medium-sized sparse solvers to maximize parallelism to achieve the minimum time-step.
- The efficiency of the automatic VHDL program generator in comparison with the handwritten textual VHDL programming should be compared. Two implementations with different number representations, including 32-bit fixed-point and 32-bit single-precision floating-point numbers, should be compared to evaluate the effect of number representation



on accuracy and hardware resource. All the parallelism potentials at the algorithmic level and the operation level should be exploited and presented in detail to maximize FPGA parallelism. Two FPGA boards should be used for comparison in terms of the hardware resource and maximum drivable clock frequency.

- Two different type of electric machines in electrified transportation systems should be studied to show the effectiveness of the real-time simulation on different case studies. A low power 3 hp 4-pole squirrel-cage induction motor applicable in passenger electric vehicles and a high power 1000 hp 21-pole industrial sample of a single-sided short primary linear induction motor for magnetic levitation application are chosen for this purpose.

## 1.7 Thesis Outline

The six chapters of this thesis is organized based on the electric machine models for different case studies as follows:

- **Chapter 2: Real-Time Lumped q-d Model of Linear Induction Motor** - This chapter proposes the q-d-based real-time digital emulation of a high power single-sided linear induction motor (SLIM) of the American Maglev Technology (AMT) with the drive system on FPGA for transportation application. Implementation of the model is performed with both fixed-point numbers using the XSG toolbox and floating-point numbers using a handwritten VHDL program. The FPGA architecture is massively paralleled to obtain the minimum time-step size, scarifying the hardware resource due to the simplicity of the model.
- **Chapter 3: Real-Time Magnetic Equivalent Circuit Model of Faulted Rotary Induction Motor** - In this Chapter the MEC-based real-time emulation of faulted induction machines, facing the timing constraint challenges with high bandwidth and non-periodic boundary condition of fault studies are investigated. The application of the TLM

method to the MEC model is evaluated in order to keep the coefficient matrix unchanged within each time-step, requiring only one LU decomposition per time-step due to movement. The MEC-based TLM matrix is re-ordered efficiently to keep the majority of the matrix unchanged during the entire simulation to facilitate partial pre-calculation of the LU decomposition through the left-looking Gilbert-Peierls algorithm.

- **Chapter 4: Hardware Acceleration of Finite Element Method Model of Linear Induction Motor** - In this chapter the FEM-based hardware acceleration of the SLIM on the parallel reconfigurable hardware of FPGA is achieved. The nonlinearity of the iron core as well as the movement are taken into consideration. A new sparse solver is proposed based on the left-looking Gilbert-Peierls algorithm for the system of linear equations of FEM that need to be solved in every iteration and time-step. Implementation of the model is performed in a massively paralleled and deeply pipelined hardware architecture using VHDL programming with single-precision floating-point number representation.
- **Chapter 5: Real-Time Finite Element Method Model of Linear Induction Motor** - For the first time, the real-time problem of FEM computations of electric machines has been resolved by employing the TLM method, which keeps the stiffness matrix unchanged within each time-step and facilitates the proposed finite pre-calculation LU decomposition for the infinite number of time-steps. The proposed scheme offers the most accurate and fast platform for testing of electric machine drives interacting with other components in the HIL scenario.
- **Chapter 6: Conclusions and Future works** - The contributions of this thesis and suggestion for future works are presented in this chapter.

# Chapter 2

## Real-Time Lumped q-d Model of Linear Induction Motor

### 2.1 Introduction

Linear induction motors (LIMs) provide both propulsion force in horizontal as well as levitation force in vertical directions. As a distinct feature, LIMs levitation force is required in conveyance systems from small to large scale applications [105, 106]. Two major large scale applications include magnetically levitated (MagLev) vehicles [107–110] and electromagnetic launch systems (EMLS) [111, 112]. LIMs are widely used in transportation system especially in urban areas, due to many advantages including high speed and noiseless operation, as well as fast acceleration and deceleration as a result of controlling the levitation force.

An equivalent circuit model of LIM to formulate the machine behavior was first proposed in [113], which was derived from the equivalent circuit of rotary induction machine. Different approaches were proposed to derive the equivalent circuit parameters, including magnetic equations of the air gap [114], field analysis [115] and winding functions [116]. Based on the presented equivalent circuits, the two axis q-d model has been derived in [117, 118], which is a useful tool for control purposes. Other models presented for LIM include FEM in 2-D [119] and 3-D [120] and finite difference time domain (FDTD) method [121]. Some other researches were conducted for simulation and validation of LIM in [122–129]. A taxonomy of research performed on

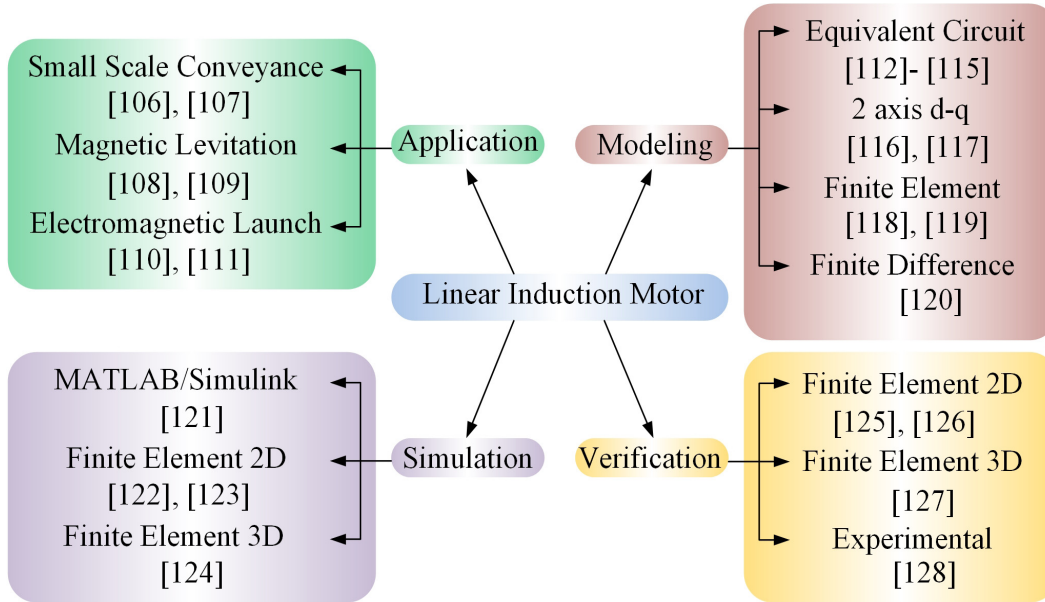


Figure 2.1: Taxonomy of literature on linear induction motor.

LIMs in different areas of application, modeling, simulation and validation is shown in Fig. 2.1.

In the process of prototyping an electric machine drive system, the design of the controllers require several cycles of testing, which have the drawbacks of expensive equipment, human resources, time consuming construction and testing, destructive to actual equipment, power consumption and physical space requirement. Conventionally, off-line simulations were used to evaluate the behavior of an electric machine and its drive, which suffer from long simulation time as well as inaccuracy due to simplification of the model. Furthermore, off-line simulations can not emulate the behavior of an electric component interacting with other equipment under realistic conditions.

HIL simulation is an efficient and cost effective approach to predict the behavior of a system prior to manufacturing and commissioning. HIL simulation allows testing newly designed simulated component interaction with external devices in a non-destructive environment, minimizing the error of the testing set up by removing inaccurate modeling of the interacted devices [57]. This scenario is efficient when the simulated component can exchange data in real-time. Real-time simulation is achieved when the system computations

take less time than a specified time-step. The time-step should be specified small enough in such a way that the required accuracy is satisfied.

There are two hardware implementation methods: the schematic block method of Xilinx System Generator (XSG) with fixed-point number representation, and the textual programming language using VHDL with floating-point number representation. Up to now, no contribution in the area of HIL simulation is available for LIM drive systems. This chapter focuses on real-time simulation of a complete LIM drive system for magnetic levitation application using both fixed-point and floating-point number representations on FPGA.

In this chapter, first the rotor field oriented control scheme is applied to LIM to facilitate decoupled control of the levitation and propulsion forces. Second, the derived machine model accompanied with the drive system model is used to obtain parallelized architecture for hardware representation using VHDL codes with both fixed-point and floating-point number representations. Then the models are implemented on the Xilinx<sup>®</sup> Virtex-7 (XC7VX485T-2FFG1761C) FPGA in order to emulate the machine behavior and control system characteristics in real-time. Finally, the real-time results are verified with experimental results as well as finite element simulation using JMAG<sup>®</sup> software and discussions are provided.

## **2.2 Hardware-in-the-Loop Emulation of LIM Drive**

### **2.2.1 Hardware Realization of LIM**

#### **a) Model Formulation**

The structure of LIM consists of the primary and secondary parts is shown in Fig. 2.2. The primary mover includes the iron core and a three-phase winding supplied by an external source to create the magnetic field. The secondary includes an aluminum sheet to conduct the induced current providing a secondary magnetic field and the back iron to provide a low reluctance flux path. The interaction between the two magnetic fields provides the propulsion and levitation forces in the horizontal and vertical directions, respectively.

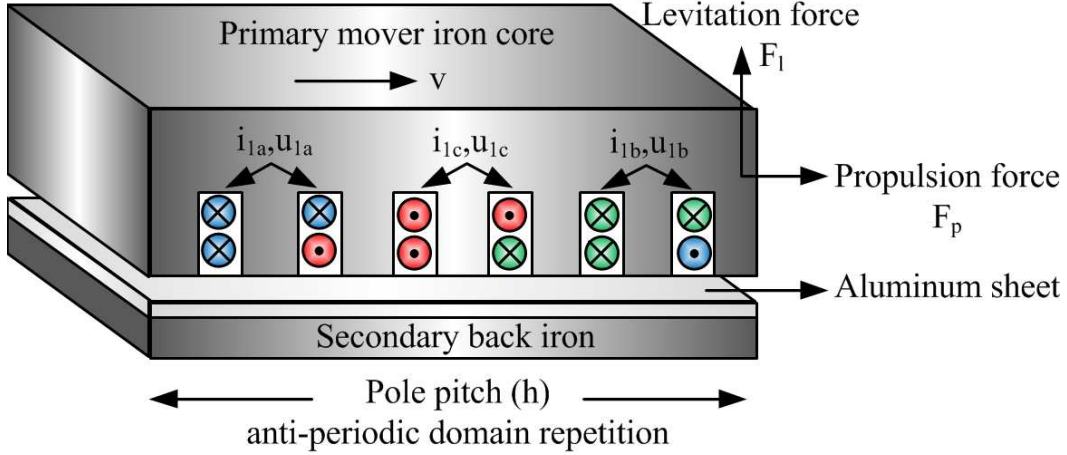


Figure 2.2: Structure of linear induction motor.

LIM model can be extracted from the conventional model of a rotary induction machine, considering the primary current and secondary flux as the state variables as follow [130]:

$$\begin{aligned} \frac{di_{1q}}{dt} = & -\frac{\pi}{h}v_e i_{1d} - \left( \frac{R_1}{\sigma L_1} + \frac{1-\sigma}{\sigma T_2} \right) i_{1q} - \\ & \frac{L_m \pi}{\sigma L_1 L_2 h} v \lambda_{2d} + \frac{L_m}{\sigma L_1 L_2 T_2} \lambda_{2q} + \frac{1}{\sigma L_1} u_{1q}, \end{aligned} \quad (2.1)$$

$$\begin{aligned} \frac{di_{1d}}{dt} = & -\left( \frac{R_1}{\sigma L_1} + \frac{1-\sigma}{\sigma T_2} \right) i_{1d} + \frac{\pi}{h} v_e i_{1q} + \\ & \frac{L_m}{\sigma L_1 L_2 T_2} \lambda_{2d} + \frac{L_m \pi}{\sigma L_1 L_2 h} v \lambda_{2q} + \frac{1}{\sigma L_1} u_{1d}, \end{aligned} \quad (2.2)$$

$$\frac{d\lambda_{2q}}{dt} = \frac{L_m}{T_2} i_{1q} - \left( \frac{\pi}{h} v_e - \frac{\pi}{h} v \right) \lambda_{2d} - \frac{1}{T_2} \lambda_{2q}, \quad (2.3)$$

$$\frac{d\lambda_{2d}}{dt} = \frac{L_m}{T_2} i_{1d} - \frac{1}{T_2} \lambda_{2d} + \left( \frac{\pi}{h} v_e - \frac{\pi}{h} v \right) \lambda_{2q}, \quad (2.4)$$

where  $u_1$ ,  $i_1$ ,  $R_1$  and  $L_1$  are the voltage, current, per phase resistance and self inductance of the primary mover respectively,  $L_m$  is the mutual inductance between the primary and secondary,  $L_2$  and  $R_2$  are the secondary inductance and resistance,  $T_2 = \frac{L_2}{R_2}$  is the secondary time constant and  $\sigma = 1 - \left( \frac{L_m^2}{L_1 L_2} \right)$  is the leakage factor,  $\lambda_{2d}$  and  $\lambda_{2q}$  are the secondary flux in  $d$  and  $q$  axis,  $h$  is the pole pitch,  $v_e$  is the synchronous linear speed and  $v$  is the linear speed of the primary mover.

The equation of motion for the LIM is as follows:

$$F_p = M\dot{v} + Dv + F_L, \quad (2.5)$$

where  $F_p$  and  $F_L$  are the LIM propulsion and load forces, respectively.  $M$  is the mover mass and  $D$  is the viscous friction and iron loss factor.

The propulsion and levitation force equations can be presented in terms of state variables, i.e. q-d primary currents and secondary fluxes as follows:

$$F_p = \frac{3\pi L_m}{2hL_2} (\lambda_{2d}i_{1q} - \lambda_{2q}i_{1d}), \quad (2.6)$$

$$F_l = \frac{3L_m}{4gL_2} (\lambda_{2d}i_{1d} + \lambda_{2q}i_{1q}), \quad (2.7)$$

where  $F_l$  is the levitation force and  $g$  is the air gap.

The state-space system of equations of (2.1)-(2.4) discretized using Forward Euler can be presented as follows:

$$\begin{cases} \mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t[\mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)] \\ \mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \end{cases}, \quad (2.8)$$

where  $\mathbf{x} \in \mathbb{R}^{4 \times 1}$  is the state variable vector, i.e.  $\mathbf{x}^T = [i_{1q}, i_{1d}, \lambda_{2q}, \lambda_{2d}]$ ,  $\mathbf{u} \in \mathbb{R}^{2 \times 1}$  is the input primary mover voltage vector, and  $\Delta t$  is the time-step.  $\mathbf{y}(t)$  is the output vector, i.e. the propulsion and levitation forces.  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  and  $\mathbf{D}$  are the coefficient matrices depending on machine parameters as presented in (2.1)-(2.7).

## b) Hardware Emulation

In hardware design, compromising between hardware resource utilization, the simulation speed, and the accuracy is challenging and depends on the case study. Massively parallel implementation of algebraic calculations (single data per logic resource per time-step) implies more resource utilization with the advantage of reaching less time-step and simplicity. However, fully pipelined implementation (multiple data per logic resource per time-step) means use an IP core several times in each time-step, which implies utilizing less hardware resources while increasing complexity as well as having larger time-step. In most

implementations for optimal hardware design both parallel and pipelined architectures should be used. For the case of vector control of electric machines, due to small amount of calculations, the hardware resource utilization might not be a problem. So, parallelism is more suitable to achieve minimum latency.

The details of both electrical and mechanical machine model implementation based on (2.1)-(2.8) using floating-point IP cores is shown in Fig. 2.3 in a massively parallel architecture by performing the independent multiplications and additions in parallel. The blocks can be also implemented using fixed-point schematic block of XSG in the desired architecture. The latency of 7 and 5 clocks is considered for additions and multiplications respectively and the LIM hardware emulation is reached within 47 clocks latency. The state variables of the next time-step are written to 32-bit RAMS with the depth of 16 in order to be used as the input of the next time-step.

## 2.2.2 Hardware Realization of the Drive System

### a) Model Formulation

Decoupled control of the propulsion and levitation forces of LIMs can be extracted from decoupled control of rotor flux and torque of conventional rotary induction machines, respectively.

Ideally, for decoupled control, the secondary flux is controlled to be aligned with the  $d$ -axis, i.e.:

$$\lambda_{2q} = 0. \quad (2.9)$$

As a result, the propulsion and levitation forces can be controlled independently through  $q$  and  $d$  components of the primary current, respectively. Fig. 2.4 shows the rotor field oriented control schematic.

Conventional PI controllers are used for primary speed and secondary flux control through  $i_{1qref}$  and  $i_{1dref}$ , respectively. The discretized speed controller based on first order Forward Euler method can be obtained as follows:

$$\begin{aligned} i_{1qref}(t + \Delta t) = & i_{1qref}(t) + k_p[2v_{err}(t) - \\ & v_{err}(t - \Delta t)] + [-k_p + k_i\Delta t]v_{err}(t), \end{aligned} \quad (2.10)$$



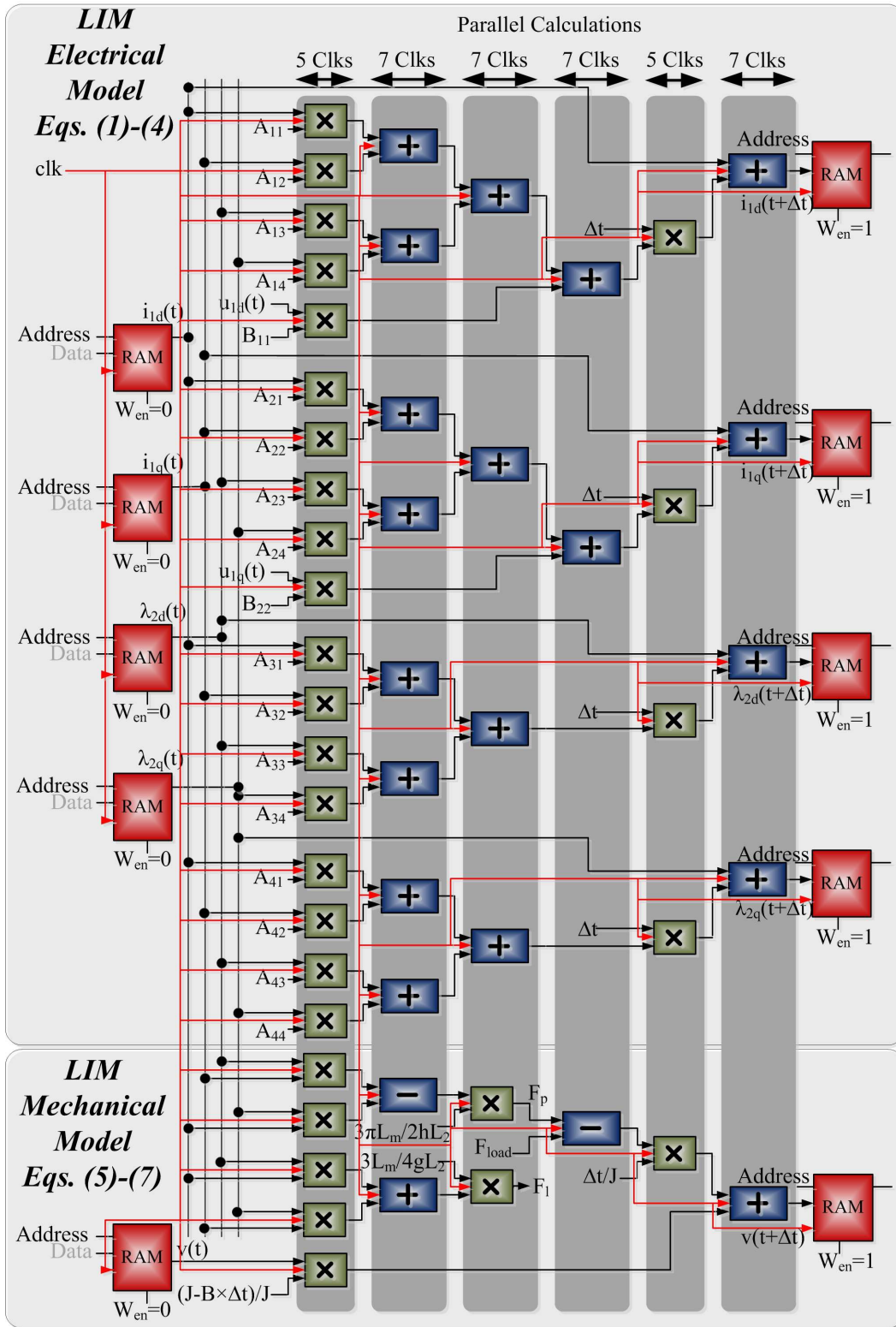


Figure 2.3: Massively parallel state-space hardware implementation of LIM model.

where

$$v_{err}(t) = v_{ref}(t) - v(t). \quad (2.11)$$

Similarly, for the flux control unit:

$$\begin{aligned} i_{1dref}(t + \Delta t) &= i_{1dref}(t) + k_{pf}[2\lambda_{2derr}(t) - \\ &\lambda_{2derr}(t - \Delta t)] + [-k_{pf} + k_{if}\Delta t]\lambda_{2derr}(t), \end{aligned} \quad (2.12)$$

where

$$\lambda_{2derr}(t) = \lambda_{2dref}(t) - \lambda_{2d}(t). \quad (2.13)$$

The next time-step position can be calculated as follows:

$$\beta(t + \Delta t) = \beta(t) + \frac{\pi}{h}v_e(t)\Delta t, \quad (2.14)$$

where

$$v_e(t) = v(t) + \frac{h}{\pi} \frac{i_{1q}(t)L_m R_2}{int(t)L_2}, \quad (2.15)$$

in which  $int$  is an intermediate signal that can be calculated as follows:

$$int(t) = \frac{L_2 - R_2 dt}{L_2} int(t - \Delta t) + \frac{L_m R_2 dt}{L_2} i_{1d}(t - \Delta t). \quad (2.16)$$

The switching model of an inverter is used to emulate a 2-level three-phase converter as a voltage source supplying the LIM.

## b) Hardware Emulation

The details of hardware architecture of the control system units including PI and hysteresis controllers, switching model of the power converter, speed and position calculations as well as sine and cosine functions for q-d to 3 phase conversion are shown in Fig. 2.5 with synchronous clock signal. The design is fully paralleled in order to decrease the time-step. The sine and cosine functions required for electric machine modeling are generated by Look-Up Table (LUT). To do so, the sine and cosine values corresponding to each angle need to be stored in RAMs. The accuracy of the sine and cosine generators using this method depends on the depth of the RAMs, which is 4096 for  $2\pi$  in this study. In order to use the angle as the input address of the RAM, the periodicity of sine and cosine functions is used to find the corresponding

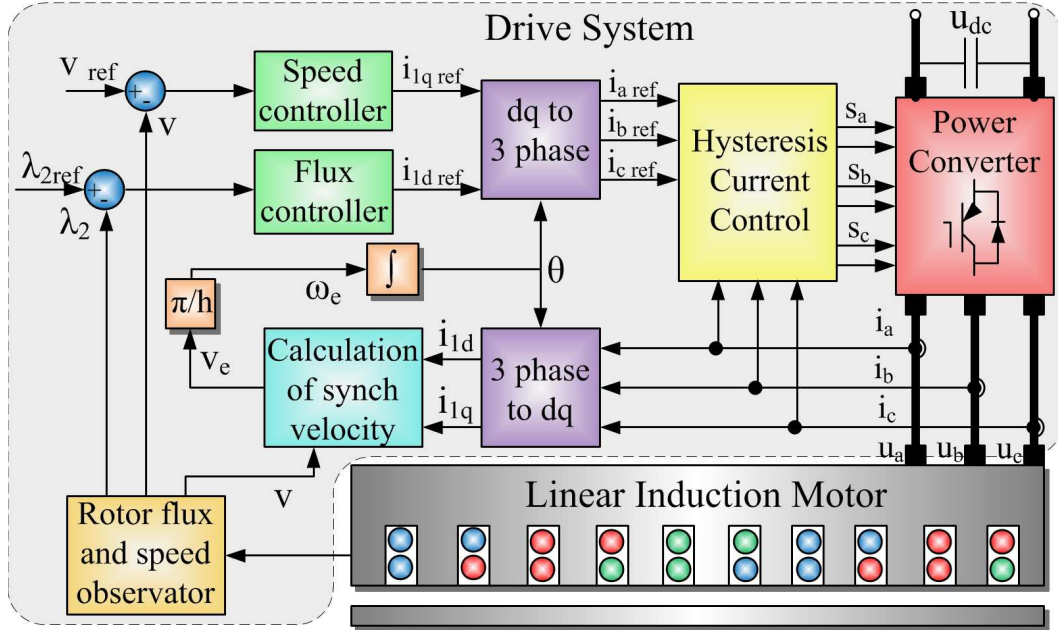


Figure 2.4: LIM rotor field oriented control scheme.

angle between  $[0, 2\pi]$ . It can be achieved by adding the corresponding angle of machine primary mover position increment to the previous time-step angle between  $[0, 2\pi]$ . Then the angle was mapped to the address of the RAM by the gain of  $depth/2\pi$ , as shown in Fig. 2.5, where the width of each block corresponding to the latency of each IP core is shown.

### 2.2.3 FPGA Implementation

Two types of implementation exist depending on the number representation using schematic blocks of XSG with fixed-point and VHDL textual programming language with floating-point number representation.

For fixed-point evaluation of the drive system, XSG can be used as a tool for implementation of the system in a relatively simple manner in comparison with HDL coding. XSG provides schematic blocks called Visual Block Programming Language (VBPL) for different operations in MATLAB/Simulink<sup>®</sup> environment. Then, the VHDL code based on fixed-point number representation can be generated and synthesized using ISE<sup>®</sup> or Vivado<sup>®</sup> softwares to be implemented on FPGA through the JTAG cable for real-time emulation.

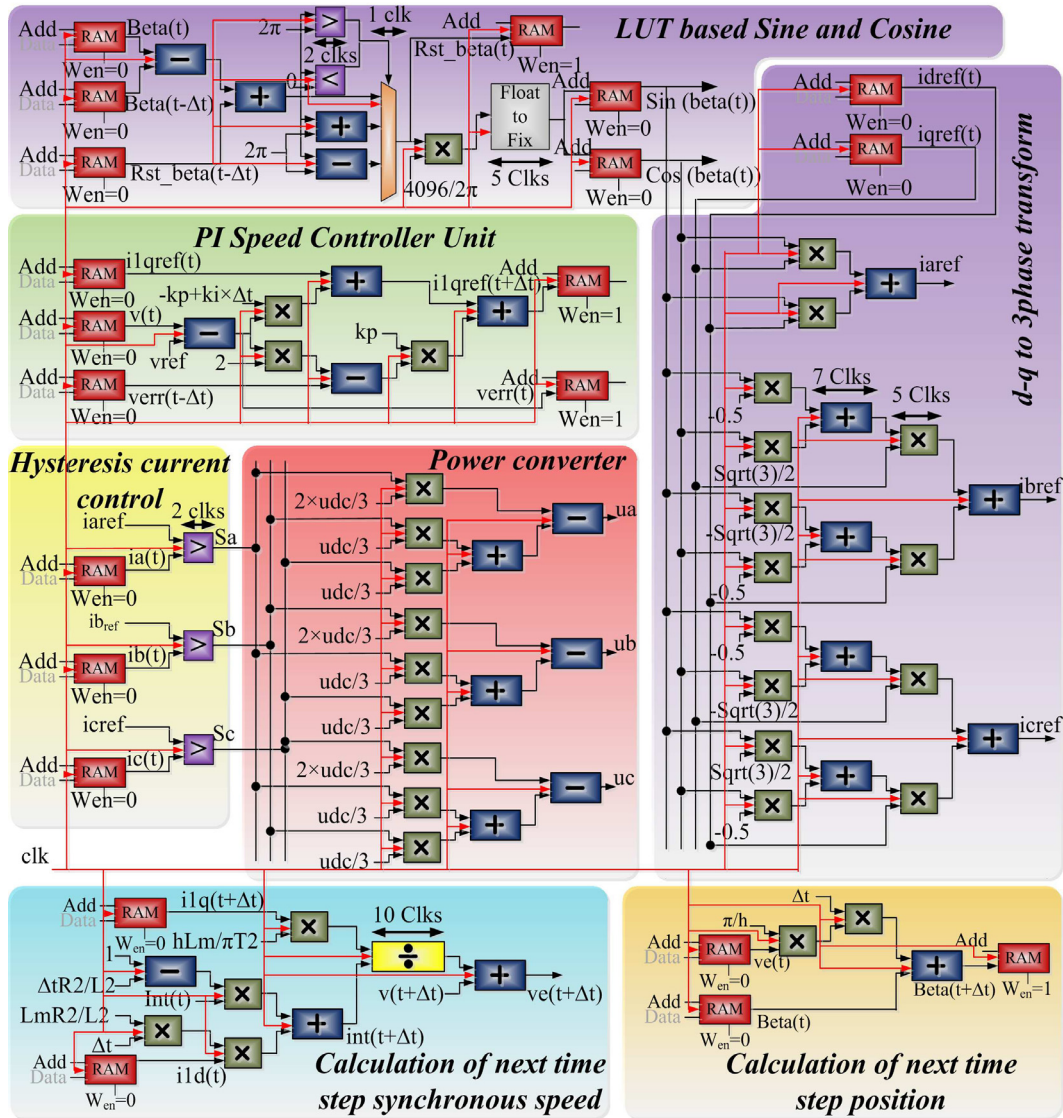


Figure 2.5: Massively parallel hardware implementation of rotor field oriented control units.

In the fixed-point number representation a specific number of bits should be allocated for integer and binary part of the number. In order to capture the best accuracy with a specified number of bits, the information about the value of each signal should be known, which is a disadvantage of the fixed-point number representation.

With allocation of the same number of bits, floating-point number representation format allows covering a wider range of numbers, which cause having more accurate results without requiring any information about the value of the data. According to IEEE standard 754, numbers have two floating-point formats, i.e. single-precision and double-precision formats with 32 and 64-bits wide, respectively.

The entire system signals for data path is shown in Fig. 5. State 0 to State 5 form the algorithm to calculate all signals for a LIM drive system in each time-step. In the design architecture, a massively parallel approach for algebraic calculations is used in order to achieve the minimum time-step. Most calculations that are independent can be performed in parallel at the beginning of each time-step (State 0) using the values calculated in the previous time-step, when Forward Euler method for discretization is utilized.

The process begins in State 0 with calculation of the sine and cosine of the corresponding rotary angle of LIM primary mover position, required for q-d to 3 phase conversion matrix in State 1, in parallel with calculation of the reference q-d current of the next time-step as well as the propulsion and levitation forces of the current time-step. Using the mechanical equation (2.5), with the calculation of propulsion force, the next time-step speed is calculated in State 1. In State 1, Park's transformation is implemented in order to calculate three-phase reference currents from the reference q-d currents. With comparison of the three-phase reference currents with measured currents, the switching signals using hysteresis current control can be achieved, which results in determination of three-phase voltages in State 2. State 3 represents the LIM equations in state-space in order to calculate state variables of the next time-step using the state of the system at current time-step. Then, the measured three-phase current as well as the electrical speed of the next time-step is

calculated in State 4. At the end of each time-step (State 5), RAMs should be updated to be used as the input for the next time-step.

## 2.3 Results and Discussion

### 2.3.1 Real-Time Simulation Results

In the real-time closed loop LIM drive system the machine speed, secondary flux and the propulsion load references are considered as the inputs. The machine current is controlled in a way that the frequency depends on the machine speed and the amplitude depends on the propulsion and levitation forces. A practical sequence of references according to time intervals based on a real transportation system are applied to emulate the LIM behavior as follows:

- 0-1 sec: For departure, at stand still, the secondary flux is ramped up to 0.5 Wb to provide the levitation force, which creates a fixed air gap for contactless operation.
- 1-2 sec: With the constant flux, the machine is accelerated and the speed and propulsion force are ramped up to 20 m/s and 17500 N, respectively.
- 2.8-3.2 sec: A disturbance in load force is considered as the result of wind speed or rail curve.
- 4-5 sec: The machine is decelerated and the speed is ramped down to stop in the station as a result of reduction in propulsion force and viscous friction.
- 5-6 sec: In the station, at stand still, more passengers are getting on and additional flux build up is made to keep the air gap fixed for appropriate contactless operation.
- 6-7 sec: The speed and an additional amount of propulsion force is ramped up to 20 m/s and 20000 N due to the additional passengers.



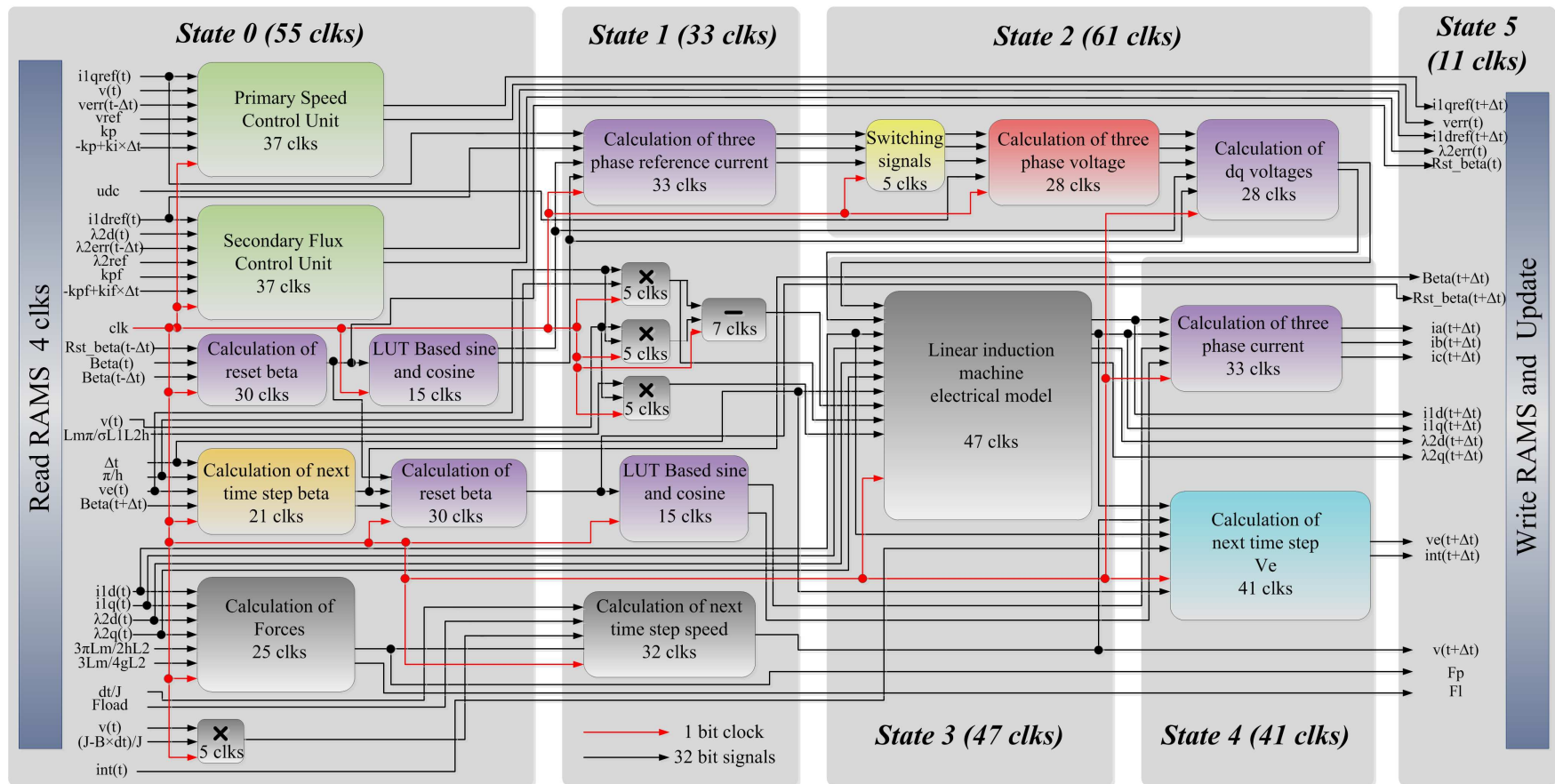


Figure 2.6: Signal integration in the hardware emulation architecture for the LIM drive system (modules scaled by latency).

- 8-8.3 sec: Emergency electromagnetic braking for an unexpected event is applied by negative propulsion force that results in sharp speed ramp down.

The real-time oscilloscope results are presented in Fig. 2.7.

## 2.3.2 Floating-Point vs. Fixed-Point Comparison

### a) Accuracy Evaluation

In the LIM control system, online control of secondary flux is performed by estimation of the reference secondary flux to adjust the levitation force in order to keep the air gap fixed. So, accurate calculation of secondary flux is crucial. Fig. 2.8 shows the accuracy of flux calculation by means of both fixed-point and floating-point number representations, when a 0.5 Wb step change on secondary flux is applied. The minimum achievable time-step for each design is utilized and Forward Euler method is used as the discretization method. In order to have a fair comparison, 32-bit single precision floating-point format (IEEE Std 754) and 32-bit fixed-point format have been chosen for number representations. The binary point in the fixed-point is located to achieve the maximum accuracy, e.g. for small values more number of bits are allocated for the binary part and for large values more number of bits are allocated for the integer part. Considering the Matlab m-file double precision number calculation as the reference, it can be seen that calculation of secondary flux with floating-point number representation using VHDL code is more accurate than the fixed-point using XSG, since with the same number of bits the floating-point format can represent a wider range of numbers.

### b) Timing Analysis

The time-step of a hardware design is based on the signal path through the circuit from input to output with longest delay called the critical path of the system, which generally contains the largest number of logic gates. The real-time emulator time-step size is important since currently available FPGAs based HIL setups are not able to accommodate hardware design time-steps



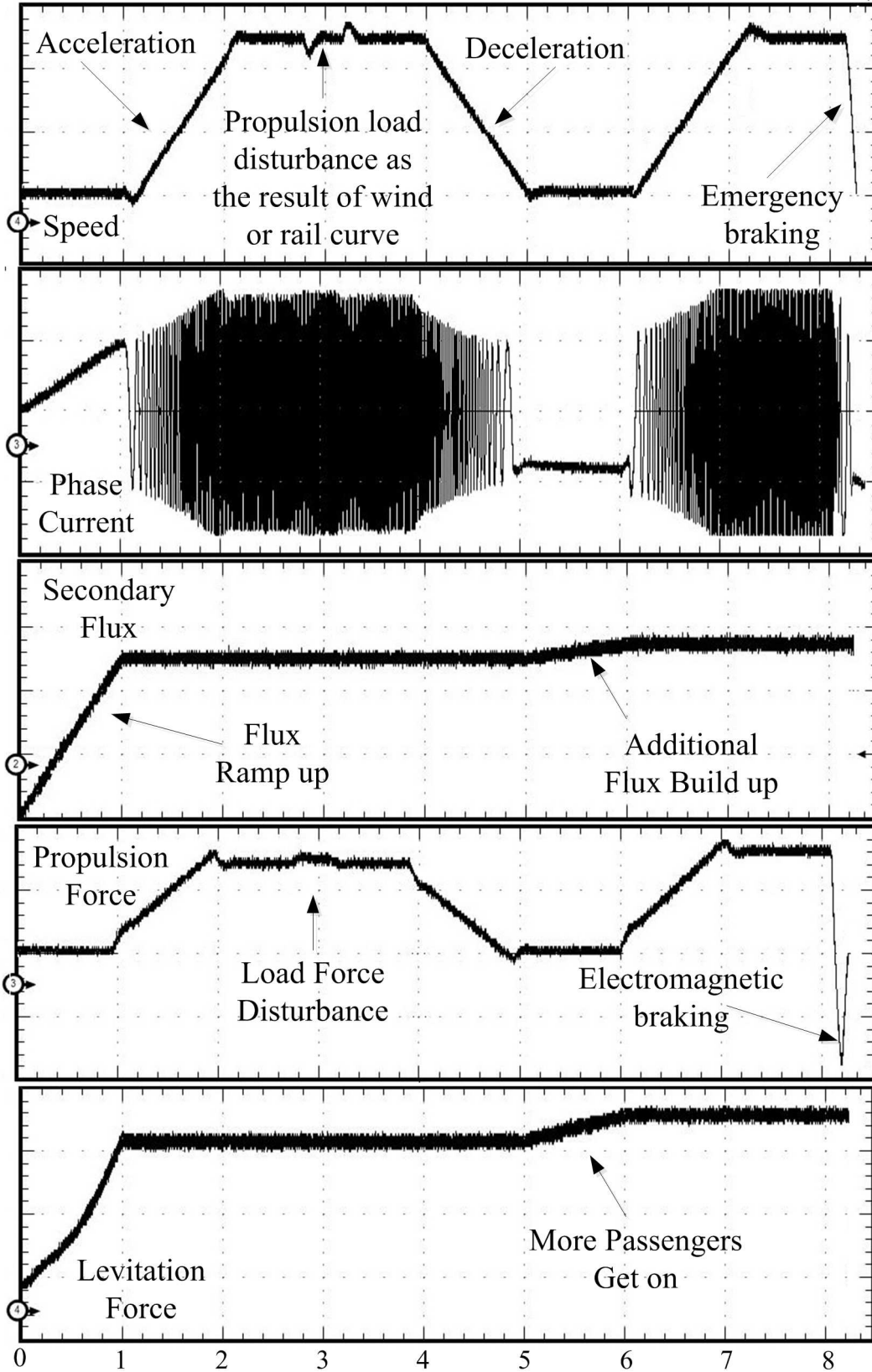


Figure 2.7: Floating-point real-time results of LIM [time: 1 sec/div, speed: 8 m/s/div, current: 1440 A/div, secondary flux: 0.2 Wb/div, propulsion force: 12000 N/div, levitation force: 16000 N/div].

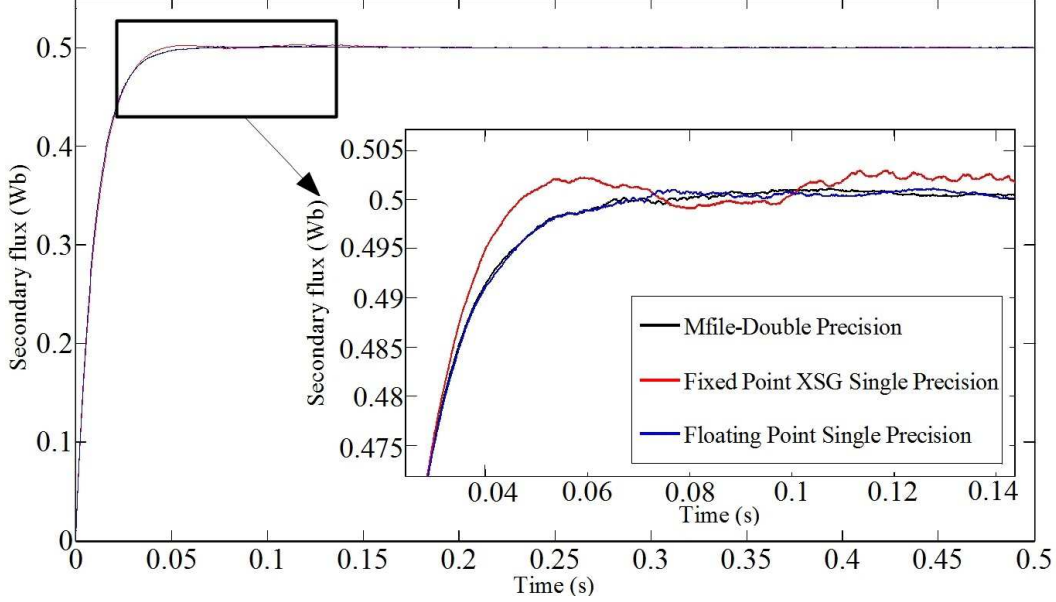


Figure 2.8: Fixed-point and floating-point accuracy comparison.

more than a few microseconds due to practical limitations on data transfer between the set up and actual external devices.

Assigning appropriate delay for each algebraic calculation is arbitrary, while at the time of implementation the FPGA clock would be affected. Therefore, 7 clock delay is considered for addition or subtraction and 5 clock delay for multiplication. With the specified delays and hardware implementation architecture, the floating-point implementation has 248 clock latency with the FPGA clock of 9.61 ns which resulted in a time-step size of  $2.3\mu\text{s}$ , while the fixed-point implementation has 11 clock latency with the FPGA clock of 74.19 ns which resulted in a time-step size of 816 ns. Thus, with fixed-point implementation a smaller time-step size can be achieved.

### c) Hardware Resource Utilization

The designed architectures are implemented on Xilinx<sup>®</sup> Virtex-7 XC7VX485T FPGA with available 607200 slice registers, 303600 slice LUTs, 75900 slices, 130800 memory units, 2800 DSP48 and 700 bonded IOBs. The hardware resource utilization in terms of the number of logic resources and the occupation percentage with both fixed-point using XSG and floating-point using VHDL

code is presented in Table 4.3. It can be seen that the floating-point approach requires more hardware resources, since the algorithms of floating-point computations require more hardware with the advantage of having more accuracy in comparison with fixed-point computations.

Table 2.1: Hardware Resource Utilization

Resources	Implementation Method	
	Floating-Point	Fixed-Point
Slice Registers	89,114 (14%)	52,632 (8%)
Slice LUTs	101688 (33%)	33,842 (11%)
Occupied Slices	26,452 (34%)	12,758 (16%)
Memory	13,483 (10%)	0 (0%)
DSP48	0 (0%)	168 (6%)
Bonded IOBs	83 (11%)	83 (11%)

### 2.3.3 Result Verification

#### a) Finite Element Simulation

The real-time emulation results were validated by FEM simulation of the full structure LIM through JMAG-Designer<sup>®</sup> software on a PC with Intel(R) Core(TM) i7-2600 CPU @3.4 GHz. The simulation time for 0.16 sec with the time-step of 1.25 msec with 194743 of elements, 107685 nodes is performed in 8 hours and 25 minutes, utilizing Newton-Raphson method as the nonlinear solver and incomplete Cholesky-conjugate gradient method (ICCG) for solving the system of linear equation. With supplying the primary mover, the flux passes through the secondary aluminum sheet and induces eddy currents. The steady state flux and eddy current distributions in the LIM is shown in Fig. 2.9.

#### b) Experimental Setup

The case study used is an industrial sample of a single sided short primary LIM with the length of 3.33 m, width of 26.67 cm, weight of 1136 kg constructed at the Center of Electromechanics, University of Texas at Austin, where the experimental results were obtained. The experimental setup of the linear induction machine is shown in Fig. 2.10.

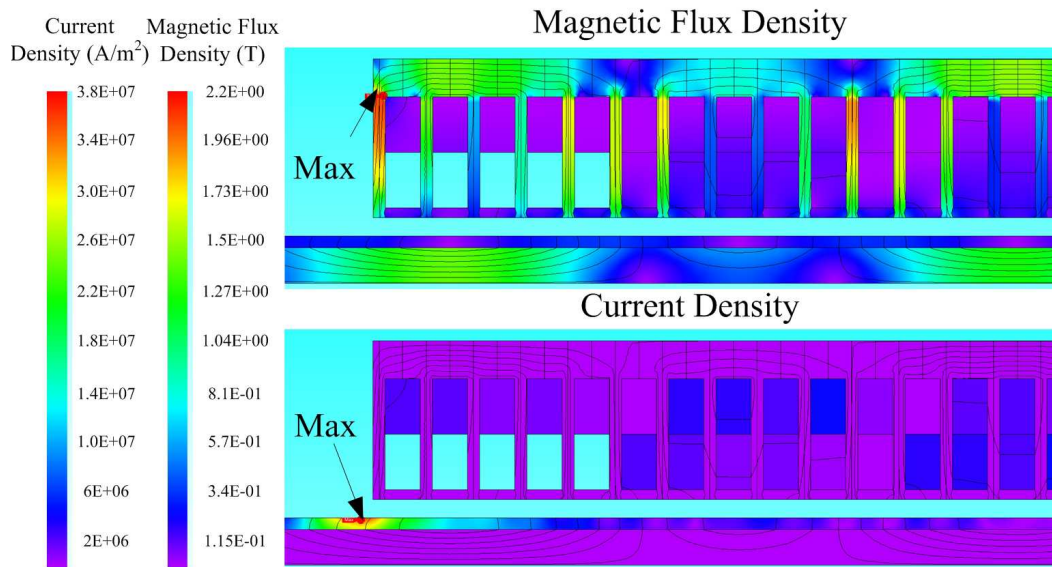


Figure 2.9: Finite element result of flux and current distribution in JMAG<sup>®</sup>.

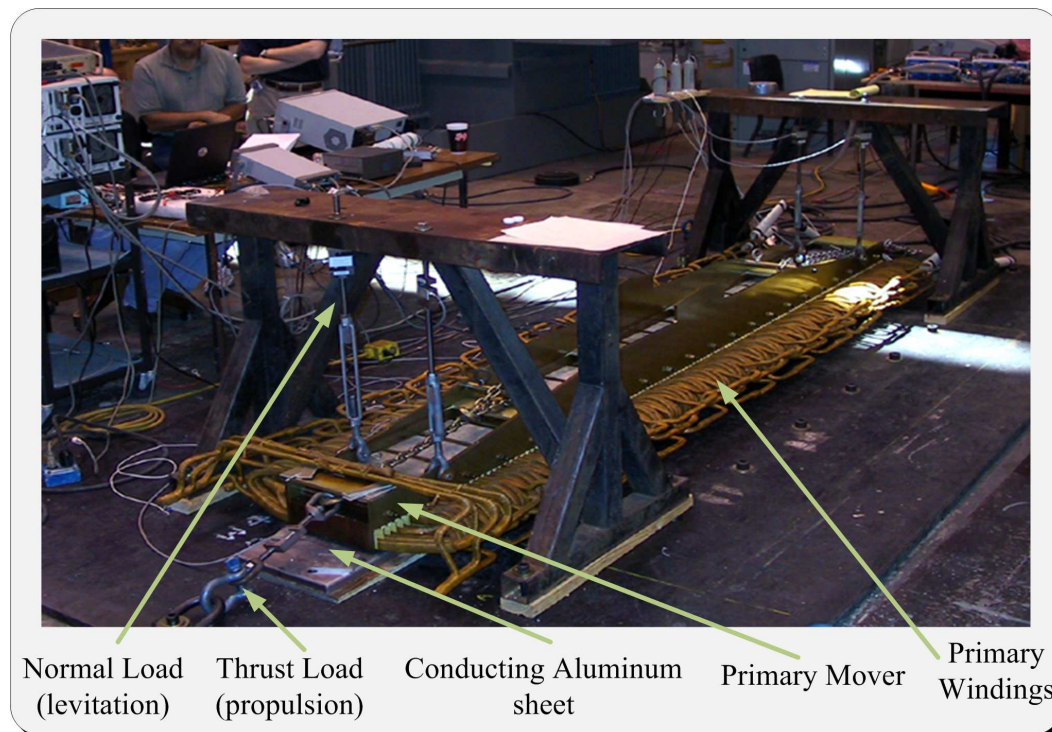


Figure 2.10: LIM experimental set up, courtesy of American Maglev Technology and Prof. K. Davey, used in [131] at Center of Electromechanics at the University of Texas at Austin.

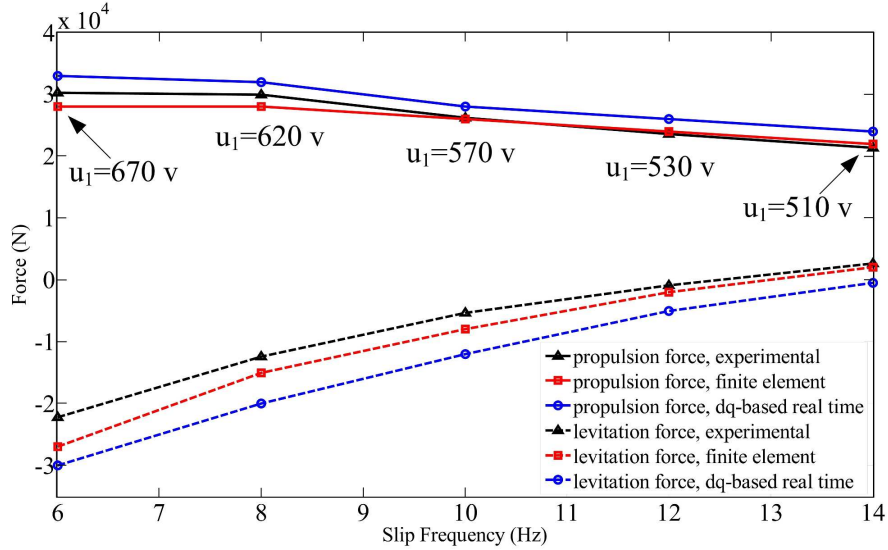


Figure 2.11: Torque vs. machine speed with experimental, real-time and finite element analysis.

Fig. 2.11 shows the propulsion and levitation forces versus speed change, indexed with slip frequency, with three different methods, i.e. experimental measurements, 2D FEM and the real-time emulation with the speed of 26.82 m/s and the primary current of 1543 A. The difference between the FEM simulation and the experimental result comes from inaccurate measurement of the propulsion and levitation forces due to practical circumstances. However, the difference between the real-time emulation and the FEM arises from the model used. The real-time simulation is based on the lumped q-d model, which does not consider the distributed nature, the nonlinearity of the iron core, as well as the end effects. Considering these reasons, the results are in acceptable agreement.

## 2.4 Summary

Linear induction machines are widely used in transportation systems due to their many advantages. Design and prototyping of electric machines is an expensive and time consuming process; HIL simulation provides an efficient alternative. In this chapter, FPGA-based real-time digital emulation of SLIM

with the drive system is proposed. Implementation of the model is performed in both fixed-point using XSG and floating-point number representations using handwritten VHDL code. Then an evaluation in terms of real-time step-size and accuracy as well as hardware resource utilization is provided. The whole design was fully paralleled, which resulted in a considerable reduction of model execution time. The minimum time-step of  $2.3 \mu s$  and  $0.8 \mu s$  was achieved for floating-point and fixed-point implementations, respectively. The results of the real-time simulation are verified by experimental results as well as a 2-D finite element simulation in JMAG<sup>®</sup> software.



# Chapter 3

## Real-Time Magnetic Equivalent Circuit Model of Faulted Rotary Induction Motor

### 3.1 Introduction

FPGA-based HIL emulation provides an accurate testing platform by real-time data transfer of the actual device under test, including control system prototypes and protection devices, interfaced with the interacted systems, involving electric machines, on real-time simulator [132–135]. Since the HIL platform is safe and non-destructive, the efficiency of the scenario is more highlighted when electric machine faults are studied on the real-time simulator. Although FEM model offers detailed and accurate modeling of electric machines, the computational intensity makes it inappropriate for real-time simulation [136]. The ability of MEC to consider magnetic nonlinearity of the iron core, spatial harmonics and machine faults with medium computational burden make it promising for real-time simulation [137–144].

The solution of MEC system of nonlinear equations requires LU decomposition of the entire coefficient matrix in each Newton-Raphson (N-R) iteration and time-step due to the nonlinearity of iron core and movement, respectively. Furthermore, fault studies require high bandwidth simulation of a non-periodic boundary problem, which challenges the timing constraints of the real-time simulation. The TLM method decouples nonlinear elements from a coupled linear network using lossless time-delayed transmission lines to keep the coef-

ficient matrix unchanged during nonlinear iterations [145, 146].

Accelerated and real-time simulation of one pole pitch of an induction motor using MEC model is performed in [101] and [102] respectively, where a look-up table (LUT) based TLM method on multi-core CPU is proposed. In [103], real-time execution of MEC model is performed on FPGA for the case of a switched reluctance motor (SRM) due to its double salient structure, which makes the MEC size much smaller than an induction motor. In [104], the MEC model of one pole pitch induction motor is used for real-time simulation on FPGA using a massively paralleled Gauss-Jordan elimination.

In this chapter, for the first time, real-time HIL emulation of faulted electric machines on FPGA is proposed, which further struggles with larger problem size of the entire study domain due to asymmetric properties of faults, and smaller time-step requirement to capture higher order of harmonics in the stator winding current under fault condition. To overcome the computational intensity, the TLM method is utilized as the base to keep the MEC coefficient matrix unchanged during the nonlinear iterations, and two novel ideas are proposed for the real-time TLM (RT-TLM) method as follows:

- An special combination of matrix re-ordering and left-looking Gilbert-Peierls algorithm for sparse LU decomposition to keep the majority of the decomposed matrix unchanged through the entire simulation, which minimizes the real-time simulation computational burden,
- FPGA hardware implementation is used to fully exploit the parallelism of the TLM algorithm and the reduced-order sparse solver.

The chapter is organized to first present the MEC-based TLM formulation and the proposed RT-TLM ideas in Section II. The massively parallel and deeply pipelined hardware design architecture and implementation on Xilinx<sup>®</sup> Virtex UltraScale+ XCVU9P FPGA board are discussed in Section III. Then, the real-time simulation results of the transient start-up and steady-state under healthy and various kind of faulty conditions are presented, including winding inter-turn faults, broken rotor bar, broken end ring, as well as power system disturbances including voltage unbalance and power supply harmonics



followed by 2-D FEM results validation in Section IV. Finally, in Section V the conclusions are provided.

## 3.2 Novel MEC-Based RT-TLM Emulation

### 3.2.1 Transmission Line Modeling (TLM) Method

TLM is a discretization method in time-domain, which proposes decoupling of nonlinear and reactive elements of an electrical circuit through lossless and time-delayed transmission lines with arbitrarily chosen characteristic impedance. Based on the TLM method, a nonlinear network is made equivalent to a linear network and decoupled nonlinear and reactive elements. The TLM procedure is based on traveling incident waves through the transmission lines and calculating the reflected waves at the linear network (sending end) and the decoupled elements (receiving end) due to mismatch of the traveling wave characteristic impedance. The calculation of the reflected waves at the sending end requires solution of a linear system of equations corresponding to the linear network including transmission lines. At the receiving end, the calculation of the reflected waves requires solution of decoupled individual nonlinear elements, where parallelism can be exploited. In comparison with the conventional method, where a linear system of equations needs to be solved in every nonlinear iteration and time-step, the TLM method requires only one LU decomposition per time-step followed by a number of forward elimination and backward substitutions equal to the number of TLM iterations.

### 3.2.2 TLM-Based MEC Modeling

Assuming an induction machine with  $N_{ss}$  number of stator slots and  $N_{rs}$  number of rotor slots, the  $i^{th}$  stator tooth and the  $j^{th}$  rotor tooth followed by their MEC models are shown in Fig. 3.1 (a) and (b), respectively. Due to the dependence of  $\mu$  on magnetic flux density, the permeances of stator base ( $P_{sb}$ ), stator tooth ( $P_{st}$ ), rotor tooth-tooth ( $P_{rtrt}$ ), rotor tooth ( $P_{rt}$ ) and rotor base ( $P_{rb}$ ) are nonlinear. However, the permeances of stator-tooth-to-rotor-tooth ( $P_{strt}$ ) and the stator tooth-tooth ( $P_{stst}$ ) are considered linear due to the dominance

of constant permeability of air ( $\mu_0$ ).

Conventionally, MEC model requires the solution of a system of equations in every iteration and time-step due to the nonlinearity of permeances and movement, respectively. The TLM method uses lossless time-delayed transmission lines to decouple nonlinear permeances from a coupled linear network as shown in Fig. 3.1 (c) to keep the MEC matrix unchanged within each time-step. The corresponding transmission line characteristic admittances ( $Y_{0P}$ ) can be determined by substituting the nonlinear permeability of the permeances ( $\mu$ ) with arbitrarily chosen permeability ( $\mu_{tlm}$ ) as follows:

$$P = \left( \int_0^L \frac{dl}{\mu S(l)} \right)^{-1}, \quad (3.1)$$

$$Y_{0P} = \left( \int_0^L \frac{dl}{\mu_{tlm} S(l)} \right)^{-1}, \quad (3.2)$$

where  $L$  and  $S$  are the length and cross-section area of the element. The nonlinear permeances are substituted by their Thevenin equivalent circuits of the transmission lines shown in Fig. 3.1 (d) [101].

The TLM iterative procedure begins with traveling incident potentials through transmission lines and calculation of the reflected potentials from the linear network (sending end) using the nodal magnetic vector potentials. Considering the  $k^{th}$  TLM iteration of the  $n^{th}$  simulation time-step, the magnetic vector potentials ( ${}^n_k \mathbf{A}$ ) can be found by the solution of MEC-based TLM linear system of equations.

In Fig. 1 (d), the Kirchhoff's current law (KCL) for the stator base and tooth nodes are respectively as follows:

$$\begin{aligned} & Y_{0P_{st,i}} ({}^n_k A_{sb,i} - {}^n_k A_{st,i} - 2^n A_{P_{st,i}}^{inc.}) + Y_{0P_{sb,i-1}} ({}^n_k A_{sb,i} \\ & - {}^n_k A_{sb,i-1} - \mathbf{N}_{qd0,i-1}^T {}^n \mathbf{i}_{qd0} - 2^n A_{P_{sb,i-1}}^{inc.}) + Y_{0P_{sb,i}} \\ & ({}^n_k A_{sb,i} - {}^n_k A_{sb,i+1} + \mathbf{N}_{qd0,i}^T {}^n \mathbf{i}_{qd0} + 2^n A_{P_{sb,i}}^{inc.}) = 0, \end{aligned} \quad (3.3)$$

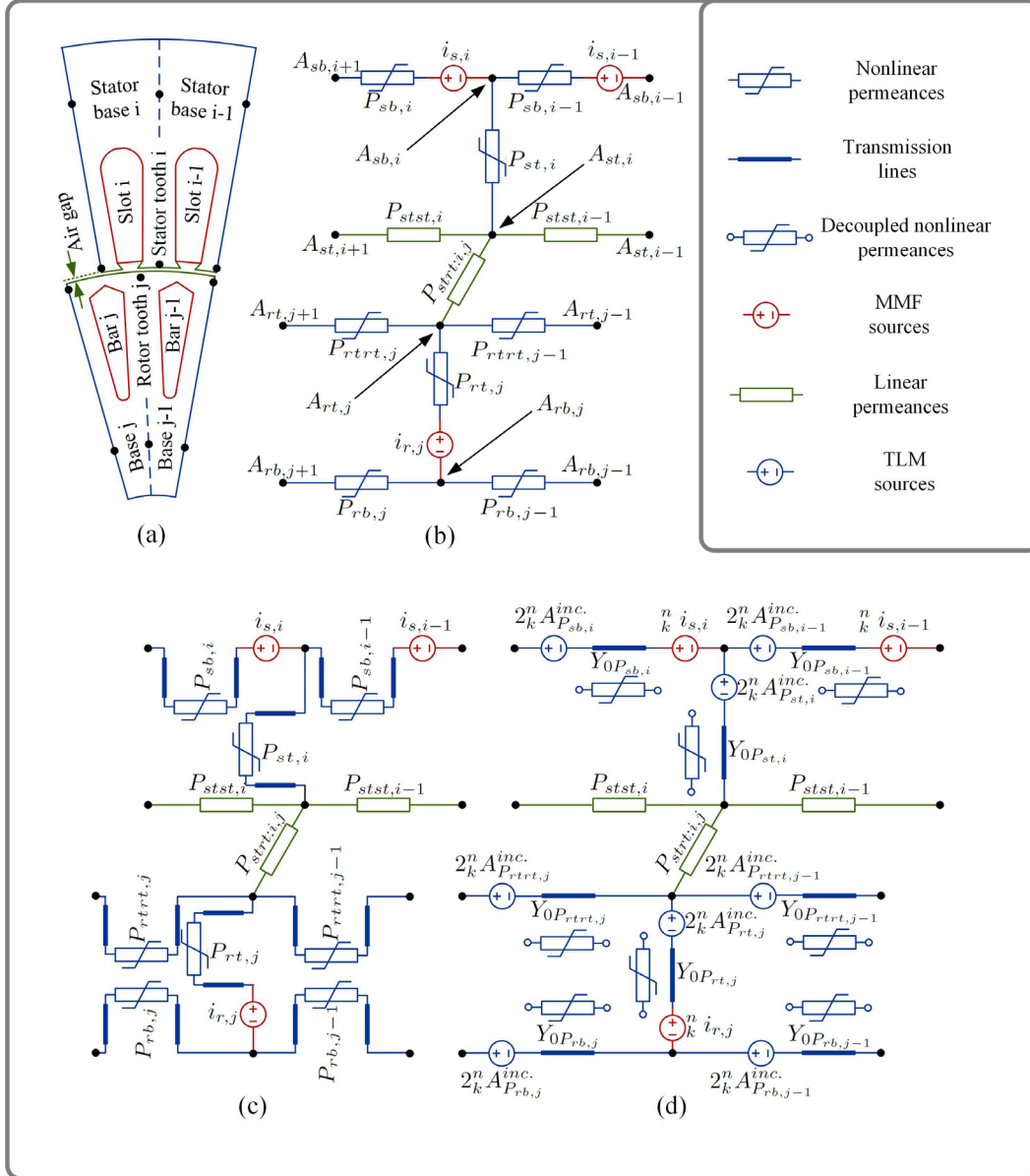


Figure 3.1: The TLM application procedure on MEC. (a) One tooth of the stator and rotor of an induction motor, (b) MEC model, (c) Transmission line decoupling, (d) MEC-based TLM network and the decoupled nonlinear permeances.

$$\begin{aligned}
& Y_{0P_{st,i}} ({}^n A_{st,i} - {}^n A_{sb,i} + 2{}^n A_{P_{st,i}}^{inc.}) + P_{stst,i-1} \\
& ({}^n A_{st,i} - {}^n A_{st,i-1}) + P_{stst,i} ({}^n A_{st,i} - {}^n A_{st,i+1}) \\
& + \sum_{j=1}^{N_{rs}} P_{strt:i,j} ({}^n A_{st,i} - {}^n A_{rt,j}) = 0,
\end{aligned} \tag{3.4}$$

where  ${}^n A_{P_{sb,i}}^{inc.}$  and  ${}^n A_{P_{st,i}}^{inc.}$  are the incident magnetic vector potentials, while  ${}^n A_{sb,i}$  and  ${}^n A_{st,i}$  are the nodal magnetic vector potentials of the  $i^{th}$  stator base and tooth, respectively. The  ${}^n i_{s,i} = \mathbf{N}_{qd0,i}^T {}^n \mathbf{i}_{qd0}$  represents the stator magnetic motive force (MMF) of the  $i^{th}$  stator slot, where  $\mathbf{N}_{qd0,i}^T$  is the  $i^{th}$  slot turn function and  ${}^n \mathbf{i}_{qd0}$  is the stator winding current in  $qd0$  frame.

Similarly, for the rotor tooth and base nodes, the KCL can be presented as:

$$\begin{aligned}
& Y_{0P_{rt,j}} ({}^n A_{rt,j} - {}^n A_{rb,j} - {}^n i_{r,j} - 2{}^n A_{P_{rt,j}}^{inc.}) + Y_{0P_{rtrt,j-1}} \\
& ({}^n A_{rt,j} - {}^n A_{rt,j-1} - 2{}^n A_{P_{rtrt,j-1}}^{inc.}) + Y_{0P_{rtrt,j}} ({}^n A_{rt,j} - \\
& {}^n A_{rt,j+1} + 2{}^n A_{P_{rtrt,j}}^{inc.}) + \sum_{i=1}^{N_{ss}} P_{strt} ({}^n A_{rt,j} - {}^n A_{st,i}) = 0,
\end{aligned} \tag{3.5}$$

$$\begin{aligned}
& Y_{0P_{rt,j}} ({}^n A_{rb,j} - {}^n A_{rt,j} + {}^n i_{r,j} + 2{}^n A_{P_{rt,j}}^{inc.}) + \\
& Y_{0P_{rb,j}} ({}^n A_{rb,j} - {}^n A_{rb,j+1} + 2{}^n A_{P_{rb,j}}^{inc.}) + \\
& Y_{0P_{rb,j-1}} ({}^n A_{rb,j} - {}^n A_{rb,j-1} - 2{}^n A_{P_{rb,j-1}}^{inc.}) = 0,
\end{aligned} \tag{3.6}$$

where  ${}^n i_{r,j}$  is the  $j^{th}$  rotor bar current of the  $k^{th}$  TLM iteration of the  $n^{th}$  simulation time-step.

The stator winding linkage flux ( $\lambda_{qd0}$ ) can be represented in terms of the magnetizing and leakage fluxes as follows:

$$\lambda_{qd0} = \sum_{i=1}^{N_{ss}} \mathbf{W}_{qd0,i} Y_{0P_{st,i}} ({}^n A_{st,i} - {}^n A_{sb,i} + 2{}^n A_{P_{st,i}}^{inc.}) + \mathbf{L}_l {}^n \mathbf{i}_{qd0}, \tag{3.7}$$

where  $\mathbf{W}_{qd0,i}$  is the winding function of the  $i^{th}$  stator slot in  $qd0$  frame, and  $\mathbf{L}_l$  is the stator winding leakage inductance, both extensively discussed in [147].

The  $j^{th}$  rotor loop and end ring linkage flux can be represented as:

$$\begin{aligned}
\hat{\lambda}_{r,j} &= Y_{0P_{rt,j}} ({}^n A_{rb,j} - {}^n A_{rt,j} + 2{}^n A_{P_{rt,j}}^{inc.} + {}^n i_{r,j}) - \\
& P_{b,j-1} {}^n i_{r,j-1} + (P_{b,j} + P_{b,j-1} + P_{fe,j} + P_{be,j}) {}^n i_{r,j} \\
& - P_{b,j} {}^n i_{r,j+1} - \frac{P_{fe,j}}{P_{fe}} \sum_{j_1=1}^{N_{rs}} P_{fe,j_1} {}^n i_{r,j_1}
\end{aligned} \tag{3.8}$$

where  $P_{b,j}$ ,  $P_{fe,j}$  and  $P_{be,j}$  are the linear permeances of the rotor bar, front end ring, and back end ring, respectively.

The MEC-based TLM equations of (2)-(7) can be reorganized into a system of equations of:

$$\mathbf{M}_k^n \mathbf{x} = {}_k^n \mathbf{b}, \quad (3.9)$$

where the MEC matrix ( $\mathbf{M}$ ) is dependent on the transmission line admittances.  $\mathbf{M}$ ,  ${}_k^n \mathbf{x}$ , and  ${}_k^n \mathbf{b}$  are ordered as follows due to an advantage discussed in *subsection B*:

$$\begin{pmatrix} \mathbf{M}_{sb-sb} & \mathbf{0} & \mathbf{M}_{sb-si} & \mathbf{0} & \mathbf{M}_{sb-st} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{rb-rb} & \mathbf{0} & \mathbf{M}_{rb-ri} & \mathbf{0} & \mathbf{M}_{rb-rt} \\ \mathbf{M}_{s\lambda-sb} & \mathbf{0} & \mathbf{L}_l & \mathbf{0} & \mathbf{M}_{s\lambda-st} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{r\lambda-rb} & \mathbf{0} & \mathbf{M}_{r\lambda-ri} & \mathbf{0} & \mathbf{M}_{r\lambda-rt} \\ \mathbf{M}_{st-sb} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{st-st} & \mathbf{M}_{st-rt} \\ \mathbf{0} & \mathbf{M}_{rt-rb} & \mathbf{0} & \mathbf{M}_{rt-ri} & \mathbf{M}_{rt-st} & \mathbf{M}_{rt-rt} \end{pmatrix} \quad (3.10)$$

$${}_k^n \mathbf{x} = \left( {}_k^n \mathbf{A}_{sb} \quad {}_k^n \mathbf{A}_{rb} \quad {}_k^n \mathbf{i}_{qd0} \quad {}_k^n \mathbf{i}_r \quad {}_k^n \mathbf{A}_{st} \quad {}_k^n \mathbf{A}_{rt} \right)^T \quad (3.11)$$

$${}_k^n \mathbf{b} = \left( {}_k^n \mathbf{b}_{sb} \quad {}_k^n \mathbf{b}_{rb} \quad \lambda_{qd0} + {}_k^n \mathbf{b}_{s\lambda} \quad \hat{\lambda}_r + {}_k^n \mathbf{b}_{r\lambda} \quad {}_k^n \mathbf{b}_{st} \quad {}_k^n \mathbf{b}_{rt} \right)^T \quad (3.12)$$

where  ${}_k^n \mathbf{b}_{sb}$ ,  ${}_k^n \mathbf{b}_{rb}$ ,  ${}_k^n \mathbf{b}_{s\lambda}$ ,  ${}_k^n \mathbf{b}_{r\lambda}$ ,  ${}_k^n \mathbf{b}_{st}$  and  ${}_k^n \mathbf{b}_{rt}$  are the TLM sources dependent on the incident potentials of the sending end that change in each TLM iteration. Since the MEC-based TLM matrix remains unchanged within each time-step, the solution requires only one LU decomposition per time-step. Meanwhile, the  ${}_k^n \mathbf{b}$  matrix changes, where only sparse forward elimination and backward substitution need to be carried out per TLM iteration.

The TLM procedure continues with the calculation of reflected magnetic vector potentials at the sending end of each permeance ( ${}_k^n \mathbf{A}_P^{ref.}$ ) as follows:

$${}_k^n \mathbf{A}_P^{ref.} = {}_k^n \mathbf{A}_P - {}_k^n \mathbf{A}_P^{inc.}, \quad (3.13)$$

where  ${}_k^n \mathbf{A}_P$  is the potential difference vector of the nodes across each permeance. The reflected potentials of the sending end travel through the transmission lines and become the incident potentials of the decoupled nonlinear permeances (receiving end) without any change since the transmission lines are lossless. The mismatch of the transmission line admittances and the nonlinear permeances results in reflecting potentials at the receiving end, which travel

toward the sending end and become the next incident potentials of the sending end and can be obtained by nonlinear solution of the decoupled nonlinear permeances as follows:

$$Y_{0P}({}^n A_P^{ref.} - {}^n_{k+1} A_P^{inc.}) = P({}^n A_P)({}^n A_P^{ref.} + {}^n_{k+1} A_P^{inc.}), \quad (3.14)$$

As the N-R solution of (3.14) converges for all nonlinear permeances, the incident magnetic vector potentials at the sending end of the transmission lines unconditionally converge and the output variables including stator and rotor currents are obtained.

The time derivative of the state variables, including the stator flux ( $\lambda_{qd0}$ ) and the rotor linkage flux of the  $j^{th}$  rotor loop ( $\lambda_{r,j}$ ) can be calculated from the electrical circuit of the stator and rotor, respectively as follows:

$$\frac{d}{dt} \lambda_{qd0} = \mathbf{v}_{qd0} - r_s \mathbf{i}_{qd0}, \quad \frac{d}{dt} \lambda_{fe} = - \sum_{j=1}^{N_{rs}} r_{fe,j} (i_{fe} - i_{r,j}) \quad (3.15)$$

$$\begin{aligned} \frac{d}{dt} \lambda_{r,j} = & -r_{b,j} (i_{r,j} - i_{r,j+1}) - r_{be,j} i_{r,j} - \\ & r_{b,j-1} (i_{r,j} - i_{r,j-1}) - r_{fe,j} (i_{r,j} - i_{fe}), \end{aligned} \quad (3.16)$$

where  $\mathbf{v}_{qd0}$  and  $r_s$  are the stator voltage and winding resistance.  $r_{b,j}$ ,  $r_{be,j}$ , and  $r_{fe,j}$  are the bar, back end ring and front end ring of the  $j^{th}$  rotor loop, respectively. The time-stepping procedure continues to the next time-step with forward Euler discretization of (3.15) and (3.16).

### 3.2.3 Matrix Re-Ordering, Partially Pre-Calculated LU Decomposition and Reduced Order Sparse Solver

Conventional randomly organized MEC-based TLM system of equations requires one LU decomposition of the MEC matrix with the size of  $2N_{ss} + 3N_{rs} + 3$  per time-step. However, only the  $\mathbf{M}_{st-rt}$ ,  $\mathbf{M}_{rt-st}$ ,  $\mathbf{M}_{st-st}$  and  $\mathbf{M}_{rt-rt}$  submatrices change during simulation due to movement through the change of  $Y_{0P_{str,j}}$ . The idea is to represent the MEC-based TLM system of equations ordered as (3.10), to keep the majority of the matrix unchanged during movement. In column by column approach of the left-looking Gilbert-Peierls algorithm for sparse LU decomposition, each element  $M'(r_i, c_i)$  on the decomposed

matrix is only dependent on the already computed elements in the decomposed sub-matrix of  $\mathbf{M}'(1 : r_i, 1 : c_i)$ , where  $r_i$  and  $c_i$  are the row and column indices respectively. Taking advantage of the algorithm, the LU decomposition of the majority of the matrix remains unchanged during the time-stepped procedure and make it amenable for partial pre-calculation to satisfy the real-time simulation timing constraints.

Fig. 3.2 shows the LU decomposition procedure broken into the two stages of pre-processing and real-time processing. The re-ordered non-zero pattern of matrix  $\mathbf{M}$  is shown in Fig. 3.2 (a), where the sub-matrices of  $\mathbf{M}_{11}$ ,  $\mathbf{M}_{12}$ ,  $\mathbf{M}_{21}$  are fixed through the entire simulation and  $\mathbf{M}_{22,1}$  contains the elements of  $\mathbf{M}_{st-st}$  and  $\mathbf{M}_{rt-rt}$ , which are motion independent. Fig. 3.2 (b) shows the non-zero pattern of the partially pre-calculated LU decomposition of matrix  $\mathbf{M}$ , where the sub-matrices of  $\mathbf{M}'_{11}$ ,  $\mathbf{M}'_{12}$ ,  $\mathbf{M}'_{21}$  are calculated independent of  $\mathbf{M}_{22,1}$  based on the left-looking Gilbert-Peierls algorithm. The pre-calculated sub-matrix of  $\mathbf{M}_{22,2}$  is part of the motion dependent sub-matrix of  $\mathbf{M}_{22,3}$ , which is dependent on already decomposed fixed sub-matrices of  $\mathbf{M}'_{11}$ ,  $\mathbf{M}'_{12}$ ,  $\mathbf{M}'_{21}$  to maximize the benefit of pre-calculation. In the real-time process of Fig. 3.2 (c), the motion dependent terms of  $\mathbf{M}_{st-rt}$ ,  $\mathbf{M}_{rt-st}$ ,  $\mathbf{M}_{st-st}$  and  $\mathbf{M}_{rt-rt}$  sub-matrices are accumulated on the pre-calculated  $\mathbf{M}_{22,2}$ , while the rest of sub-matrices are unchanged. Finally, only the sparse LU decomposition of the  $\mathbf{M}_{22,3}$  needs to be carried out to obtain  $\mathbf{M}'_{22}$ . As a result, the LU decomposition computation burden of a matrix dimension of  $2N_{ss} + 3N_{rs} + 3$  is considerably reduced to  $N_{ss} + N_{rs}$ .

To maximize the solver efficiency in the pre-processing stage, the sparsity pattern of the partially pre-calculated LU decomposition of the MEC-based TLM matrix (Fig. 3.2 (b)) is stored through 6 auxiliary matrices, where SNZL/SNZU addresses the first nonzero number in each column of the lower/upper triangular, NNZL/NNZU addresses the number of nonzeros in each column of the lower/upper triangular, NZRL/NZRU addresses the nonzeros row index of the lower/upper triangular. Afterward, the six auxiliary matrices are partially updated during the real-time processing of reduced-order sparse LU decomposition to be readily used in the sparse forward elimination

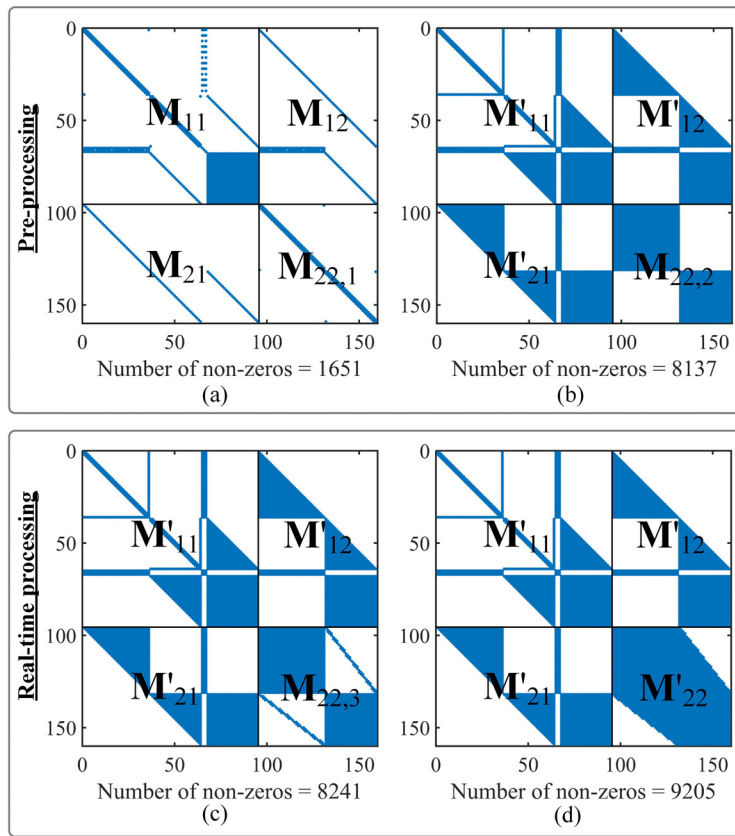


Figure 3.2: Proposed RT-TLM MEC matrix LU decomposition procedure.



and backward substitution.

### 3.2.4 Parallelism Exploitation

One source of parallelism can be found in the column-by-column approach of the left-looking Gilbert-Peierls algorithm for sparse LU decomposition with respect to rows when the column is being updated by fill-ins calculation. Similarly, for the sparse forward elimination and backward substitution, computations are independent with respect to rows.

Owing to the TLM algorithm, another source of parallelism can be found in the nonlinear solution of (3.14) to find the next incident potentials of the sending end from decoupled nonlinear permeances at the receiving end, which makes the TLM method highly effective on parallel processors rather than sequential ones. FPGAs as the massively paralleled hardware can be utilized to take the benefit of parallel processing to achieve real-time simulation. The parallelism exploitation is extensively discussed in Section III.

## 3.3 MEC-Based RT-TLM Hardware Implementation

When the case study machine changes, the pre-processing stage includes defining the machine geometry and material properties, calculation of the electrical parameters, calculation of the linear, TLM and motion independent permeances, the formation of motion independent MEC-based TLM matrix of Fig. 3.2 (a), and finally performing the partial LU decomposition to calculate the matrix of Fig. 3.2 (b). The pre-processing stage can be performed on either the same hardware platform or another CPU-based processor and loaded into the hardware, where both are followed by the same real-time processing stage. The later strategy is used in this work for simplicity.

The proposed MEC-based RT-TLM methodology state diagram is shown in Fig. 3.3. The diagram includes three cascaded loops, i.e. the time-step, the TLM, and the N-R loop. It can be seen that with the proposed methodology, taking the advantage of TLM method, the reduced-order sparse LU decom-

position in State S2, which has the most computational burden, needs to be carried out only once per time-step. However, the sparse forward elimination and backward substitution in State S4 should be calculated for the TLM number of iterations ( $N_{TLM}$ ) per time-step. In addition, massive parallelism can be found on State S7 as most inner loop with highest number of iterations ( $N_{TLM} \times N_{N-R}$ ) per time-step, which is the calculation of the next incident magnetic vector potential for decoupled nonlinear permeances of  $P_{sb}$ ,  $P_{st}$ ,  $P_{trt}$ ,  $P_{rt}$  and  $P_{rb}$  through the N-R method. The numbered items in states indicate the component level parallelism, while the state-to-state transition is sequential. The external inputs and outputs are connected to the FPGA board input and output pins (I/O pins) for HIL emulation.

Fig. 3.4 shows the hardware implementation, where paralleled and pipelined implementation are shown for the most critical designed states of Fig. 3.3 in terms of execution time per iteration and the number of iterations per time-step, i.e., S2: reduced-order sparse LU decomposition, S4: sparse forward elimination and backward substitution, and S7: calculation of nonlinear permeances next incident magnetic vector potentials. The parallelism in the algorithm-level and operation-level is fully exploited through parallelism and pipelining schemes, where parallelism is simultaneous operations through parallel hardware and pipelining is a chain of independent operations in each clock through the same hardware due to the involvement of RAMs with a limited number of ports (dual port). The data is pipelined from the component input to the output by changing the row and column indices of  $r_i$  and  $c_i$ , respectively.

The maximum latency is considered for each floating point operation IP core to achieve the highest drivable FPGA clock frequency and total reduction in time-step size, which is 28 clocks for floating point divisions (Flp Div), 23 clocks for the logarithm (Flp Log), 12 clocks for additions/subtractions (Flp Add/Sub), 8 clocks for multiplications (Flp Mul), 2 clocks for 16 bits fixed point addition (Fix Add), 1 clock for 8 bits fixed point multiplication (Fix Mul), and 1 clock for RAMs read/write. The register IP blocks used for signal timing coordination in the pipelining scheme is not shown in the Fig. 3.4 to improve readability. A compromise between the result accuracy, hardware

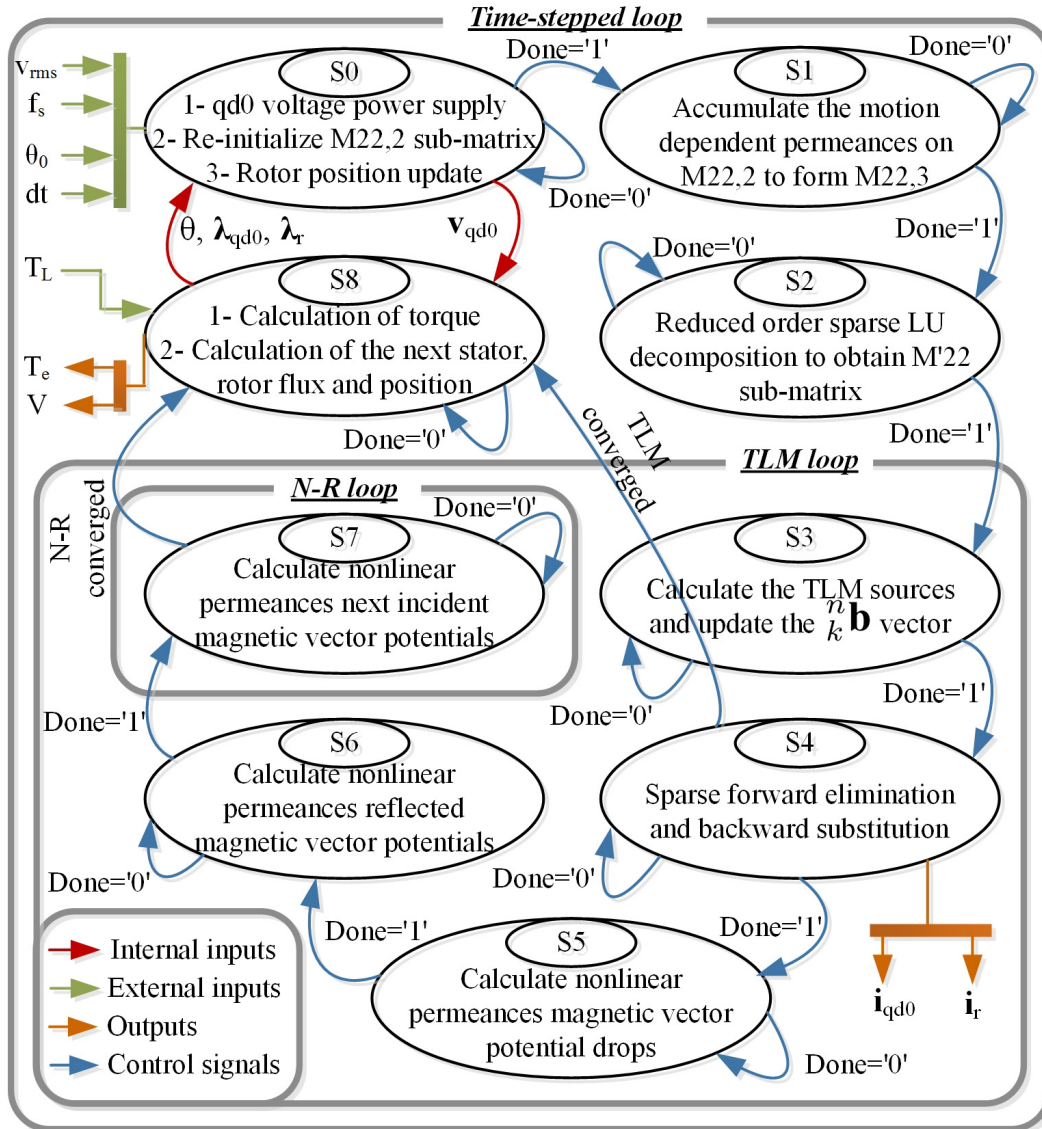


Figure 3.3: Proposed MEC-based RT-TLM FPGA implementation state diagram.

resource, and achievable frequency make the standard 32-bit single precision number representation for floating point operations ideal.

The hardware is designed by 10,000 lines of handwritten VHDL code in Xilinx Vivado<sup>®</sup> software and the bit stream is generated after the behavioral simulation, synthesis, design initialization, optimization and placing the design, respectively. The bit-stream is implemented using JTAG interface on the Xilinx<sup>®</sup> Virtex UltraScale+ XCVU9P FPGA board shown in Fig. 3.5. The digital outputs of the FPGA board on FMC port are converted to analog using DAC board to be shown on the oscilloscope.

## 3.4 Real-Time Simulation Results

In this section, the fidelity of the proposed MEC-based RT-TLM method under transient start-up and steady-state operation of a wye-connected, 3-hp and 4-pole squirrel-cage induction motor with closed rotor slots is evaluated with the 2-D FEM model on the commercial Jmag-Designer<sup>®</sup> software.

### 3.4.1 Transient Start-Up

Figs. 3.6 and 3.7 show the induction motor real-time stator phase current and rotor speed transients validated by FEM and experimental results when the induction motor is directly line-fed from a 208 V three-phase 60 Hz power supply. During the start-up period of  $t=0$  s to  $t=0.4$  s, a high ( $42 A_{rms}$ ) no load inrush current passes through the winding to rotate the rotor to the asynchronous speed of 1792 rpm. Afterward, during the time period of  $t=0.4$  s to  $t=0.5$  s, the winding current settles down to the steady-state no load current ( $2.8 A_{rms}$ ).

It is shown that the time-domain transient current wave shape of the proposed real-time method is in a good agreement with FEM and experimental results.

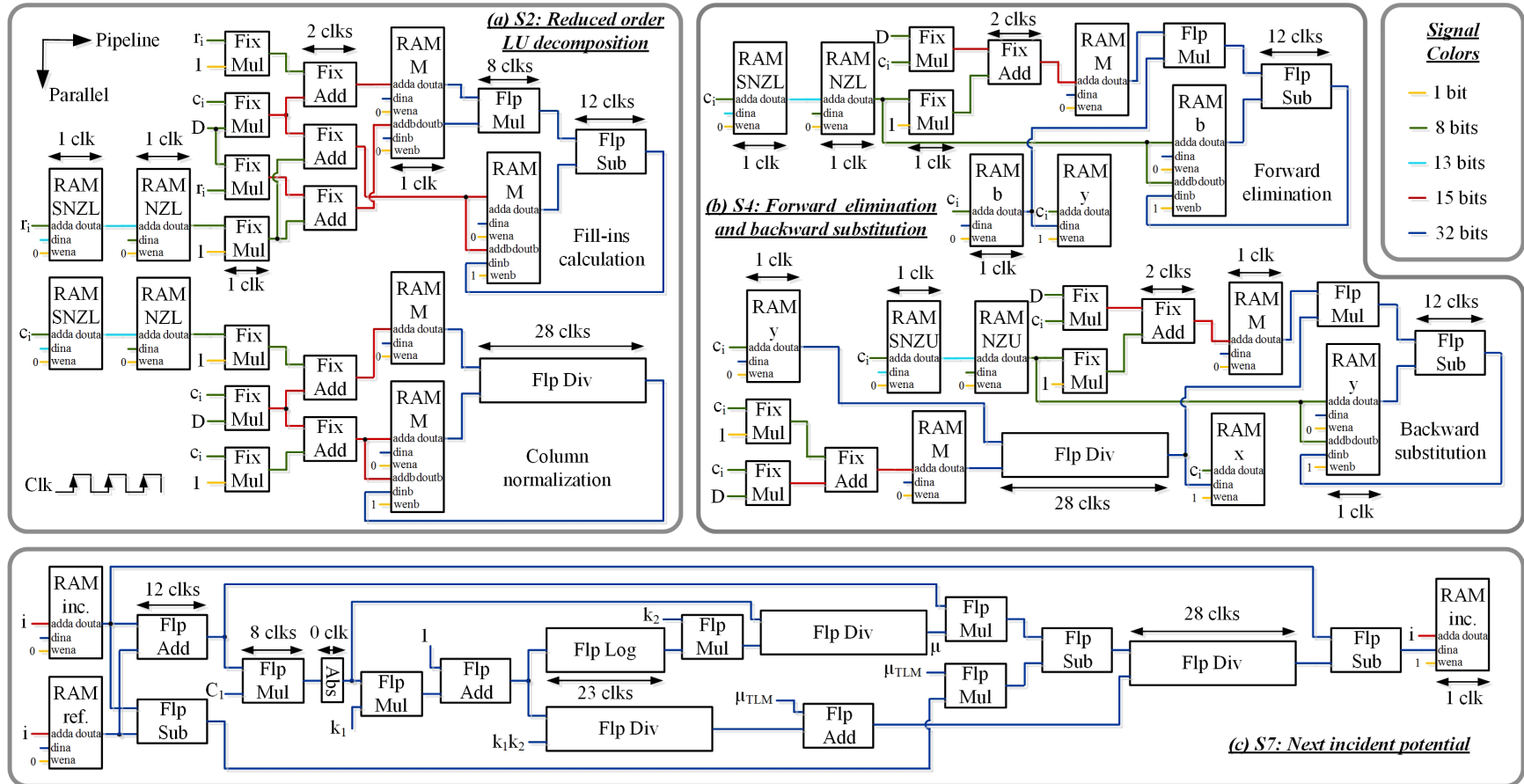


Figure 3.4: Massively parallelled and deeply pipelined FPGA hardware architecture. (a) S2: reduced-order sparse LU decomposition, (b) S4: sparse forward elimination and backward substitution, (c) S7: next incident magnetic vector potentials.

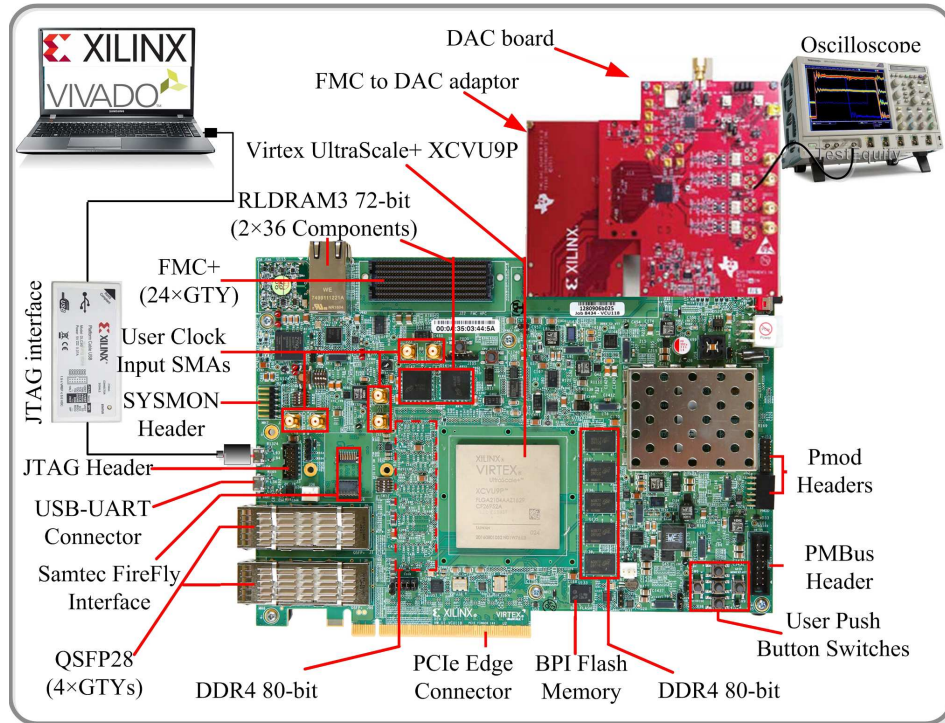


Figure 3.5: Real-time simulator hardware set up.

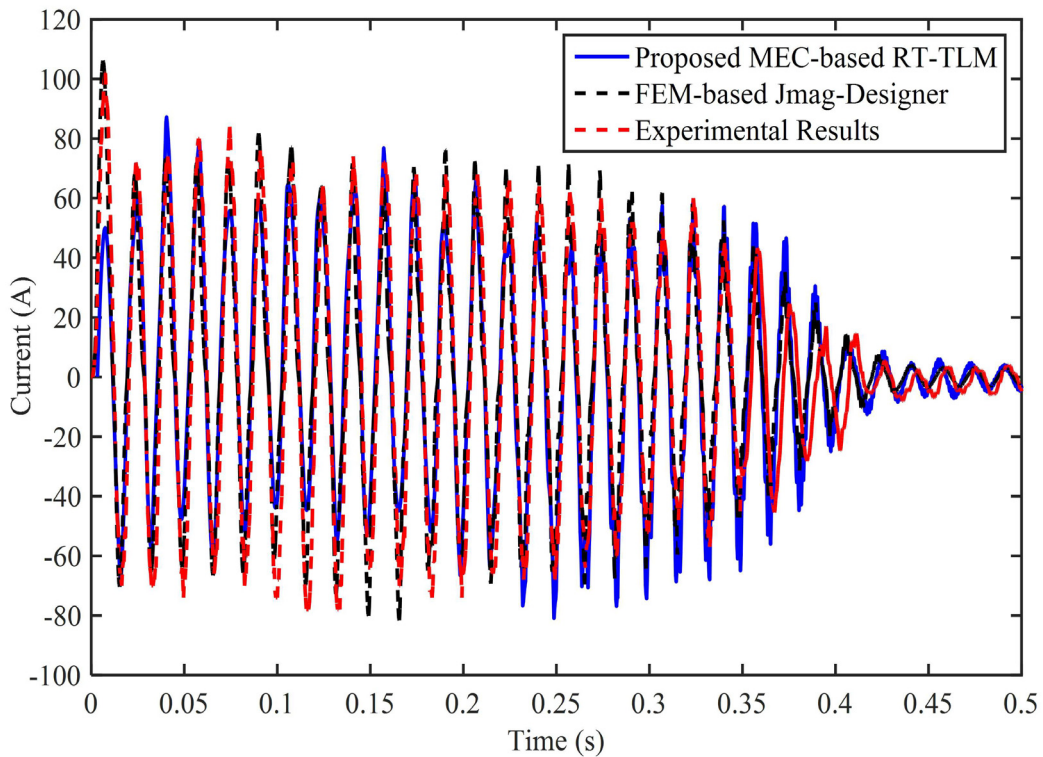


Figure 3.6: Stator winding phase current transients during no load start-up.



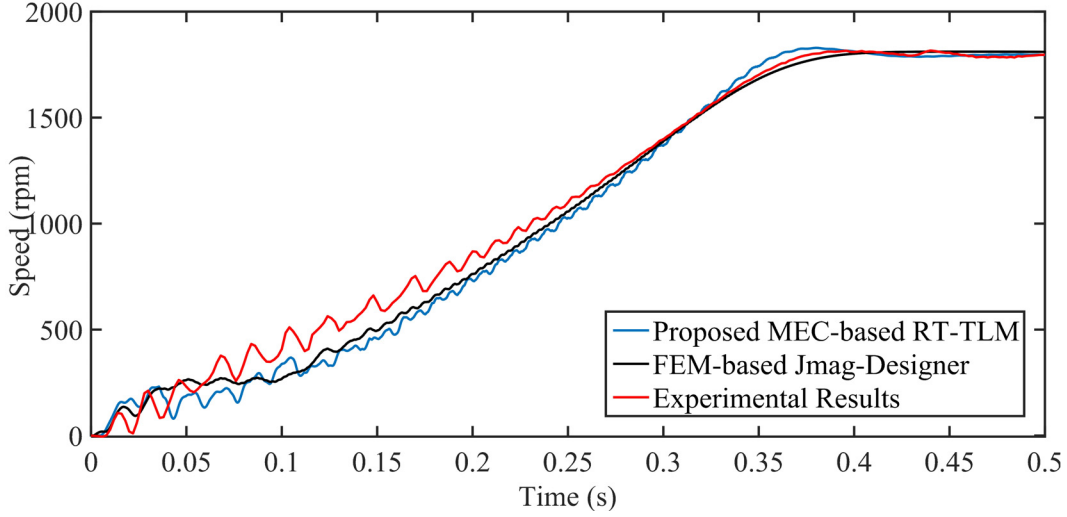


Figure 3.7: Rotor speed transients during no load start-up.

### 3.4.2 Steady-State

In this part, the induction motor under ideal steady-state operation is compared with abnormal operating conditions under various types of machine faults and supply disturbances, which do not initially and necessarily lead the machine operation to fail. The stator winding current harmonic spectrum is shown for 6 cases in Fig. 3.8 with a line-fed 208 V three-phase 60 Hz power supply and full load torque of 13 Nm at the speed of 1746 rpm.

#### Case 1: Healthy

In practice, in a healthy condition, the air-gap MMF and flux are not purely sinusoidal and contain spatial harmonics as a result of winding distribution and slotting effect. The harmonic spectrum of the stator winding current using the Discrete Fourier Transform (DFT) of Fig. 3.8 (a) shows those odd harmonic contents.

#### Case 2: Stator winding inter-turn short circuit fault

The 4-pole induction motor is equipped with 2 parallel coils per phase with 40 turns per coil. The inter-turn short-circuit fault is modeled by reducing the number of turns of one coil of *phase-a* to 30. Under the assumption that the inter-turn fault is not destructive to the machine, Fig. 3.8 (b) shows the

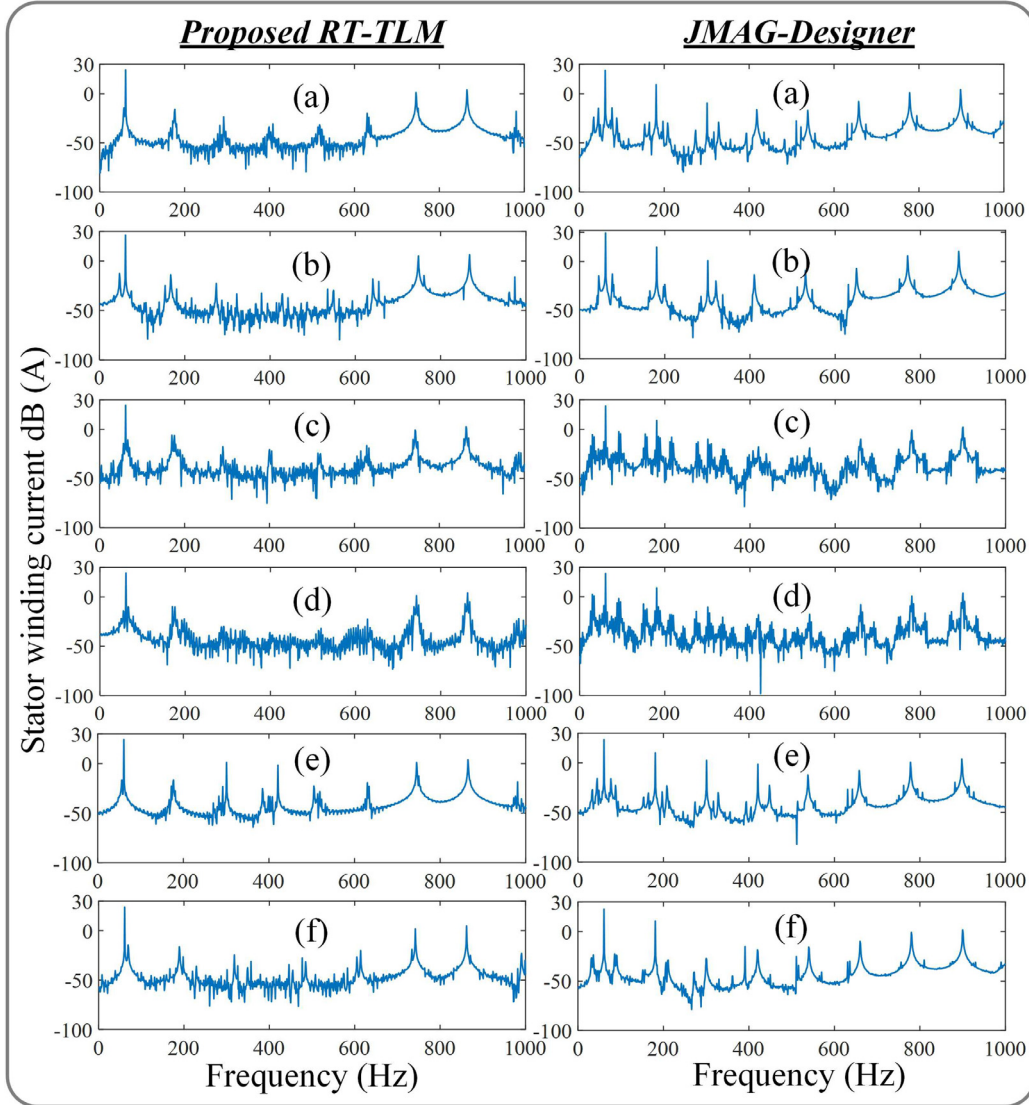


Figure 3.8: Steady-state stator current result validation. (a) Case 1: Healthy, (b) Case 2: Stator winding inter-turn short circuit fault, (c) Case 3: Rotor broken bar fault, (d) Case 4: Rotor broken end ring fault, (e) Case 5: Supply 5<sup>th</sup> and 7<sup>th</sup> harmonics, (f) Case 6: Unbalanced supply.



current amplitude of *phase-a* is increased as a result of the partial short-circuit winding. In addition, side band harmonics of the main and spatial harmonics manifested in the winding current DFT as expected.

### **Case 3: Rotor broken bar fault**

The induction motor case study contains 28 rotor bars. The broken rotor bar is modeled through a  $10\text{ m}\Omega$  resistance in the electrical circuit instead of the  $48.72\text{ }\mu\Omega$  of the healthy bar, since in most cases the bar is cracked rather than entirely disconnected. From Fig. 3.8 (c), it can be seen the fault broadens the harmonic spectrum in the current waveform as a result of facing the broken bar to the stator winding MMF.

### **Case 4: Rotor broken end ring fault**

Similar to the broken bar, the cracked end ring is modeled by substituting a  $100\text{ }\mu\Omega$  resistance instead of the healthy end ring resistance of  $1.38\text{ }\mu\Omega$ . The broken end ring interferes with the rotor bar current flow, and for the same reason as the broken bar, it results in broadening the harmonics as shown in Fig. 3.8 (d).

### **Case 5: Supply 5<sup>th</sup> and 7<sup>th</sup> harmonics**

In the case of direct grid connected induction motors, the supply voltage is not purely sinusoidal. Harmonics 5 and 7 are the dominant time harmonics in the power system due to the presence of power electronic devices. In this case, 10% harmonics of 5<sup>th</sup> and 7<sup>th</sup> are considered in the voltage supply to evaluate the ability of the model in capturing those harmonics with the real-time sampling period. Fig. 3.8 (e) shows that the harmonics are effectively captured in the DFT of the winding current.

### **Case 6: Unbalanced supply**

Finally, The DFT of the stator current with 10% voltage drop in *phase-a* is shown in Fig. 3.8 (f).

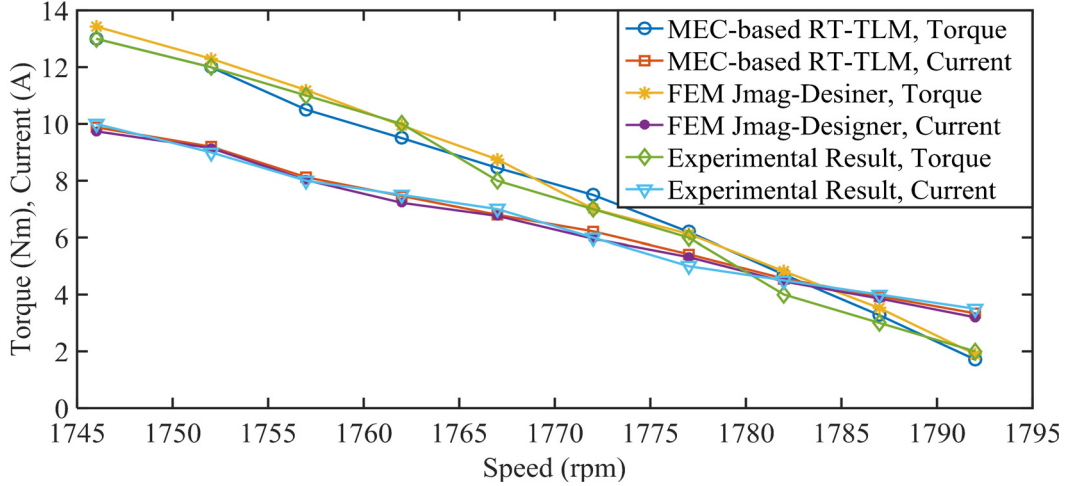


Figure 3.9: Steady-state torque and current vs. speed from no load to full load.

Fig. 3.9 shows the steady-state torque and current versus speed to validate the proposed RT-TLM results with FEM.

The comparison of the MEC-based real-time and 2-D FEM results shows the validity and efficiency of the proposed RT-TLM method in estimating the stator winding phase current harmonic contents. The discrepancy of the results is mainly due to the distributed effects as the MEC is a lumped parameter model, which is the reason for generally broader harmonic contents of FEM. For instance, the spatial harmonics in the air-gap cannot be modeled accurately due to the lumped modeling of the stator tooth to rotor tooth permeances. On the hardware side, utilization of the 32-bit single precision floating point number of the MEC-based real-time model in comparison to the 64-bit double precision floating point number of the FEM model in Jmag-Designer<sup>®</sup> can result in some deviations by the accumulation of the round-off errors through the computation process. Fig. 3.10 shows the FEM magnetic field and current density distribution analyzed by Jmag-Designer<sup>®</sup> for healthy and broken bar conditions. It is shown that the rotor broken bar prevents the normal distribution of the magnetic field surrounding the broken bar as well as the stator winding faced to the bar.

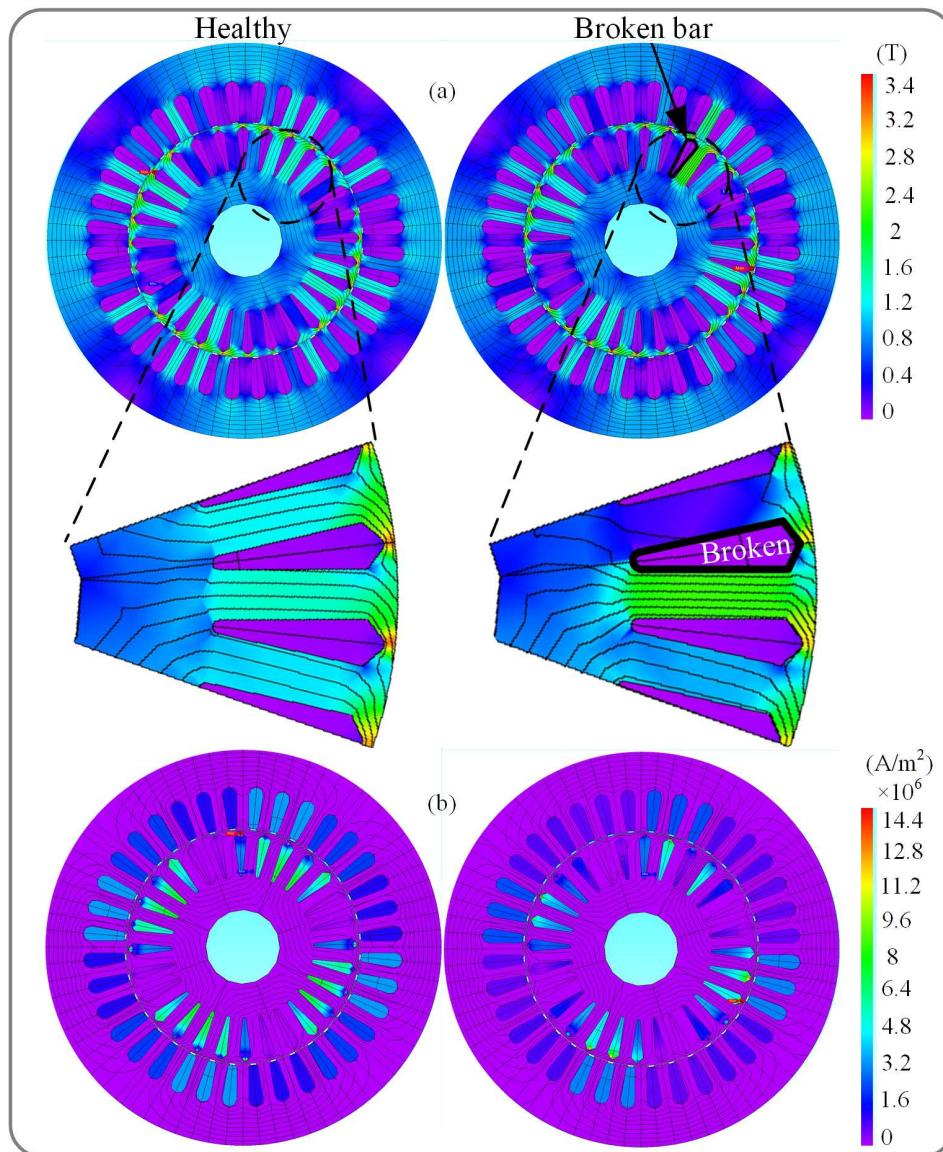


Figure 3.10: FEM results of Jmag-Designer<sup>®</sup> for healthy and broken rotor bar. (a) Magnetic flux density, (b) Current density.

Table 3.1: Hardware resource utilization

Resource	Utilization	Available	Percentage
Look-up table (LUT)	63,421	1,182,240	5.36
LUT-RAM	4,343	591,840	0.73
Flip flop (FF)	96,163	2,364,480	4.07
Block RAM (BRAM)	53.50	2,160	2.48
Input/output (IO) port	78	832	9.38
Global buffer (BUFG)	2	1,800	0.11

Table 3.2: Latencies of each state in real-time processing

State No.	No. of clocks per iteration	No. of iterations per time-step	No. of clocks per time-step	Execution time ( $\mu s$ )
S0	2,498	1	2,498	8.32
S1	4,081	1	4,081	13.6
S2	95,128	1	95,128	317.1
S3	153	$N_{TLM}$	612	2.04
S4	9,582	$N_{TLM}$	38,328	127.76
S5	138	$N_{TLM}$	552	1.84
S6	115	$N_{TLM}$	460	1.53
S7	534	$N_{TLM} \times N_{N-R}$	6,408	21.36
S8	120	1	120	0.4
<b>Total</b>	-	-	<b>148,187</b>	<b>493.96</b>

### 3.4.3 Hardware Resource

The implementation of the proposed RT-TLM method is performed on Xilinx<sup>®</sup> Virtex UltraScale+ XCVU9P FPGA board, which is among the highest speed and largest FPGAs available on the market. The concern behind selecting the device is more to achieve the highest clock frequency rather than hardware resource. Table 5.1 shows the hardware resource utilization. Since the design is deeply pipelined, just a small portion of the device is used.

### 3.4.4 Timing Analysis

Table 5.2 shows the latencies of the real-time processing states of Fig. 3.3. The MEC-based RT-TLM solution timing is reported based on the required number of 3 N-R and 4 TLM iterations with the convergence tolerance of  $10^{-3}$  for both iterative loops when the  $\mu_{TLM} = 1000\mu_0$  is chosen.

The designed hardware on the targeted FPGA is derivable with the maximum clock frequency of 300 MHz. Consequently, considering the number of

clock latencies of Table 5.2 and an idle time of  $6.04 \mu s$ , the minimum time-step of  $500 \mu s$  is achieved. The  $500 \mu s$  time-step can be considered small enough to efficiently model the transients of the induction motor since it delivers 33 data points on each cycle of the 60 Hz. Based on Nyquist theorem, a signal can be reproduced with at least two data points per cycle of the highest frequency content of a periodic waveform. Accordingly, the achieved time-step can reproduce the waveform up to the 16<sup>th</sup> harmonic, i.e. 960 Hz. The  $500 \mu s$  electric machine model can be interfaced with a few  $\mu s$  fast switching power converter model in an electric machine drive system through digital integration of the converter output PWM voltage over the PWM period to supply the electric machine in a variable frequency drive system. The proposed methodology is efficient and can be extended not only to MEC of other types of electric machines but also to FEM computation of any other small to medium size magnetodynamic problem with moving structures subject to availability of the hardware resource and satisfying the real-time constraint.

### 3.5 Summary

In electric machine drive systems, HIL emulation provides accurate testing of actual control system prototypes and protection devices interfaced with the electric machine model on a real-time simulator in a non-destructive environment particularly when faults are studied. A compromise between the model accuracy and computational burden makes the MEC model ideal for real-time simulation of electric machines. However, satisfying the timing constraints of real-time simulation to accommodate internal machine faults is still challenging due to the nonlinearity and movement of electric machines. In this chapter, the TLM method is utilized to keep the MEC coefficient matrix unchanged during nonlinear iterations. Afterwards, for the first time, the entire potential of the TLM method for pre-calculation is exploited by proposing an efficient matrix re-ordering combined with the left-looking Gilbert-Peierls algorithm to minimize the computational burden of the sparse MEC matrix LU decomposition required in each time-step due to movement. Furthermore, the massive

hardware architecture of the FPGA is used as the platform for implementation to fully exploit parallelism. With the proposed MEC-based RT-TLM method, the minimum time-step as low as  $500 \mu s$  can be achieved and the results validation with 2-D FEM model of the commercial Jmag-Designer<sup>®</sup> software shows the accuracy and efficiency of the proposed methodology.

# Chapter 4

## Hardware Acceleration of Finite Element Method Model of Linear Induction Motor

### 4.1 Introduction

Linear induction machines (LIMs) are able to provide levitation force in addition to the propulsion force of conventional rotary machines. Due to the distinct feature, LIMs are widely used in industry for contactless motion applications, including magnetically levitated (MagLev) machines [148, 149] and electromagnetic launch systems (EMLS) [150]. The single-sided LIM consists of a short primary mover with three-phase windings connected to the source, and the secondary part that includes an aluminum sheet to conduct the induced eddy currents with a back iron for the primary flux path.

Designing an electric machine requires numerous cycles of simulations for performance evaluation and testing before construction. Various models of LIM include equivalent circuit model [151], 2 axis  $q - d$  model for control purposes [152–154], and FEM model for design purposes [155]. FEM is well recognized as one of the most accurate models for detailed magnetic field analysis of electric machines considering the nonlinearity of the iron core; however, it suffers from high computational burden [156].

The most time consuming part of a nonlinear time-stepped FEM problem is the solution of a large system of linear equations in every iteration and time-step. The aim of sparse solvers is to reduce the computation time as well as



the memory usage [157]. Generally, sparse solver algorithms include storage of the non-zero matrix elements using compressed memory storage schemes, partial pivoting to avoid zeros on the main diagonal, symbolic analysis to find the fill-ins to allocate memory and performing LU decomposition, and finally using forward elimination as well as back substitution in order to find the solution [158]. This work is based on left-looking Gilbert-Peierls algorithm for LU decomposition as a direct sparse solver for the system of linear equations originally proposed in [159] and used for speed up in [160–162].

Parallel hardwired architecture, reconfigurability and reliability of FPGAs make them an attractive platform for hardware-based acceleration of FEM computations. Acceleration of system of linear equations is a topic of interest in many science and engineering applications. There are extensive works on linear solvers and algorithms for acceleration on FPGA, including LU decomposition for circuit simulation utilizing matrix column dependency [163], matrix re-ordering in bordered diagonal block (BDB) scheme [164], preconditioned conjugate gradient algorithm (PCG) [165,166], and the Jacobi iterative method [167]. More specific study performed on matrix multiplication in [168], proposed a new striping method for sparse matrix vector multiplication as the main kernel of iterative conjugate gradient (CG) method.

Up to now, no research has been reported on FPGA-based acceleration of FEM simulation for induction machines. In most previous studies, the linear solver acceleration on FPGA is designed for applications with fixed position non-zero matrix elements through the simulation time-steps, where the symbolic analysis was needed to be performed only once for the whole simulation. However, in FEM of induction machines due to the nonlinearity and movement the coefficient matrix changes in each iteration and time-step.

In this chapter, the complete problem of 2-D FEM emulation of LIM on FPGA is addressed and a new efficient sparse linear solver is proposed to speed up the emulation. To do so, first the finite element model of a LIM considering the movement and nonlinearity of the iron core is presented, which results in the solution of a sparse system of linear equations. Second, a new sparse matrix solver based on the left looking Gilbert-Peierls algorithm for LU decomposition



as a direct method is proposed. Then, the hardware implementation of the model on Virtex-7 FPGA is explained using deeply pipelined and massively paralleled scheme. Finally, the results are verified with JMAG-Designer<sup>®</sup> software and discussions are provided.

## 4.2 Finite Element Hardware Acceleration

### 4.2.1 Model Formulation and Numerical Solution

The Maxwell's equation of a transient 2-D eddy current problem representing a LIM, discretized in time domain using Backward Euler method can be derived as:

$$\nabla \cdot \frac{1}{\mu} \nabla A(t + \Delta t) - \sigma \frac{A(t + \Delta t)}{\Delta t} + \sigma \frac{A(t)}{\Delta t} + J_s(t + \Delta t) = R, \quad (4.1)$$

where  $\Delta t$  is the time-step,  $\mu$  is the magnetic permeability,  $\sigma$  is conductivity, and  $J_s$  the is current density of the external current source.  $A$  is the magnetic vector potential in  $z$  direction within the first order triangular element shown in Fig. 4.1 and has been defined as follows [169]:

$$A(x, y) = \sum_{i=1}^3 \frac{1}{2D} (p_i + q_i x + r_i y) A_i, \quad (4.2)$$

where  $D$  is the element area,  $p_1 = x_2 y_3 - x_3 y_2$ ,  $q_1 = y_2 - y_3$ ,  $r_1 = x_3 - x_2$ , and  $p_2, p_3, q_2, q_3, r_2, r_3$  can be calculated by cyclic permutation of the indices. The coefficients of the magnetic vector potential in (4.2) are called shape functions.

In (4.1),  $R$  is the residual value arising from the residual method approximate solution of FEM for calculation of the magnetic vector potential by satisfying the following equation:

$$\int_{\Omega} W R d\Omega = 0, \quad (4.3)$$

where  $\Omega$  is the study domain and  $W$  is the weighting function, which is considered equal to the shape function of (4.2) according to Galerkin's method. For numerical solution, the matrix form of (4.1) using (4.3) needs to be obtained for the study domain nodes as follows:

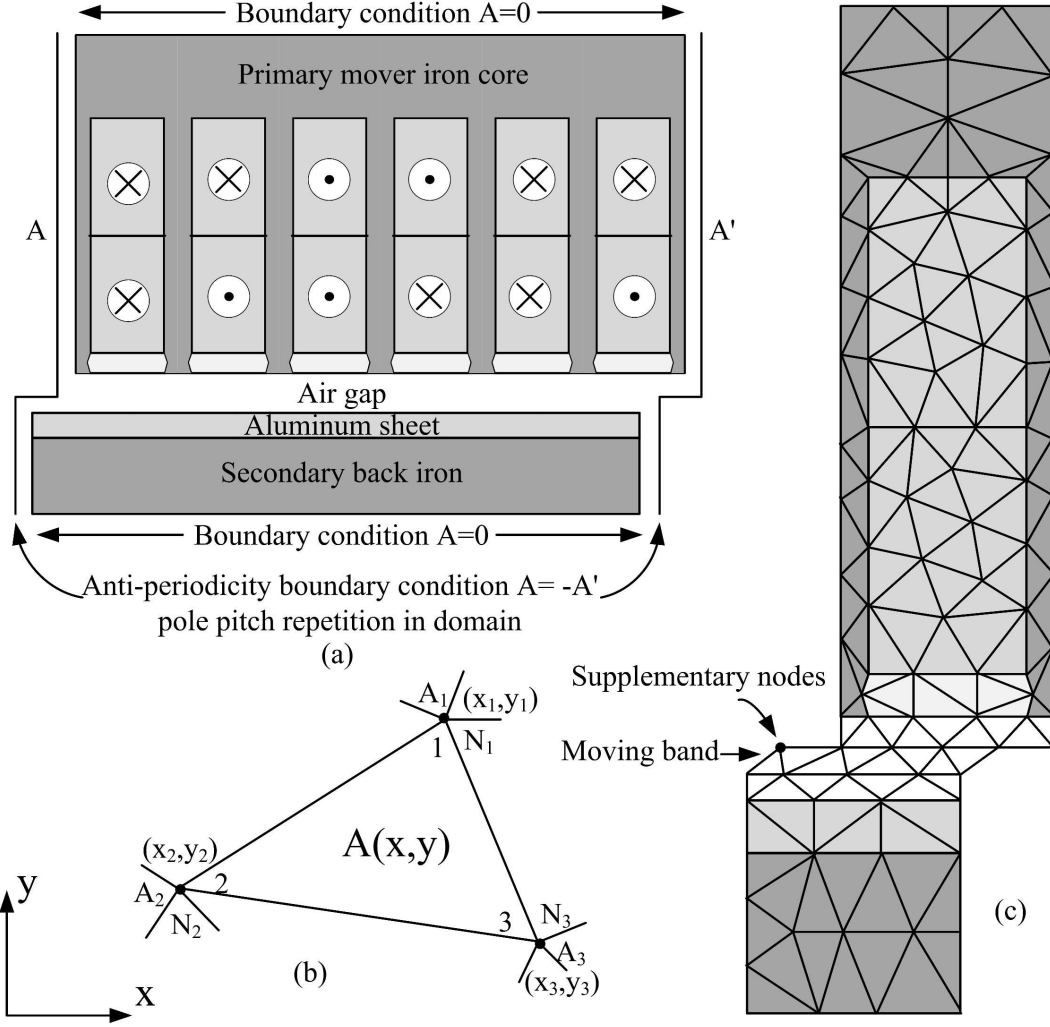


Figure 4.1: Linear induction machine (a) structure and boundary conditions (b) triangular element (c) one slot meshing during movement.

The first term of (4.1) corresponds to the space derivative of the magnetic vector potential, which is nonlinear for the primary and secondary iron cores and linear for other parts of the machine as follows:

$$\int_{\Omega} W \nabla \cdot \frac{1}{\mu} \nabla A(t + \Delta t) d\Omega = -\frac{1}{4\mu D} \cdot \begin{pmatrix} q_1 q_1 + r_1 r_1 & q_1 q_2 + r_1 r_2 & q_1 q_3 + r_1 r_3 \\ q_2 q_1 + r_2 r_1 & q_2 q_2 + r_2 r_2 & q_2 q_3 + r_2 r_3 \\ q_3 q_1 + r_3 r_1 & q_3 q_2 + r_3 r_2 & q_3 q_3 + r_3 r_3 \end{pmatrix} \begin{pmatrix} A_1(t + \Delta t) \\ A_2(t + \Delta t) \\ A_3(t + \Delta t) \end{pmatrix}. \quad (4.4)$$

The second and third terms of (4.1) represent the induced eddy current,

which is non-zero only in the aluminum sheet as follows:

$$\int_{\Omega} W \left( -\sigma \frac{A(t + \Delta t)}{\Delta t} \right) d\Omega = -\frac{\sigma D}{6\Delta t} \cdot \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix} \begin{pmatrix} A_1(t + \Delta t) \\ A_2(t + \Delta t) \\ A_3(t + \Delta t) \end{pmatrix}, \quad (4.5)$$

$$\int_{\Omega} W \sigma \frac{A(t)}{\Delta t} d\Omega = \frac{\sigma D}{6\Delta t} \begin{pmatrix} A_1(t) + 0.5A_2(t) + 0.5A_3(t) \\ 0.5A_1(t) + A_2(t) + 0.5A_3(t) \\ 0.5A_1(t) + 0.5A_2(t) + A_3(t) \end{pmatrix}. \quad (4.6)$$

The fourth term of (4.1) represents the external current source, which is non-zero only in the primary windings as follows:

$$\int_{\Omega} W J_s(t + \Delta t) d\Omega = \frac{J_s(t + \Delta t) D}{3} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \quad (4.7)$$

For each element in the study domain, these matrices should be evaluated for the three nodes of the element. Based on the moving band technique, the meshing of the air gap changes in each time-step to connect the moving band nodes of the stationary part to the moving part. Equations (4.4), (4.5) should be assembled to a global matrix known as the stiffness matrix ( $\mathbf{S}_g$ ). Furthermore, (4.6) and (4.7) are the source terms ( $\mathbf{b}_g$ ) and should be considered as the right hand side of the system of nonlinear equations of FEM as follows:

$$\mathbf{S}_g \mathbf{A}^{t+1} = \mathbf{b}_g, \quad (4.8)$$

where  $\mathbf{S}_g \in \mathbb{R}^{N \times N}$ ,  $\mathbf{A} \in \mathbb{R}^{N \times 1}$ ,  $\mathbf{b}_g \in \mathbb{R}^{N \times 1}$  and  $N$  is the number of nodes in the study domain.

Nonlinear solution of (4.8), due to dependency of  $\mathbf{S}_g$  matrix on  $\mathbf{A}$ , can be obtained by Newton-Raphson method as follows:

$$\mathbf{A}_{i+1}^{t+1} = \mathbf{A}_i^{t+1} - \mathbf{J}_g^{-1} (\mathbf{S}_g \mathbf{A}_i^{t+1} - \mathbf{b}_g), \quad (4.9)$$

where  $i$  is the iteration number.  $\mathbf{J}_g$  is the global Jacobian matrix, which is the derivative of the system equation ( $\mathbf{P}_g = \mathbf{S}_g \mathbf{A}_i^{t+1} - \mathbf{b}_g$ ) with respect to the unknown vector. The local Jacobian matrix for each element ( $\mathbf{J}_l$ ) that should

be assembled in the  $\mathbf{J}_g$  can be obtained from:

$$\mathbf{J}_l = \begin{pmatrix} \frac{\partial P_1}{\partial A_1} & \frac{\partial P_1}{\partial A_2} & \frac{\partial P_1}{\partial A_3} \\ \frac{\partial P_2}{\partial A_1} & \frac{\partial P_2}{\partial A_2} & \frac{\partial P_2}{\partial A_3} \\ \frac{\partial P_3}{\partial A_1} & \frac{\partial P_3}{\partial A_2} & \frac{\partial P_3}{\partial A_3} \end{pmatrix}, \quad (4.10)$$

where

$$P_i(k) = \sum_{l=1}^3 \frac{\nu}{4D} (q_l q_k + r_l r_k) A_l, \quad (4.11)$$

where  $\nu$  is the magnetic reluctivity.

To find the Jacobian matrix terms, the derivative of (4.11) with respect to magnetic vector potential can be found as follows:

$$\begin{aligned} \frac{\partial}{\partial A_n} P_i(k) &= \frac{\nu}{4D} (q_n q_k + r_n r_k) + \\ &\quad \frac{1}{4D} \frac{\partial \nu}{\partial B^2} \frac{\partial B^2}{\partial A_n} \sum_{l=1}^3 (q_l q_k + r_l r_k) A_l. \end{aligned} \quad (4.12)$$

It can be shown that  $\nu$  depends on  $B^2$  which depends on  $A_n$  as follows:

$$\frac{\partial B^2}{\partial A_n} = \frac{1}{2D^2} \left[ \sum_{l=1}^3 (q_l q_n + r_l r_n) A_l \right]. \quad (4.13)$$

Thus, each term of the Jacobian matrix can be calculated as follows:

$$J(n, k) = S(n, k) + \frac{2}{D\nu^2} \frac{\partial \nu}{\partial B^2} \left[ \sum_{l=1}^3 S(n, l) A_l \right] \left[ \sum_{l=1}^3 S(k, l) A_l \right]. \quad (4.14)$$

After calculation of study domain nodes magnetic vector potential, the propulsion and levitation forces can be calculated through air gap elements as follows:

$$F_{propulsion} = \frac{1}{\mu_0} \sum_l B_x B_y dl, \quad (4.15)$$

$$F_{levitation} = \frac{1}{\mu_0} \sum_l (B_x^2 - B_y^2) dl. \quad (4.16)$$

As in this chapter the computational speed is targeted, one pole pitch (out of 21) modeling of the LIM is adopted and the symmetry of the study domain is reflected in anti-periodicity boundary condition. So, the results with no end effect are considered accurate enough based on the study performed in [170], and the experimental measurements for results validation is reported in the chapter.

## 4.2.2 Linear Sparse Solver

Nonlinear time-stepped solution of FEM requires solving the system of linear equations of (4.17) in each iteration and time-step:

$$\mathbf{J}_g \mathbf{X} = (\mathbf{S}_g \mathbf{A}_i^{t+1} - \mathbf{b}_g), \quad (4.17)$$

where  $\mathbf{X}$  is the solution for the second term in right hand side of (4.9).

During the solution procedure, the matrix element values and non-zeros location change due to nonlinearity of the iron core and the movement. This change implies solution of a new system of linear equations in each iteration and time-step, which makes the linear solver critically important. Looking at the global Jacobian matrix properties, it is highly sparse and diagonally dominant since each node in the meshing is connected to only a few adjacent nodes. Furthermore, the modification of Jacobian matrix elements in rows and columns corresponding to the Dirichlet and anti-periodic boundary nodes make the Jacobian matrix asymmetric. Although the boundary nodes can be removed from the system of equations, however as the permutation of the stiffness matrix rows and columns in FPGA RAMs is time consuming due to the limitation of the number of RAMs input ports (only 2 ports in dual port RAMs), the boundary nodes are kept in the matrix.

In the proposed method, the full  $\mathbf{J}_g$  matrix is stored, which facilitates LU decomposition process, finding the fill-ins, convenient accumulation of the local matrices of each element into the global matrix and also applying the boundary condition. Furthermore, 6 small auxiliary matrices (3 associated with the upper and 3 with the lower triangular matrices) are assigned to address the non-zero matrix element indices required for sparse solution as follows:

- SNZL/SNZU addresses the first non-zero number in each column of the lower/upper triangular ( $1 \times N$ ),
- NNZL/NNZU addresses the number of non-zeros in each column of the lower/upper triangular ( $1 \times N$ ),

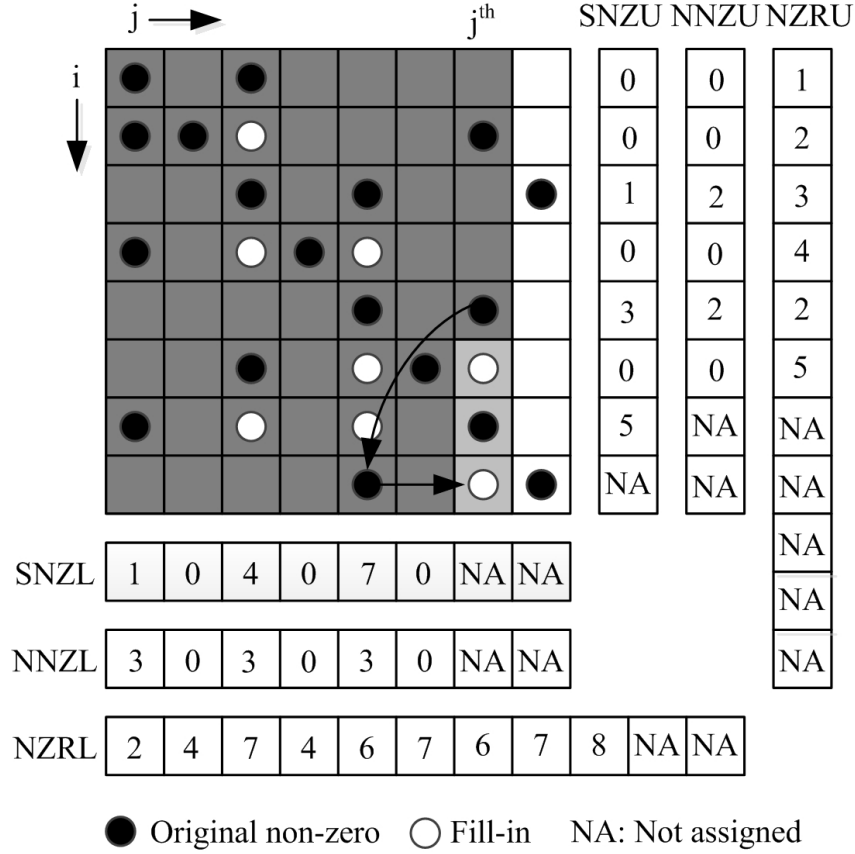


Figure 4.2: Proposed linear sparse solver scheme.

- NZRL/NZRU addresses the non-zeros row index of the lower/upper triangular ( $1 \times NNZ$ ), where  $NNZ$  is the number of non-zeros in the lower/upper triangular.

Similar to the conventional left-looking Gilbert-Peierls algorithm, the  $j^{th}$  column of  $L$  and  $U$  is computed using the already calculated columns 1 to  $(j-1)^{th}$ , which is stored on the same  $\mathbf{J}_g$  matrix. The non-zero element indices of the columns 1 to  $(j-1)^{th}$  have been already stored in the auxiliary matrices during the previous steps and can be reached quickly, required for sparse calculation of  $j^{th}$  column of  $L$  and  $U$ . Fig. 4.2 shows the proposed scheme for FEM linear sparse solver and the pseudo-code of the proposed algorithm is shown in Algorithm 1.

At the end, the  $L$  and  $U$  matrices are stored on the same  $\mathbf{J}_g$  as  $\mathbf{U} + \mathbf{L} - eye(size(\mathbf{J}_b))$  in accompanied with the sparsity pattern for sparse forward

---

**Algorithm 1** Proposed LU decomposition pseudo-code

---

```
1: for each column j do
2:   for each non-zero upper triangular element do
3:     for k=SNZL(i):SNZL(i)+NNZL(i)-1 do
4:        $\mathbf{J}_g(\text{NZRL}(k),j) = \mathbf{J}_g(\text{NZRL}(k),j) -$ 
5:          $\mathbf{J}_g(\text{NZRL}(k),i) \times \mathbf{J}_g(i,j);$ 
6:     end for
7:   end for
8:   for k=SNZL(j):SNZL(j)+NNZL(j)-1 do
9:      $\mathbf{J}_g(\text{NZRL}(k),j) = \mathbf{J}_g(\text{NZRL}(k),j) / \mathbf{J}_g(j,j);$ 
10:  end for
11: end for
```

---

elimination and back substitution.

The explained procedure is for the first iteration of the Newton-Raphson, while for the next iterations within the time-step the 6 auxiliary matrices for non-zero elements pattern remain unchanged. As a result, there is no need to *read* the zero elements from the matrix.

### 4.2.3 Hardware Emulation

FPGA hardware design can benefit from two strategies that compromise between the hardware resources and timing constraints. The first strategy is pipelining that uses an IP core for several repetitive floating-point operations with a consecutive set of input data with a specified (mostly one) clock latency. The second is parallelization that performs the floating-point operations in parallel to save time at the cost of utilizing more hardware resources. Due to high computational effort of FEM, the hardware design strategy is critical. In order to have an optimal hardware design, depending on the properties of each part the architecture is deeply pipelined to minimize the hardware resources and massively parallelized to reach an acceptable time-step.

Finite element calculations mainly consist of loops that perform a repetitive floating-point operation on a string of input data (for each node/mesh), which can be unrolled through pipelining for hardware efficiency. Furthermore, dual-port block RAMs that facilitate *read* and *write* data simultaneously are one of the main elements in the calculation path for the loops that makes pipelining

inevitable. On the other hand, inside of each loop some level of parallelism can be designed, where independent signals and RAMs are available. In addition, as some of the loops are independent of each other, they can be paralleled to reach less total latency.

The hardware architecture of system equation formation is illustrated in Fig. 4.3 to effectively show the pipelined hardware architecture design in horizontal and parallelism in vertical directions. The 32-bit single precision floating-point format (IEEE Std 754) data representation is used to accurately model the system. The latency of 20, 7, and 5 clocks is considered for divisions, additions/subtractions and multiplications respectively. Furthermore, one clock latency for reading data from the dual port block RAMs is considered. Additional registers are added in different data path in order to synchronize data for pipelining requirement. As the same calculations should be performed for each element or nodes, pipelining scheme is used through the *Cnt* counter signal. Accumulation of each mesh Jacobian matrix implies simultaneous *read* and *write* of  $3 \times 3 = 9$  elements from the Jacobian matrix and having the same calculations for each mesh. Considering the limitation of dual port RAMs that allows having access to only two elements at a time (one used for *read* and one used for *write* in accumulation procedure), 9 clock latency is required. Furthermore, the latency of 7 clocks should be considered for writing the last element to ensure availability of that node for the next mesh calculation. Then, pipelining of the input data is performed each 16 clocks.

#### 4.2.4 FPGA Implementation

The state diagram of the implemented hardware design using VHDL programming is shown in Fig. 4.4. The initialization contains loading the input data into the corresponding RAMs and signals by entering the machine geometry, material properties including the saturation curves, meshing data, machine speed, simulation time-step, supply current amplitude and frequency. The output of the hardware prototype would be magnetic vector potentials, magnetic field distribution, propulsion and levitation forces. The computation



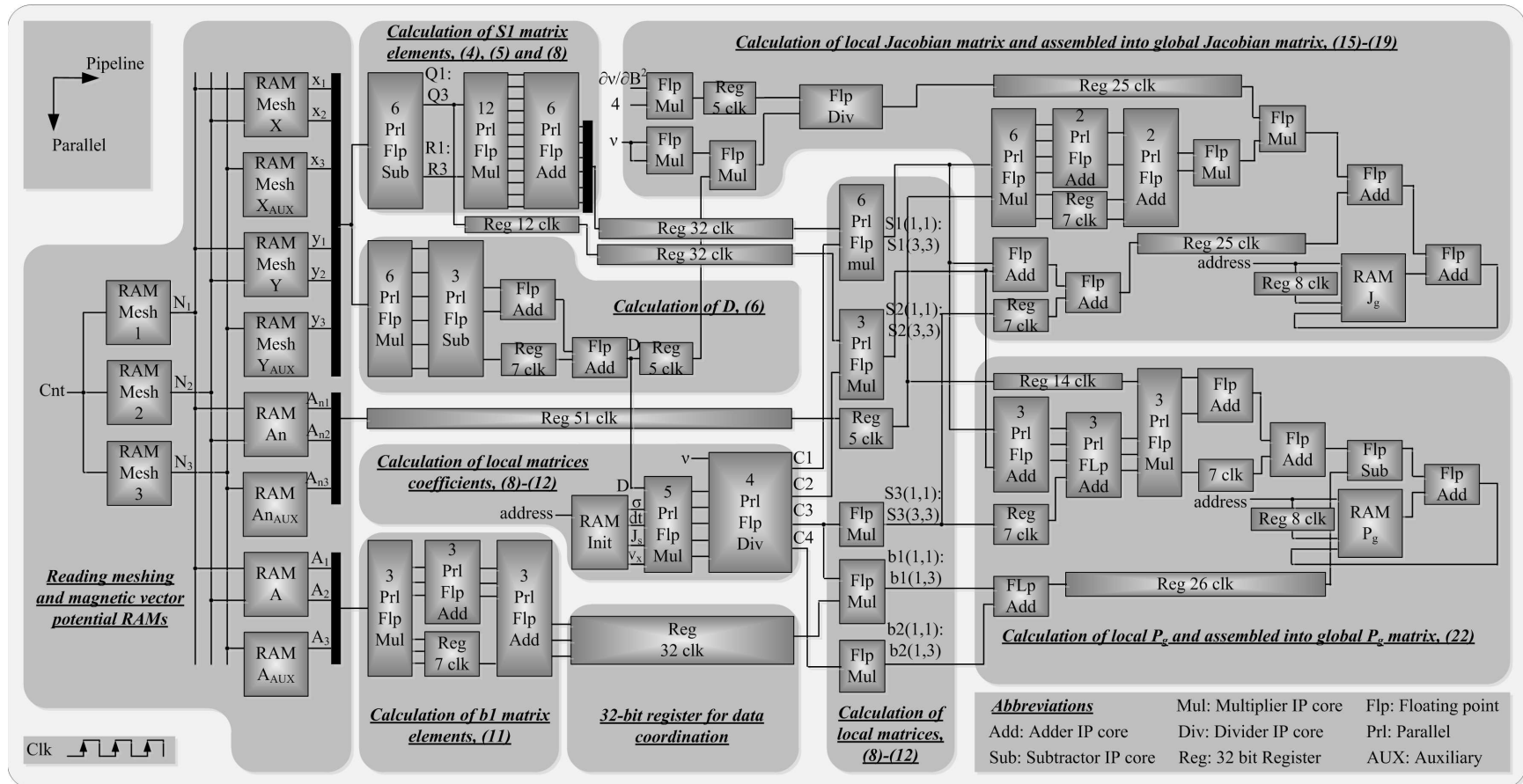


Figure 4.3: Pipelined and parallel hardware architecture of the system equation formation for the finite element emulator of LIM.

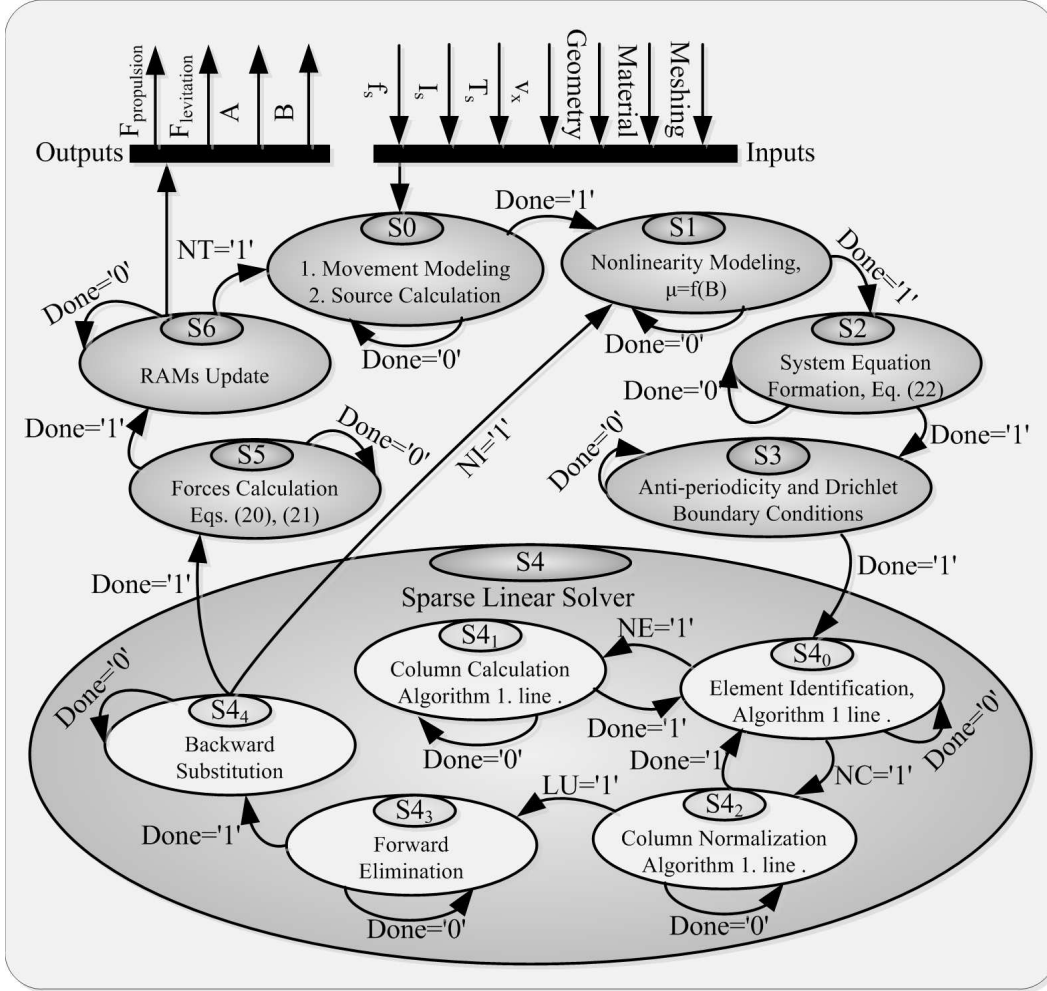


Figure 4.4: Finite state machine (FSM) of the hardware prototype

process on FPGA begins with re-meshing the study domain by updating the mesh data in parallel with assigning the current source of each element of the primary winding corresponding to the time-step in *State S0*. At the end of each state by enabling the *done* signal the process goes to the next state. In *State S1* using the saturation curve, the permeability of the iron core depending on the magnetic vector potential of the nodes is determined. Accordingly, the computation of the Jacobian matrix as well as the source vector are implemented in *State S2*. As shown in Fig. 4.1, the study domain is repeated anti-periodically in  $x$  direction and bounded in  $y$  direction. By implementation of the anti-periodicity and Dirichlet boundary conditions on the Jacobian and source matrices in *State S3*, the system of linear equations is formed.

The sparse linear solver implemented in *State S4* contains five sub-states. For the first iteration, the *StateS4<sub>0</sub>* identifies the matrix element properties. If the element is an upper triangular and non-zero, the process goes to *StateS4<sub>1</sub>* (signal *NE* is enabled) to update the column and then return to the *StateS4<sub>0</sub>*. As the computation reaches the last element of the column, the process goes to the *StateS4<sub>2</sub>* (signal *NC* is enabled) for column normalization and then return to the *StateS4<sub>0</sub>* again to evaluate the next column. The process continues until it reaches the last column of the Jacobian matrix. Afterward, the LU decomposition is performed (signal *LU* is enabled) followed by the forward elimination (*StateS4<sub>3</sub>*) and back substitution (*StateS4<sub>4</sub>*) states respectively. For the next iterations the process is similar, however the matrix element properties are already identified in the first iteration and stored in the auxiliary matrices. Meanwhile, if the result of the Newton-Raphson is converged, the process goes to the force calculation followed by the RAMs reset for the next time-step (signal *NT* enabled), otherwise the process goes to *StateS1* (signal *NI* enabled) to re-evaluate the Jacobian and source matrices based on the permeability.

The VHDL program was compiled and synthesized in the Xilinx ISE<sup>®</sup> software and mapped into the targeted FPGA device (Virtex 7 XC7VX485T).

### 4.3 Results, Verification and Discussion

The single-sided short primary LIM used in this study is an industrial prototype for maglev-based transportation system for contactless operation in urban areas. The specification of the LIM is presented in Appendix A. The FPGA-based 2-D FEM hardware prototype of the LIM has been evaluated based on the machine speed of 26.82m/s, supply frequency of 92Hz and primary mover phase current of  $1541A_{RMS}$ . A comparison between the hardware prototype results and the JMAG-Designer<sup>®</sup> is provided in Fig. 4.5, where the magnetic flux density vector, current density (the source and eddy currents) and the magnetic flux density distribution are plotted for results verification.

The percentage of error between the hardware prototype and JMAG-Designer<sup>®</sup>

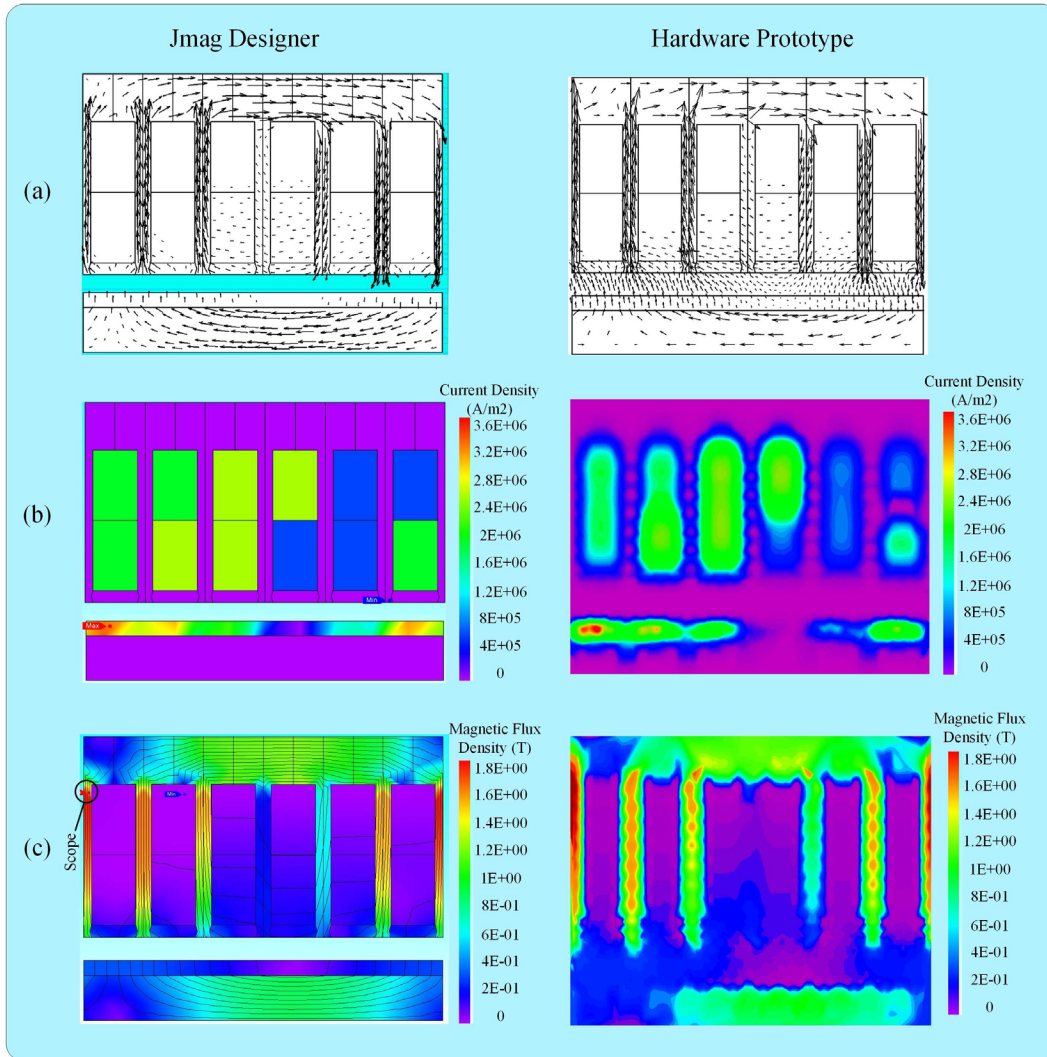


Figure 4.5: Hardware prototype results comparison with JMAG-Designer<sup>®</sup> (a) magnetic flux density vector (b) current density (c) magnetic flux density distribution.

for magnetic flux density of nodes all over the study domain and the eddy current density in the aluminum sheet are evaluated as follows:

$$\frac{\| B_{Hardware} - B_{JMAG} \|}{\| B_{Hardware} \|} \times 100 = 1.97\%, \quad (4.18)$$

$$\frac{\| J_{Hardware} - J_{JMAG} \|}{\| J_{Hardware} \|} \times 100 = 2.64\%. \quad (4.19)$$

The results of the hardware prototype and the JMAG-Designer<sup>®</sup> are in a good agreement. The minor deviations of Fig. 4.5 are due to the plotting tool of the hardware prototype results and the following technical considerations:

- The JMAG-Designer<sup>®</sup> uses the incomplete Cholesky conjugate gradient (ICCG) as the linear solver, which is an iterative solver resulting in an approximate solution. However in the hardware prototype a new sparse linear solver is proposed appropriate for FPGA hardware architecture based on Gilbert-Peierls algorithm for LU decomposition, which is a direct solver resulting in an exact solution.
- JMAG-Designer<sup>®</sup> uses 64-bit double precision floating point number representation, however in the hardware prototype 32-bit single precision floating point number representation is used for hardware efficiency while keeping the accuracy in an acceptable range.

In this study, the same meshing based on the commonly used first order triangular elements is utilized for both proposed hardware prototype and Jmag-Designer with 547 nodes and 986 elements. It should be mentioned that as the problem size increases, e.g. by 3-D modeling or full structure modeling, the number of floating point operations increases accordingly. As a result, the negligible truncation made by the 32-bit number representation accumulates through the floating point computations, although for most practical cases this point will not affect the results considerably. In the case of FEM analysis of an electric machine for the same case study, 3-D modeling as well as full structure modeling and having finer mesh result in higher accuracy that offset the round off error accumulation of the 32-bit number representation. Furthermore, for very large size 3-D models, the double precision 64-bit floating point number

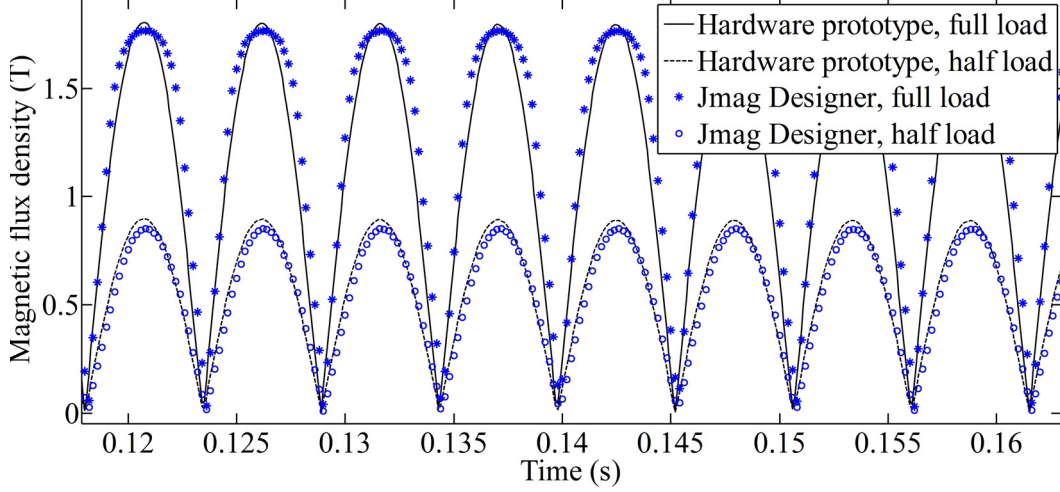


Figure 4.6: Nodal magnetic flux density of the hardware prototype and JMAG-Designer<sup>®</sup> in the highest field intensity region shown by scope in Fig. 4.5.

representation can be used to avoid this deviation at the cost of more hardware resource.

The nodal magnetic field in the highest magnetic field intensity region that is shown by “Scope” in Fig. 4.5 is presented in Fig. 4.6 for both hardware prototype and JMAG-Designer<sup>®</sup> with respect to time for full and half load currents. It can be seen that the results are in a good agreement. Further comparison is made by the LIM propulsion and levitation forces with respect to slip frequency with the machine speed of 26.82m/s with full and half load currents for both FPGA-based hardware prototype and JMAG-Designer<sup>®</sup> as shown in Fig. 4.7. The primary mover full load phase current of  $1541A_{RMS}$  is divided between four parallel circuits and distributed throughout the machine. Based on Fig. 4.7, the forces change with primary mover frequency, and the maximum achievable propulsion and levitation forces are 30 kN and 70 kN, respectively. Noting that the torque of a rotary induction machine is equivalent to the propulsion force in a LIM, the torque vs. slip and the propulsion force vs. slip frequency curves for a rotary induction motor and a LIM, respectively, confirm each other. The slip frequency in a linear induction motor can be expressed as:

$$sf_s = f_s - \frac{1}{2h}v_x. \quad (4.20)$$



where  $s$  is the slip,  $f_s$  is the supply frequency and  $h$  is the pole pitch length. The slip frequency ( $sf_s$ ) becomes zero when the supply frequency becomes  $f_s = \frac{1}{2h}v_x$ , which is the synchronous frequency corresponding to the machine speed. As can be seen in Fig. 4.7, in this situation the secondary aluminum sheet and the primary winding magnetic fields are in the same phase resulting in maximum levitation force in the negative direction meaning a high attraction force between the primary and secondary. In addition, there is negligible propulsion force (50 N) arisen from the end effect, which does not exist in a rotary induction machine. In [171], it is shown the end effect force can be obtained as a function of goodness factor (67.78 for the studied LIM), number of poles, and slip. Ideally LIMs can be designed to have zero propulsion force at zero slip using optimized goodness factor, which implies having no end effect force at synchronous frequency. As the supply frequency goes higher than the synchronous frequency determined by the control system, the levitation force decreases while it creates a positive propulsion that operates the machine in the motoring mode until the levitation force reaches zero. Further increase in the supply frequency leads to a positive levitation force causing a repulsion between the primary and secondary, where a LIM is not supposed to conventionally operate. In reverse, as the supply frequency goes lower than the synchronous frequency with the same speed, the kinetic energy stored in the machine can generate power as regenerative braking. A guideway test track is built to ensure safe testing the LIM in a wide range of speeds. The test rig is equipped with load cells to measure the propulsion and levitation forces, and the measurements are recorded in a data bucket. Alternatively, the propulsion force can be found by measurement of the input power subtracted by the losses and the linear speed using  $P = F_{propulsion} \times V$ . Furthermore, in order to keep the LIM stable and have a frictionless air gap fixed at 10 mm, an air gap sensor is used to control the levitation force with the frequency of 10kHz. In Fig. 4.7, a set of experimental results are provided for the slip frequency range between 6 to 14 Hz (or supply frequency range between 94 to 102 Hz), where the machine is normally operated and practical consideration such as tolerable mechanical forces make the measurements feasible. The propulsion and

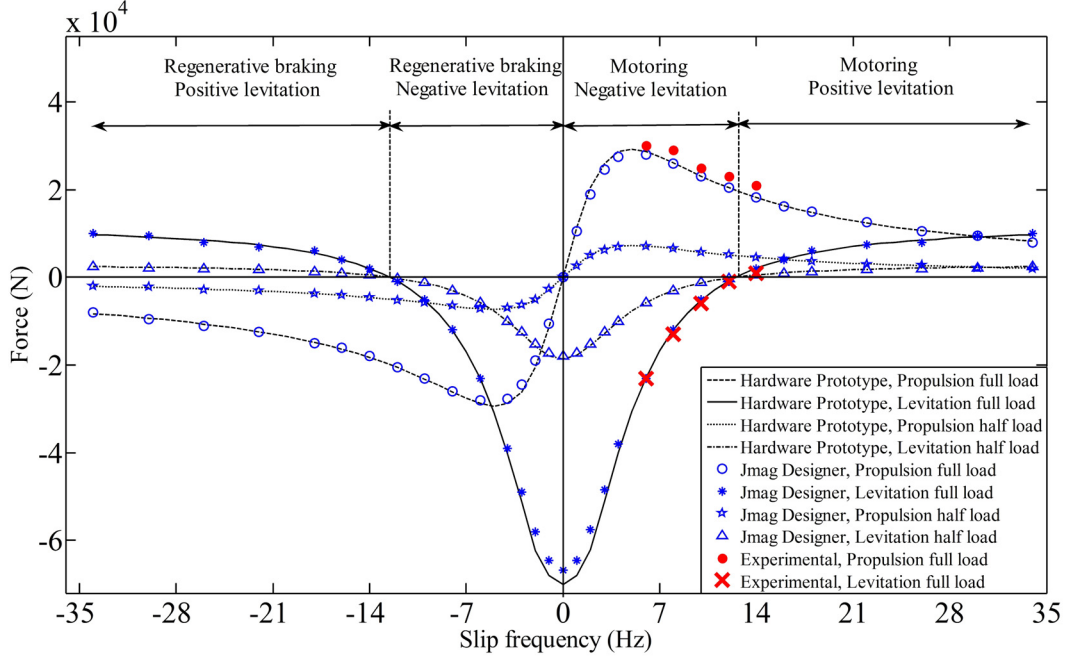


Figure 4.7: LIM propulsion and levitation forces vs. slip frequency

levitation forces at a specific slip (negative) in the regenerative braking mode have the same amplitude of those at the same slip (positive) in the motoring mode. It can be seen that the experimental results are in a good agreement with both hardware prototype and JMAG-Designer<sup>®</sup> and the agreement of the results for the specified slip frequency range shows the correctness of the results for the entire frequency range. The small differences arise from practical inaccuracies in measurement of forces, neglecting the end effect and inherent approximations associated with FEM modeling. As there is no considerable deviation between the hardware prototype and JMAG-Designer<sup>®</sup>, both simulations satisfy a close agreement with the experimental results. Figs. 4.6 and 4.7 are shown under full and half load currents to validate the hardware prototype results. In addition, it can be observed that the magnetic field changes proportionally with the primary current, while the propulsion and levitation forces change proportionally to square of the current.

In the hardware design, the number of clock latencies between the input and output, as well as the FPGA clock frequency affect the emulation time-step. Table 5.2 shows the latencies corresponding to each state in the finite



state machine of Fig. 4.4, as the Newton-Raphson method converges in three iterations with a tolerance of  $10^{-3}$ . Since the LIM structure contains moving part, the meshing is different in each time-step which in turn changes the Jacobian matrix in the linear system of equations of (4.17). As the nonlinearity of the iron core is considered, the Jacobian matrix elements are changed, while the position of the non-zero elements does not change because the meshing is the same within the time-step. Therefore, the movement modeling state is performed at the beginning of each time-step while the force calculation and RAM reset states are performed after the Newton-Raphson loop results have converged at the end of each time-step. In rest of the states, the clock latency of the second and third iterations are twice of the clock latency of the first iteration, except the element identification state in which the auxiliary matrices for the sparse linear solver are determined only in the first iteration and used in the second and third iterations. It can be seen that the need for the solution of the system of linear equation in each time-step and each iteration is the main computational burden of finite element calculation in moving structures with nonlinearity.

Table 4.1: Latencies for Each State of the Finite State Machine of Fig. 4.4

<b>State</b>	<b>1<sup>st</sup> iteration</b>	<b>2<sup>nd</sup> and 3<sup>rd</sup> iterations</b>
S0) Movement Modeling		512
S1) Nonlinearity Modeling	434	434×2
S2) System Equation	16,443	16,443×2
S3) Boundary Condition	76,023	76,023×2
S4) Sparse Linear Solver	3,548,564	2,477,019×2
-S4 <sub>0</sub> ) <i>Element Identification</i>	1,071,545	0
-S4 <sub>1</sub> ) <i>Column Calculation</i>	2,352,627	2,352,627×2
-S4 <sub>2</sub> ) <i>Column Normalization</i>	58,328	58,328×2
-S4 <sub>3</sub> ) <i>Forward Elimination</i>	33,543	33,543×2
-S4 <sub>4</sub> ) <i>Backward Substitution</i>	32,521	32,521×2
S5) Force Calculation		739
S6) RAMs Reset		21,173
<b>Total</b>		<b>8,803,726 Clks</b>

From the latency results of Table 5.2 the maximum FPGA clock frequency of 178.9MHz (5.59ns) is obtained after the design mapping, and the overall latency of each time-step computation considering the number of clock latencies

of the design would be 49.2ms. Execution times of 1 sec emulation of the LIM operation with the simulation time-step range from  $10\mu s$  to 1ms are shown in Table 4.2 with the hardware prototype on FPGA and the JMAG-Designer<sup>®</sup> on a PC with Intel Core i7-2600 CPU at 3.4GHz. Consequently, the proposed hardware prototype proposes average 9.73 times speed-up in comparison with the same mesh size and the same inputs of JMAG-Designer<sup>®</sup> with per phase current of  $1541A_{RMS}$ , frequency range of 53-123 Hz and machine speed of 26.82m/s.

Table 4.2: Execution time latencies of the JMAG-Designer<sup>®</sup> and Hardware prototype

Time step (s)	JMAG-Designer <sup>®</sup> (s)	Hardware prototype (s)	Speed-up	Error (%)
$1e^{-3}$	480	49.2	9.75	1.97
$5e^{-4}$	960	97.9	9.80	1.83
$1e^{-4}$	4,740	487.4	9.72	1.80
$5e^{-5}$	9,450	974.0	9.70	1.80
$1e^{-5}$	47,320	4,870.3	9.71	1.77

The hardware prototype platform is Xilinx<sup>®</sup> Virtex 7 XC7VX485T FPGA with available 37,080 kb of BRAM, 607200 slice registers, 303600 slice LUTs, 75900 slices and 700 bonded IOBs. The hardware resource utilization in terms of the number of logic resources and the occupied percentage is summarized in Table 4.3. As a result of the proposed deeply pipelined architecture, it can be seen that the massive computation of the case study can be fitted into the targeted device with a maximum resource utilization of 67%.

Table 4.3: Hardware Resource Utilization

Resources	Utilization	
	Number	Percentage
BRAM	12,742 kb	34%
Slice Registers	24,294	4%
Slice LUTs	27,291	8%
Occupied Slices	8,741	11%
LUT-FF Pairs	20,059	67%
Bonded IOBs	83	11%

Due to the limitation of BRAM resource for storage of the global Jaco-

bian matrix as well as the auxiliary matrices, our implementation is targeted for medium sized FEM problems with node numbers in the order of 1000. However, extension of the proposed methodology to larger sized problems is achievable through external RAMs or hardware efficient strategies including partial reconfiguration or compressed storage of non-zeros. It is worth noting that by utilizing pipelined hardware design scheme, although increasing the number of nodes/elements affects the execution time, the FPGA hardware resource utilization except the BRAM is not considerably affected.

The speed-up achieved with the proposed FEM hardware prototype highlights the application of the proposed methodology for electric machine design optimization process that requires numerous cycle of testing. In order to facilitate an automatic design optimization, the design variables coming out of a multi-dimensional search space are interfaced with FPGA I/O pins to be updated in every iteration of the design procedure, while the FEM-based hardware architecture of the FPGA is remained unchanged.

## 4.4 Summary

FEM model is widely used for accurate design and analysis of electric machines; however, it suffers from long execution time. In this chapter, for the first time hardware acceleration of 2-D FEM for a single-sided linear induction motor (SLIM) on FPGA is proposed. The nonlinearity of the iron core as well as the movement are taken into consideration. A new sparse solver is proposed based on left looking Gilbert-Peierls algorithm for the system of linear equations of FEM that need to be solved in different iterations and time-steps. Implementation of the model is performed in a massively paralleled and deeply pipelined hardware architecture using VHDL coding with single precision floating-point number representation. The proposed emulation was performed at various time-steps resulting in significant average speed-up of 9.73 times in comparison with JMAG-Designer<sup>®</sup> as a commercial finite element software, and the overall hardware latency of each time-step for the emulation was 49.2ms in average with minimum achievable FPGA clock of 5.59ns.

# Chapter 5

## Real-Time Finite Element Method Model of Linear Induction Motor

### 5.1 Introduction

Traditional off-line simulations require modeling of all components of the system, which suffers from degradation of accuracy proposed by the model simplification assumptions of each component. In addition, off-line simulations are often time consuming with accurate and complex models, e.g. FEM. HIL simulation provides real-time data transfer between the emulated component on hardware and the interacted actual device under test to avoid inaccurate modeling of the device under test resulting in fast and accurate prototyping.

All real-time HIL studies for electric machines are performed with simplified models applicable for limited application including equivalent circuit [103, 172],  $q-d$  [69, 74, 173–177], analytical space harmonic [178], and magnetic equivalent circuit [42] models. Specifically, [175–177] considered nonlinearity in the  $q-d$  model by off-line FEM pre-calculation of nonlinear inductances as a function of a multi-dimensional set of inputs. However, due to continuous change of each input variable in its corresponding range, the pre-processing unit is time consuming. The inefficiency of the method is not only due to requiring a large pre-processing stage but also with the change of machine structure the massive pre-calculations should be carried out again. In this chapter, FEM-based computation in real-time for HIL emulation of electric

machines is targeted with minimized pre-calculation for detailed analysis, accurate testing, and design purposes.

The computational intensity of nonlinear time-stepped FEM analysis of electric machines arises from the change of the FEM system of equations in every iteration and time-step due to their nonlinearity and movement respectively, which makes real-time simulation very challenging. The TLM method proposes to decouple the nonlinear elements of a coupled network through time delayed transmission lines with specified characteristic impedance in order to keep the FEM stiffness matrix unchanged during nonlinear iterations. The method results in considerable reduction of time especially for static problems since the LU decomposition of the global stiffness matrix needs to be carried out only once per simulation and the nonlinearity are solved independently through decoupled elements.

The TLM method was initially proposed for wave scattering problems in [179]. Afterwards it was used for circuit analysis with linear and nonlinear elements [180]. Later, based on the analogy of the nodal admittance matrix in an electric circuit and the stiffness matrix of FEM, the TLM method was applied on magnetostatic [181] and magnetodynamic problems [182], and further improved for quasi-static field analysis [183], all of them for non-moving structures. In [146] the TLM method was applied on moving structure of an induction machine for FEM computation on CPU, where it is shown that the TLM itself is not significantly faster than the conventional Newton-Raphson (N-R) method for two main reasons: first, movement structure of electric machines, which results in change of the global stiffness matrix for each time which requires LU decomposition of the matrix at the beginning of each time-step; and second, solution of numerous independent nonlinear elements for fairly large number of TLM iterations in comparison with conventional N-R iterations due to mismatch of the chosen TLM line characteristic impedance and the actual nonlinear element impedance.

For the first time, this chapter proposes the following solutions to overcome the problem of low speed execution of FEM problem with moving structures to enforce real-time execution:

- A novel RT-TLM method based on *finite pre-calculated LU decompositions* to avoid performing LU decomposition of the global stiffness matrix in each time-step.
- FPGA-based hardware implementation to exploit the TLM method for potential parallelism.

The chapter is organized to first present the problem formulation when TLM is applied on FEM, then the novel methodology is proposed to overcome the bottlenecks to achieve real-time execution in Section II. In Section III, the proposed methodology is implemented on Xilinx Virtex Ultrascale+ FPGA with deeply pipelined and massively paralleled hardware architecture for HIL scenario. Finally, in Section IV the results of real-time simulation are presented, validated and discussed to show the effectiveness of the proposed methodology.

## 5.2 Novel RT-TLM Method for Real-Time Finite Element Modeling Emulation

### 5.2.1 FEM Element Equivalent Circuit

The Maxwell's equation representing an eddy current problem can be expressed as:

$$\nabla \cdot \nu \nabla A - \sigma \frac{\partial A}{\partial t} + J_s = 0, \quad (5.1)$$

where  $A$  is the 2-D magnetic vector potential in  $z$  direction of the corresponding element. The matrix form of (5.1) can be simplified into the system of nonlinear equations of:

$$\mathbf{G}(\mathbf{A})\mathbf{A} + \mathbf{C} \frac{\partial \mathbf{A}}{\partial t} = \mathbf{I}, \quad (5.2)$$

where  $\mathbf{S} = \mathbf{G} + \mathbf{C} \frac{\partial}{\partial t}$  is called the finite element stiffness matrix, and the global matrices of conductance ( $\mathbf{G}$ ), capacitance ( $\mathbf{C}$ ) and current ( $\mathbf{I}$ ) require accumulation of the following matrices for each element:

$$\mathbf{G}^{(e)} = \frac{\nu^{(e)}}{4\Delta^{(e)}} \cdot \begin{pmatrix} q_1q_1 + r_1r_1 & q_1q_2 + r_1r_2 & q_1q_3 + r_1r_3 \\ q_2q_1 + r_2r_1 & q_2q_2 + r_2r_2 & q_2q_3 + r_2r_3 \\ q_3q_1 + r_3r_1 & q_3q_2 + r_3r_2 & q_3q_3 + r_3r_3 \end{pmatrix}, \quad (5.3)$$

$$\mathbf{C}^{(e)} = \frac{\sigma^{(e)} \Delta^{(e)}}{12} \cdot \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \quad (5.4)$$

$$\mathbf{I}^{(e)} = \frac{J_s^{(e)} \Delta^{(e)}}{3} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad (5.5)$$

where  $q_1 = y_2 - y_3$ ,  $r_1 = x_3 - x_2$  and  $q_2, q_3, r_2, r_3$  can be calculated by cyclic permutation of the indices. It can be seen that each element represents an electrical circuit of connected nonlinear conductances and capacitors as shown in Fig. 5.1 (a) with the following values:

$$G_{ii'}^{(e)} = -\frac{\nu^{(e)}}{4\Delta^{(e)}} \cdot (q_i q_{i'} + r_i r_{i'}), \quad (5.6)$$

$$C_{ii'}^{(e)} = -\frac{\sigma^{(e)} \Delta^{(e)}}{12}, \quad (5.7)$$

$$C_{i0}^{(e)} = 4\frac{\sigma^{(e)} \Delta^{(e)}}{12}, \quad (5.8)$$

$$I_i^{(e)} = \frac{J_s^{(e)} \Delta^{(e)}}{3}, \quad (5.9)$$

where  $i, i'=1,2,3$  and  $i \neq i'$  for each element ( $e$ ).

The TLM method uses lossless time delayed transmission lines, named TLM links, to decouple the nonlinear elements from the linear network to keep the stiffness matrix unchanged within each time-step as shown in Fig. 5.1 (b). The capacitors are also connected through the transmission lines for discretization in time domain. The transmission line characteristic impedances/admittances ( $Z_0/Y_0$ ) can be chosen arbitrarily; however practically it is advantageous to determine them as close as possible to the corresponding nonlinear element impedance for faster convergence of the TLM iterations.

The TLM method is based on traveling incident waves through transmission lines and calculation of the reflected waves at the sending end (network side) and the receiving end (element side) due to mismatch of the route characteristic impedances illustrated in lattice diagram of Fig. 5.1 (c). Considering the  $k^{th}$  TLM iteration of the  $n^{th}$  time-step, the magnetic vector potential ( ${}^n_k \mathbf{A}_S$ ) and current ( ${}^n_k \mathbf{I}_S$ ) at the sending end are represented according to transmission line theory as:

$${}^n_k \mathbf{A}_S = {}^n_k \mathbf{A}_S^i + {}^n_k \mathbf{A}_S^r, \quad (5.10)$$

$${}^n\mathbf{I}_S = ({}^n\mathbf{A}_S^r - {}^n\mathbf{A}_S^i)/Z_0, \quad (5.11)$$

where  ${}^n\mathbf{A}_S^i$  and  ${}^n\mathbf{A}_S^r$  are the incident and reflected magnetic vector potentials at the sending end. The reflected waves from the sending end travel toward the receiving end without change since the transmission lines are lossless. Then, the reflected waves of the sending end become the incident waves of the receiving end and reflected as the next incident waves of the sending end. So, the magnetic vector potential ( ${}^n\mathbf{A}_R$ ) and current ( ${}^n\mathbf{I}_R$ ) at the receiving end can be extracted based on incident and reflected waves of the sending end as follows:

$${}^n\mathbf{A}_R = {}^n\mathbf{A}_S^r + {}^{n+1}\mathbf{A}_S^i, \quad (5.12)$$

$${}^n\mathbf{I}_R = ({}^n\mathbf{A}_S^r - {}^{n+1}\mathbf{A}_S^i)/Z_0, \quad (5.13)$$

where  ${}^{n+1}\mathbf{A}_S^i$  is the next incident magnetic vector potential of the sending end.

The TLM procedure begins with using the incident magnetic vector potential in the calculation of the current source of Norton equivalent circuit ( $\mathbf{I}_n = 2{}^n\mathbf{A}_S^i$ ) of the TLM sections. Afterward, the system of linear equations is solved for the linear TLM equivalent circuit of Fig. 5.1 (d) to find the nodal magnetic vector potentials at the sending end as follows:

$$[\mathbf{Y}_0][{}^n\mathbf{A}_S] = [\mathbf{I}_n({}^n\mathbf{A}_S^i)] + [\mathbf{I}], \quad (5.14)$$

where  $\mathbf{Y}_0$  is the admittance matrix of the TLM network, dependent on the characteristic admittance of the transmission lines as follows:

$$Y_{0G_{ii'}}^{(e)} = -\frac{\nu_{TLM}^{(e)}}{4\Delta^{(e)}} \cdot (q_i q_{i'} + r_i r_{i'}), \quad (5.15)$$

$$Y_{0C_{ii'}}^{(e)} = -\frac{\sigma^{(e)} \Delta^{(e)}}{6T_{TLM}}, \quad (5.16)$$

$$Y_{0C_{i0}}^{(e)} = 4\frac{\sigma^{(e)} \Delta^{(e)}}{6T_{TLM}}, \quad (5.17)$$

where  $\nu_{TLM}^{(e)}$  is the arbitrarily reluctivity chosen for the transmission lines of the corresponding element and  $T_{TLM}$  is the time delay of all transmission lines in the study domain. As the  $\mathbf{Y}_0$  is fixed within each time-step, the LU decomposition is performed only once for the first TLM iteration and for the next



iterations only the forward elimination and backward substitution are required. The process continues with calculation of the sending end reflected magnetic vector potential ( ${}^n_k \mathbf{A}_S^r$ ) based on transmission line theory of (5.10) and (5.11). The reflected waves of the sending end travel toward the receiving end. At the receiving end as shown in Fig. 5.1 (e), the 3 nonlinear conductances of each element  $G_{ii'}^{(e)}$  are still coupled due to the dependency of the element reluctivity to element vertex magnetic vector potentials, which requires solution of a  $3 \times 3$  system of nonlinear equations of each element ( $e$ ) in order to find the next incident magnetic vector potentials as follows:

$$Y_{0G_{ii'}}^{(e)} ({}^n_k A_{G_{ii'}}^r - {}^n_{k+1} A_{G_{ii'}}^i) = G_{ii'}^{(e)} ({}^n_k A_{G_{12}}^{(e)}, {}^n_k A_{G_{23}}^{(e)}, {}^n_k A_{G_{31}}^{(e)}) ({}^n_k A_{G_{ii'}}^r + {}^n_{k+1} A_{G_{ii'}}^i), \quad (5.18)$$

As the nonlinear solution at the receiving end using conventional N-R method is reached, the TLM iterations will be unconditionally converged when the incident magnetic vector potential at the sending end of all transmission lines becomes stable.

## 5.2.2 Strategies for Enforcing Real-Time Execution

### a) Finite Pre-Calculated LU decompositions for moving structures

The TLM-based FEM simulation of electric machines still requires LU decomposition of the FEM stiffness matrix for the number of simulation time-steps due to movement. Since in the TLM method the characteristic admittance of the linear network can be arbitrarily chosen for all time-steps with infinite combination of frequency, current/voltage amplitude, and machine speed, the remaining factor that changes the FEM stiffness matrix is the movement. During movement, finite number of relative position of the moving part to the stationary part can be considered as shown in Fig. 5.2 for a linear induction motor as the case study, resulting in finite number of stiffness matrices corresponding to the positions. Therefore, for the first time *finite pre-calculated LU decompositions* is proposed to be carried out by a small pre-processing unit to be used in the time-stepping procedure. The proposed methodology enables

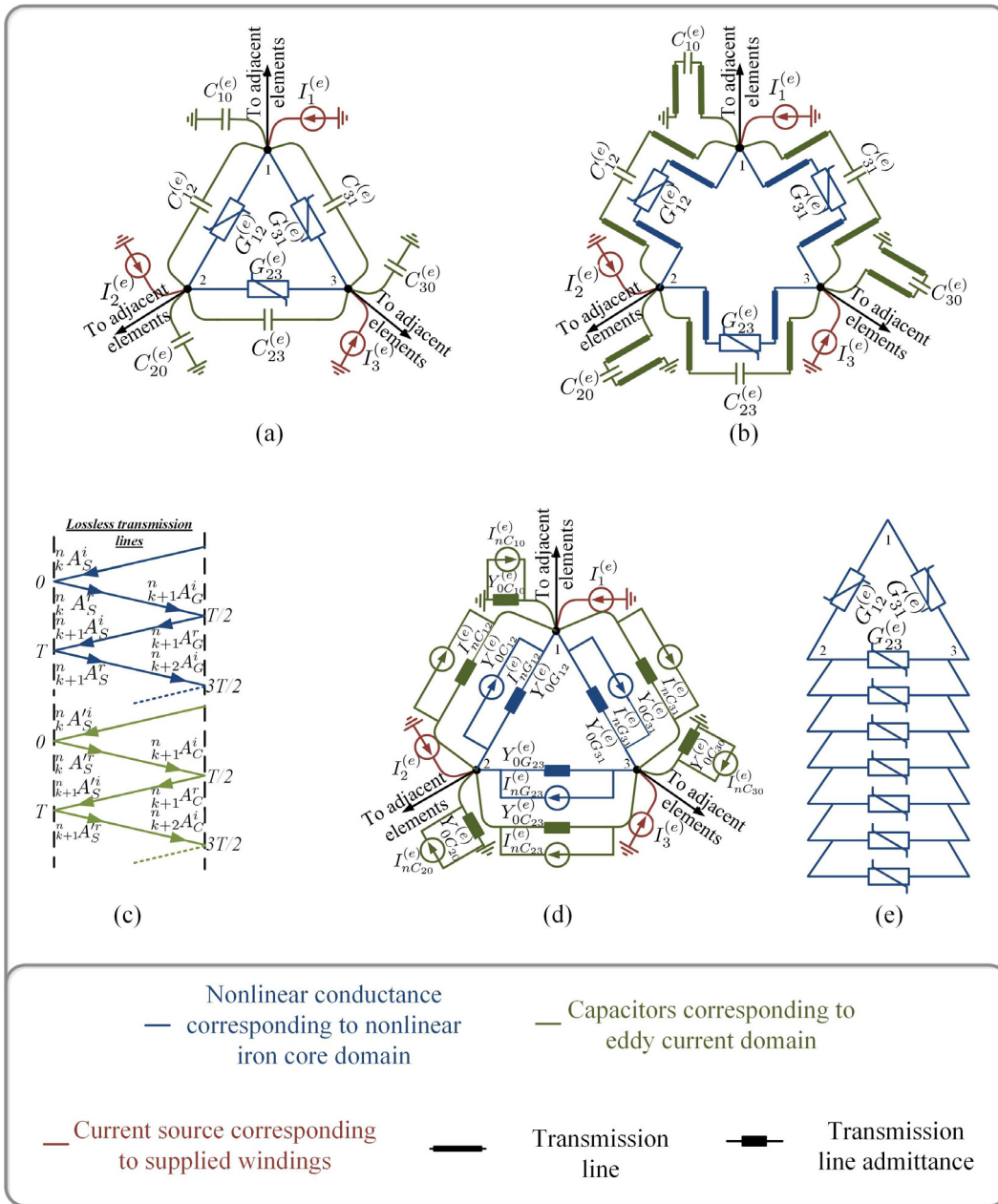


Figure 5.1: The TLM application procedure on an eddy current FEM element. (a) FEM element equivalent circuit, (b) Transmission line decoupling, (c) Lattice diagram of the transient incident and reflected potential waves, (d) TLM equivalent circuit, (e) Decoupled nonlinear conductances.

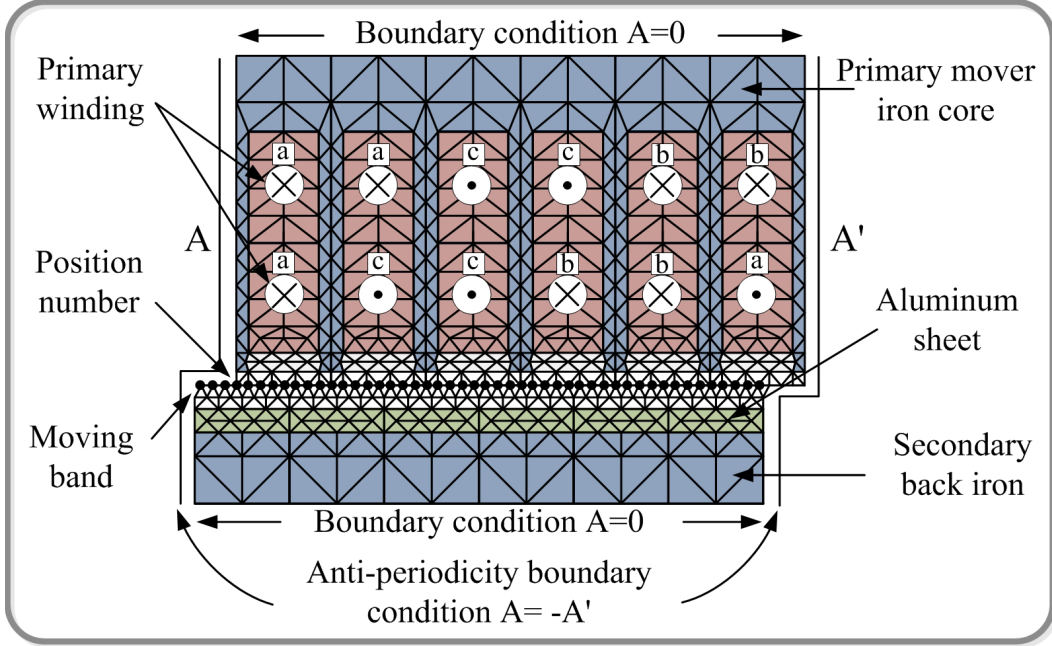


Figure 5.2: One pole pitch structure of the linear induction motor case study.

achieving real-time execution by eliminating the TLM-based FEM computational bottleneck, i.e. the solver, and then only forward elimination and backward substitutions of each TLM iteration are required to be processed.

The conventional moving band technique is used and depending on the position number, high quality elements in terms of size and equilateral triangle elements in the air gap have been re-meshed corresponding to that position number.

### b) Parallelism exploitation

Another distinct feature of TLM is parallelism, which has not been well promoted. Although the conventional TLM method decomposes a large coupled nonlinear network into small decoupled elements; however, the solution of numerous decoupled nonlinear equations for the number of TLM iterations is not highly efficient on sequential processors. The parallelism capability of the decoupled equations to calculate the next incident magnetic vector potentials of (5.18) can be exploited by proposing FPGA hardware implementation.

On the other hand, the sparse forward elimination and backward substitu-

tions are performed in a massively paralleled manner with respect to rows and sequential with respect to columns, which provides another avenue for parallelism exploitation on FPGA. Utilizing massive parallelism in the two highest computationally burdensome stages enforces the execution of the TLM-based FEM in real-time.

## 5.3 FPGA Design Architecture for the Novel RT-TLM FEM Emulation

### 5.3.1 Hardware Architecture

An optimal hardware design pursues four objectives, i.e. minimize number of clocks, maximize achievable clock frequency, minimize hardware resources and maximize accuracy, which are generally in contradiction with each other. The massive computational logic blocks (CLBs) of FPGA can be either configured in parallel where a number of synchronized independent data operations are needed, or exposed to pipelined data where numerous and repetitive independent operations can be performed with minimum hardware resource usually having RAMs in the data path. The application of TLM on FEM efficiently increases the level of parallelism in the computational algorithm, which is made of massive independent repetitive operations for each node/element.

Fig. 5.3 shows the FPGA hardware design architecture for the next incident potential computation, forward elimination and backward substitution. It is illustrated that the hardware is massively paralleled in vertical and the data is deeply pipelined in horizontal directions. The pipelined data of the signals are constantly changing in each clock to perform the calculations for each element of FEM meshing. As in a pipelining scheme the number of pipelined data is significantly higher than the latency of the IP cores between the inputs and outputs, maximum achievable clock frequency is targeted by setting maximum latencies of 28 clocks for divisions/square root, 12 clocks for additions/subtractions, 8 clocks for multiplications, 2 clocks for comparisons, 1 clock for RAMs *read/write* and 1 clock for multiplexers. The IP cores for floating operations in Fig. 5.3 are scaled based on the assigned latencies ex-

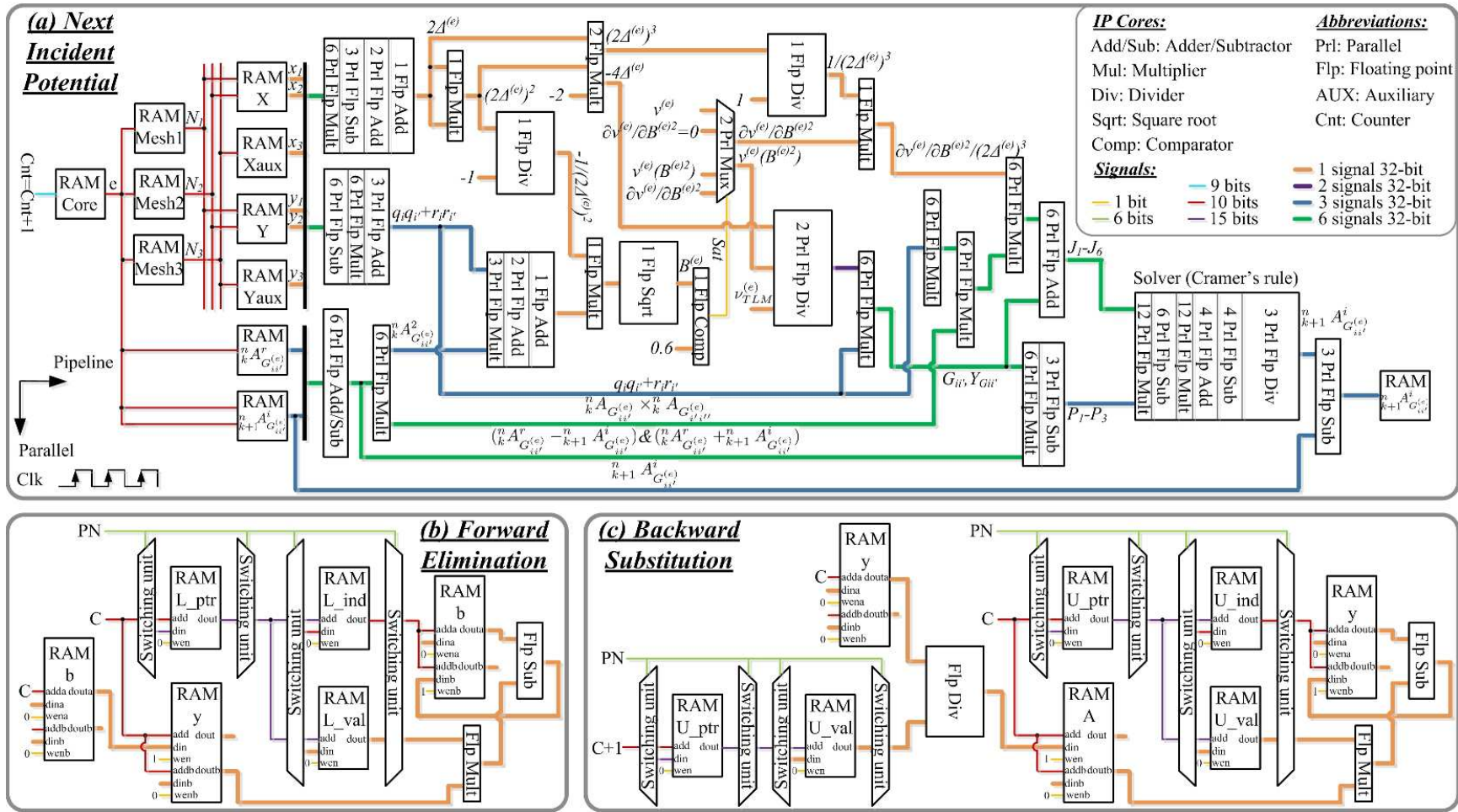


Figure 5.3: Massively paralleled and deeply pipelined hardware architecture. (a) The next incident potential, (b) Forward elimination and (c) Backward substitution.

cept the RAMs and Mux. The 32-bit floating point single-precision format IP core architectures are optimized to achieve maximum frequency to drive the FPGA with non-blocking mode of flow control using only logics without usage of DSP slices.

#### **a) Calculation of the next incident potentials**

For each time-step of simulation, the  $3 \times 3$  system of nonlinear equations to calculate the next incident potentials at the receiving end, is located inside of three cascaded loops, i.e. the nonlinear iron core elements, the NR iterations and the TLM iterations. Therefore, the hardware architecture of the solver becomes critically important. Among various solver algorithms, the Cramer's rule is implemented due to its inherent parallelism, which requires computation of 4 independent  $3 \times 3$  determinants, and provides better efficiency for small system of linear equations. The nonlinearity in elemental level is treated by conventional N-R solution. The deeply pipelined and parallelized hardware is shown in Fig. 5.3 (a).

#### **b) Sparse forward elimination and backward substitution**

Performing sparse forward elimination followed by sparse backward substitution is required for the number of TLM iterations in each time-step in order to compute the magnetic vector potential of the nodes at the sending end. Fig. 5.3 (b) and (c) shows the schematic of pipelined data in the sparse forward elimination and backward substitution for the first column elimination and last column substitution, respectively, and the subsequent columns of  $L$  and  $U$  matrices are treated similarly. Due to limitation of RAMs number of ports (two for dual port RAMs) the pipelining hardware design strategy is used to read the data from the first port and write the updated data through the second port.

The stationary edge of the moving band is divided into 48 positions and for each position the LU decomposition of the corresponding FEM stiffness matrix is stored. For efficient RAMs utilization, only the nonzero elements of the finite pre-calculated decomposed  $L$  and  $U$  are stored with the sparsity pattern using



the compressed column storage (CCS) scheme in 6 RAMs for each position. In Fig. 5.3 (b) and (c) the  $L/U\_val$  RAM with the data width of 32-bit for single precision floating point numbers and the length of  $NNZ$  stores the nonzero element values, the  $L/U\_ind$  RAM with the data width of  $\log_2^{NNO}$  and the length of  $NNZ$  stores row indices of each nonzero, and the  $L/U\_ptr$  RAM with the data width of  $\log_2^{NNZ}$  and the length of  $NNO$  stores the indices of the elements in the  $L/U\_val$  vector starting each column, where  $NNO$  and  $NNZ$  are the number of nodes and nonzero elements, respectively. The switching unit determines which pre-calculated RAMs should be connected to the generic signal corresponding to the position of the moving part.

### 5.3.2 FPGA Implementation

Fig. 5.4 shows the state diagram of the designed hardware for FPGA implementation. The design consists of 6 modules equipped with command ( $cmd$ ) and done ( $dn$ ) signals to connect the modules through a main control top module. The procedure begins with the *Movement modeling* module that uses the mover speed ( $V_x$ ) and emulation time-step ( $T_s$ ) inputs to update the node coordinates RAMs ( $X - RAM$ ), determine the anti-periodicity ( $AP$ ) and position number ( $PN$ ) output signals and update the meshing data RAM ( $Mesh - RAM$ ). As the  $MOV\_dn$  signal is enabled, the process switches to the *Power supply* module and uses the computed input of  $AP$  and external inputs of effective current amplitude ( $I_{rms}$ ) and supply frequency ( $f_s$ ) to update the primary current density supply RAM ( $J - RAM$ ). Then, the process goes to the *TLM source* module with the TLM parameters inputs of  $\nu_{TLM}$  and  $T_{TLM}$ , where the right hand side of (5.14) is calculated and stored in the  $b - RAM$  followed by applying the boundary condition. In the *Forward elimination and backward substitution* module, using the  $PN$  internal input signal, the corresponding pre-calculated LU decomposition signals are connected to update the node magnetic vector potential RAM ( $A_{Nodes} - RAM$ ). Afterward, the *Next incident potential* module updates the reflected magnetic vector potential RAM ( $A_{reflected} - RAM$ ), followed by updating the next incident potential RAMs ( $A_{NIP} - RAM$ ) for linear con-

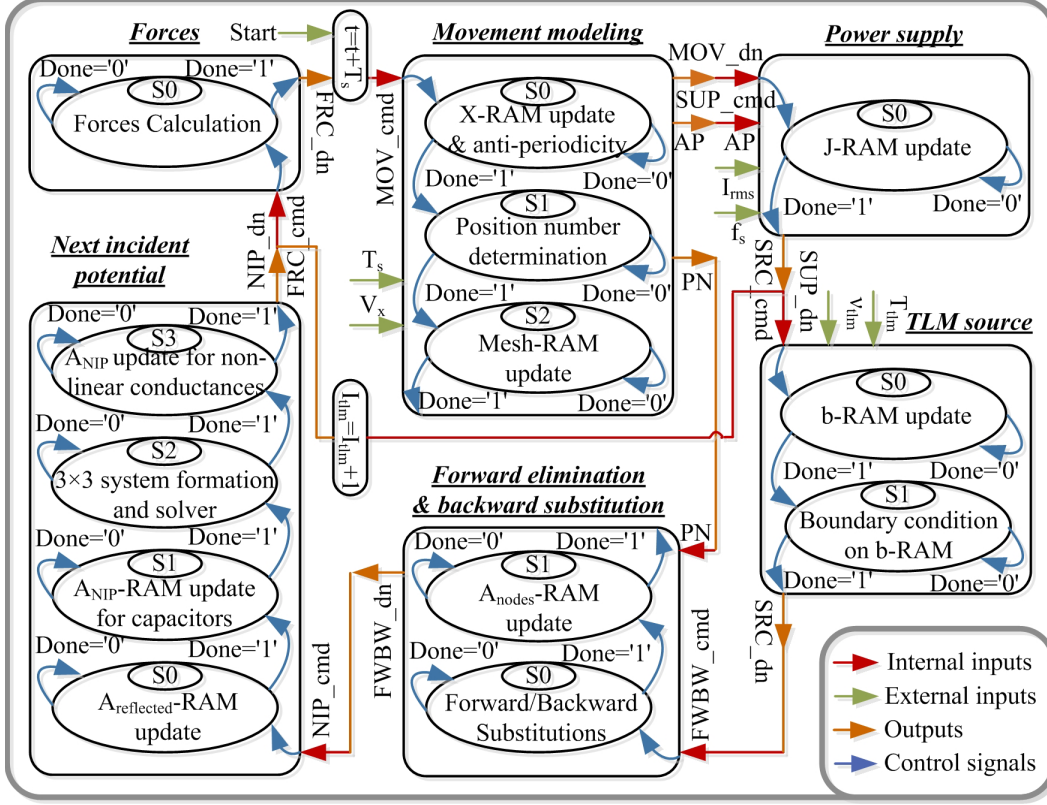


Figure 5.4: Finite state machine of the novel RT-TLM FEM hardware.

ductances and capacitances, and the nonlinear conductances by convergence of N-R method. If the TLM is not converged, the process goes back to re-updating the  $b - RAM$  with the updated  $A_{NIP} - RAM$ , otherwise the force module calculates the forces and the time-step emulation has ended. External inputs shown in Fig. 4 are the interface to exchange data through input/output pins (I/O pins) of the FPGA with the interacted devices required for HIL simulation.

The design includes 20,000 lines of handwritten VHDL code with Xilinx Vivado<sup>®</sup> software, and afterward behavioral simulation, synthesis, design initialization, optimization and placing the design, respectively. The bit stream is generated and implemented on the Virtex UltraScale+ XCVU9P FPGA board using a JTAG interface. The two main reasons behind selecting the large and recently developed device include massive on chip BRAM resources and high operating frequency. The board FMC port is connected to the DAC board by



a FMC to DAC adapter to send the results to the oscilloscope.

## 5.4 Real-Time Hardware Emulation Results

### 5.4.1 Real-Time Results

The case study is a 1.41 MVA industrial prototype SLIM designed for a maglev-based transportation system with 3.33 m length, 26.67 cm width, 1136 kg weight, 21 pole pitches, and the air gap length of 1 cm. Fig. 5.5 shows the novel RT-TLM method real-time oscilloscope results of the FPGA for the SLIM propulsion and levitation forces in time domain. At  $t = 0s$  the command speed, frequency, and rated phase current of 18.5 m/s, 69 Hz, and 1541 A are applied to the machine to obtain 26,000 N of the propulsion and 11,110 N of the levitation forces. The primary mover speed and supply frequency changes simultaneously from 18.5 to 26 m/s and 69 to 92 Hz at  $t = 1s$ , resulting in the increase of the propulsion and levitation forces to 28,100 N and 19,000 N, respectively. At  $t = 2s$  the supply current reduces from 1541 A to 1386 A, resulting in the decrease of the propulsion and levitation forces to 22,800 N and 15,400 N, respectively.

### 5.4.2 Results Verification and Accuracy Evaluation

Fig. 5.6 shows the real-time results of the propulsion and levitation forces versus speed based on the proposed RT-TLM FEM validated by Jmag-Designer<sup>®</sup> software for a wide range of current amplitudes ( $I_1 = 1541A_{rms}$ ,  $I_2 = 0.5I_1$ ), frequencies ( $f_1 = 92Hz$ ,  $f_2 = 0.75f_1$ ,  $f_3 = 0.5f_1$ ,  $f_4 = 0.25f_1$ ) and speeds ( $[0 - 40]m/s$ ) to show the accuracy of the proposed methodology. It can be seen the propulsion force behavior in the SLIM is similar to the torque of a rotary induction motor. The positive and negative propulsion forces indicate the machine motoring and regenerating modes, respectively. Similarly, the positive and negative levitation forces indicate the attraction and repulsion between the primary mover and secondary back iron, respectively.

Fig. 5.7 shows the SLIM losses including iron core hysteresis and eddy current losses, primary winding copper losses, secondary aluminum sheet losses,

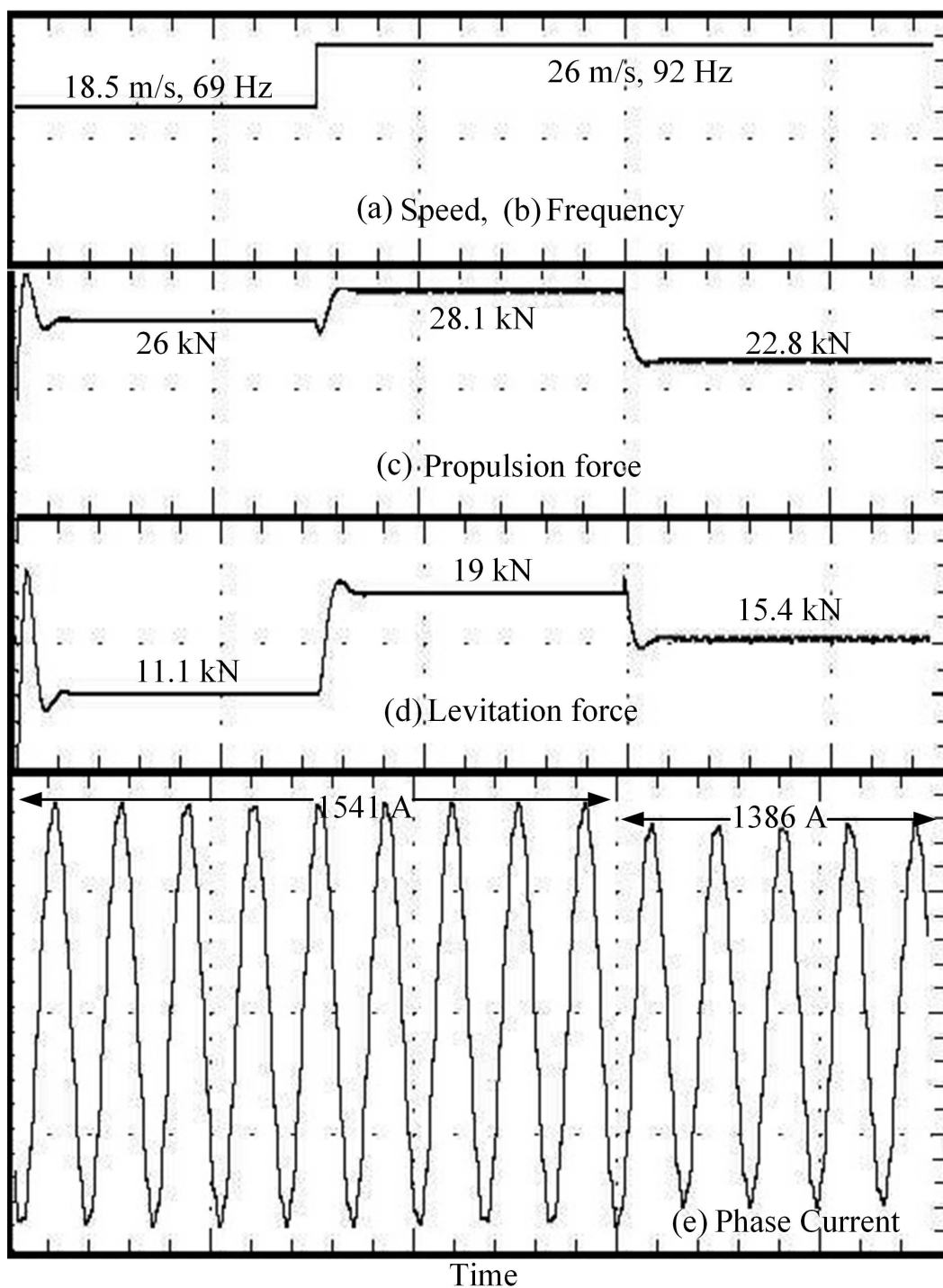


Figure 5.5: Novel RT-TLM method real-time oscilloscope results of the FPGA. (a) Primary mover speed 15 m/s/div, (b) Supply frequency 50 Hz/div, (c) Propulsion force 15,000 N/div, (d) Levitation force 15,000 N/div, (e) Supply phase current 2,500 A/div.

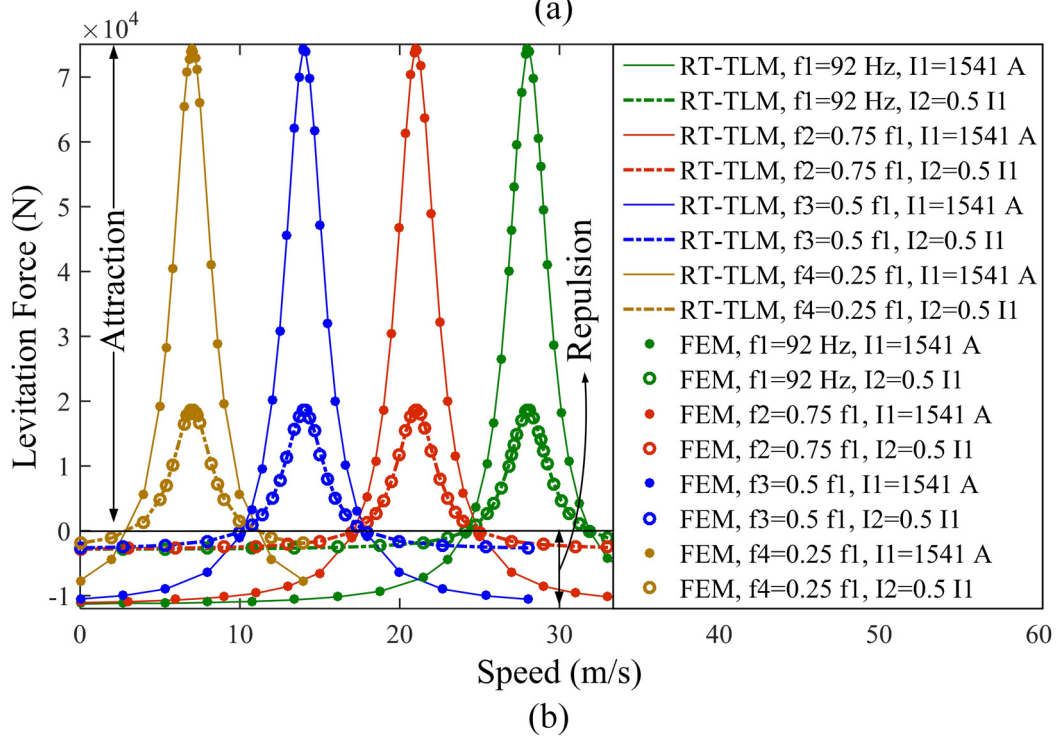
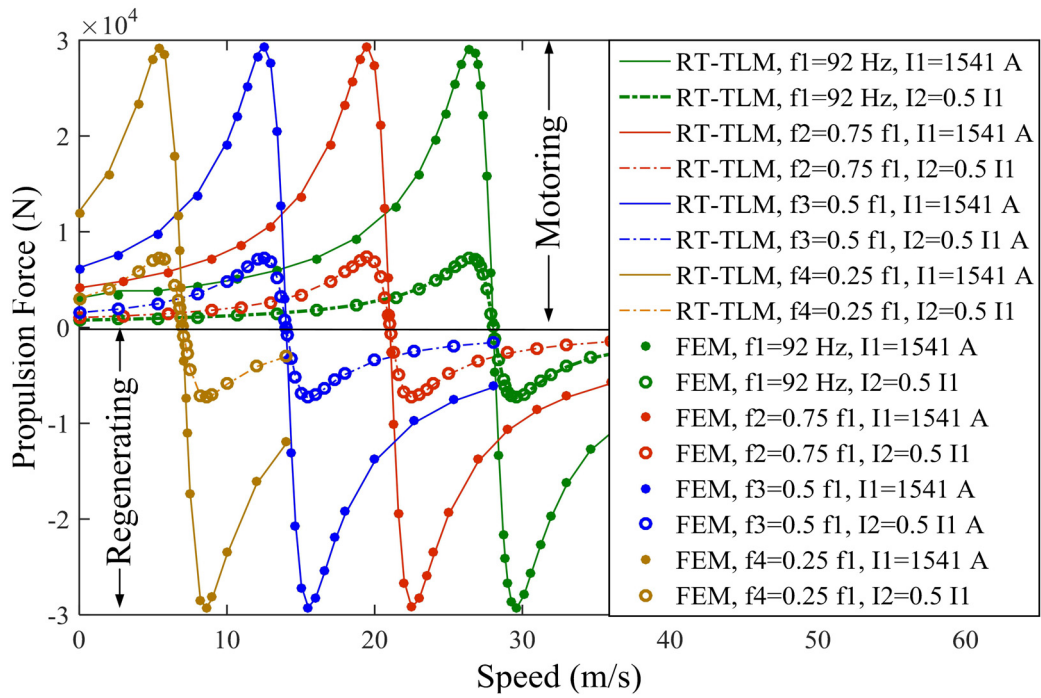


Figure 5.6: The novel RT-TLM results validation with Jmag-Designer<sup>®</sup>. (a) Propulsion force vs. speed, and (b) Levitation force vs. speed.

and the total losses versus a range of slip frequencies between 6 to 14 Hz with the primary mover speed of 26.82 m/s. The primary windings supplied by the rated current of 1541 A, generate frequency independent copper losses neglecting the skin effect. However, the iron core hysteresis and eddy current losses as well as aluminum sheet losses are a function of frequency, speed, and magnetic field density. It is worth noting the iron losses in the secondary are negligible since the frequency of the magnetic field in the secondary is the slip frequency,  $sf_s = f_s - V_x/2\pi h$ , where  $V_x$  is the primary mover speed and  $h$  is the pole pitch length. It can be seen that increasing the slip frequency in the range reduces the iron core losses while the aluminum sheet losses increase.

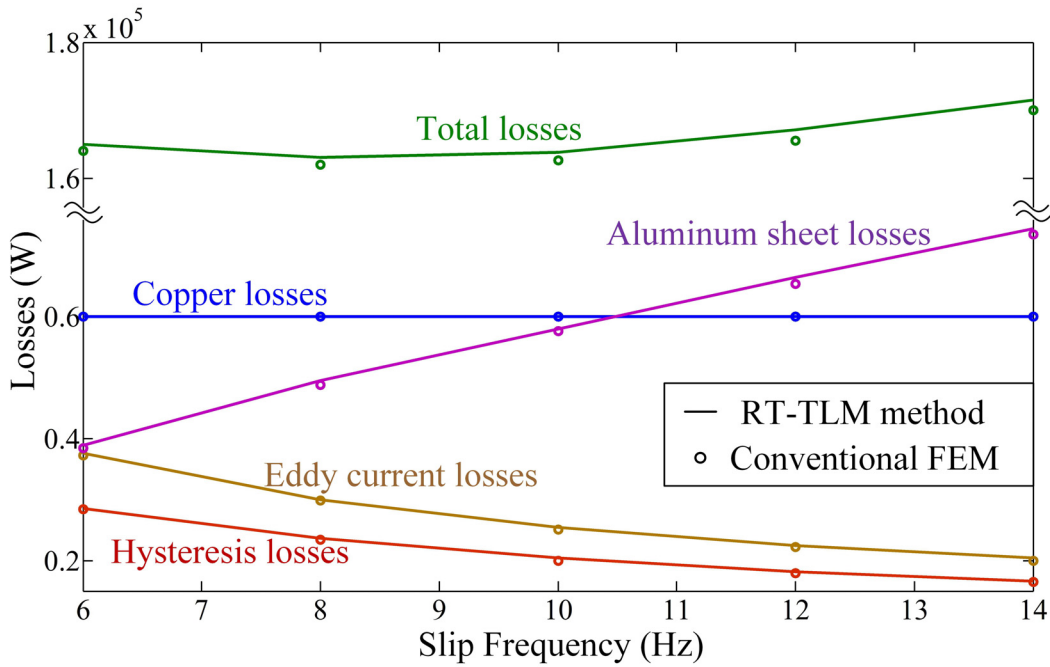


Figure 5.7: The novel RT-TLM method losses, validated by conventional FEM.

Fig. 5.8 shows the propulsion and levitation forces versus slip frequency results comparison between the experimental, Jmag-Designer<sup>®</sup> software, conventional FEM, conventional TLM-based FEM, and the novel RT-TLM FEM for the same inputs of Fig 5.7. The conventional FEM code in Matlab and the Jmag-Designer<sup>®</sup> software with the same set of inputs are simulating a single pole pitch of the LIM with anti-periodicity boundary condition to efficiently reduce the size of the problem. The solver in Matlab FEM code

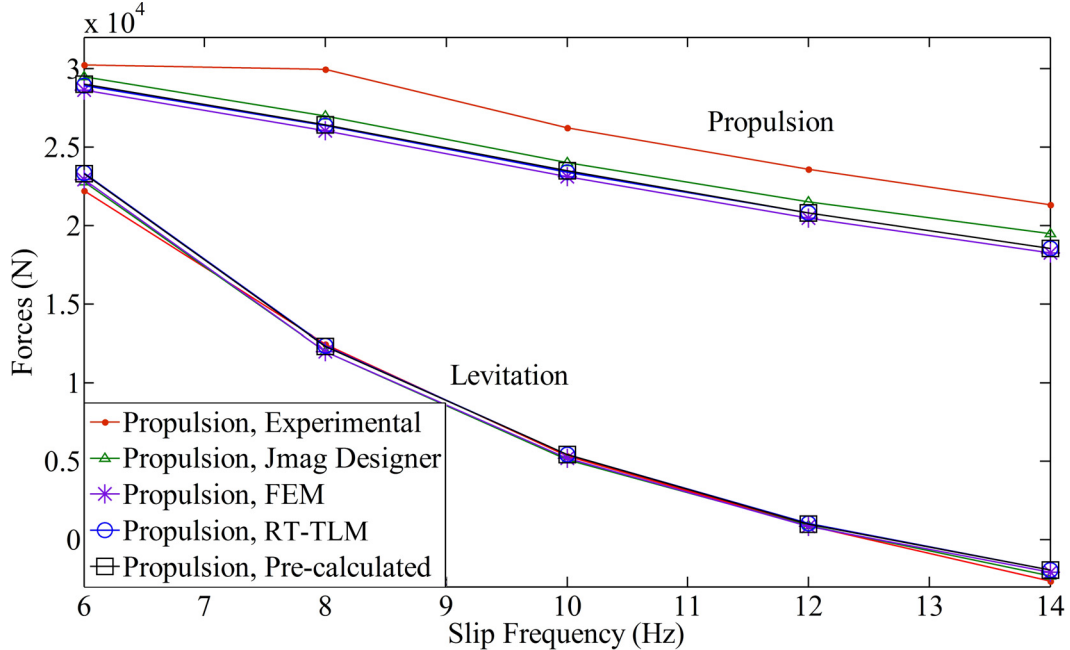


Figure 5.8: Real-time results validation with experimental, Jmag-Designer<sup>®</sup>, conventional FEM, conventional TLM, and the novel RT-TLM method.

is a direct solver with exact solution based on LU decomposition using left-looking Gilbert-Peierls algorithm, while the solver in Jmag-Designer<sup>®</sup> software is an iterative method based on the incomplete Cholesky conjugate gradient (ICCG). The difference between both simulations and the experimental results are arisen from the FEM approximations as well as neglecting the transverse edge and longitude effects due to 2-D and one pole pitch modeling, respectively, to take the advantage of much less computational effort. The off-line TLM-based FEM code in Matlab cause more deviations associated with TLM iterative algorithm. The proposed RT-TLM FEM with *finite pre-calculated LU decompositions* introduces additional approximations in both algorithm and hardware implementation. On the algorithm side, *finite pre-calculated LU decompositions* maps unlimited possible number of relative positions of the moving part to the stationary part into the finite ones. The approximations can be reduced by increasing the number of finite positions, while the hardware resource especially the BRAMs required for implementation will proportionally increase. On the hardware side, the design is implemented using 32-bit single precision floating point numbers resulting in a round off error,

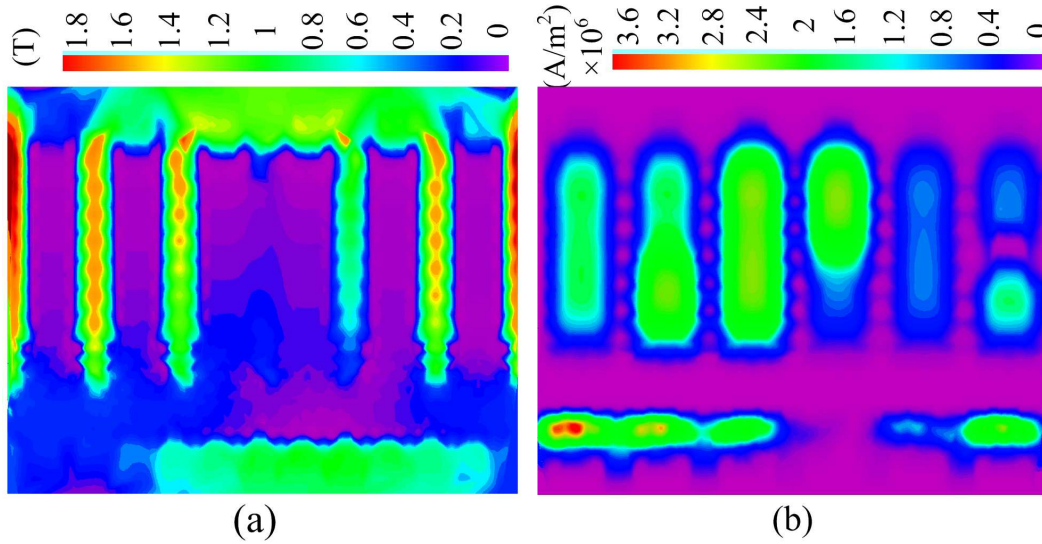


Figure 5.9: The novel RT-TLM results. (a) Magnetic field distribution, (b) Current density distribution.

while the other simulation results were based on 64-bit double precision floating point numbers. Although in hardware 64-bit can be also used for higher precision, but it results in utilizing more resources and longer latencies. In spite of the multiple levels of accuracy degradation for faster execution, the real-time results are still in good agreement with the experimental results as the benchmark.

In Fig. 5.9, the novel RT-TLM method real-time results are obtained and plotted using Matlab to show the magnetic field distribution and the current density distributions with the supply rated current of 1541 A and frequency of 92 Hz. As indicated in Fig. 5.2, the one pole pitch meshing of the SLIM includes 583 nodes and 1008 elements.

### 5.4.3 Resource Utilization and Scalability

The designed hardware is implemented on the Virtex UltraScale+ XCVU9P FPGA to evaluate the hardware resource utilization reported in Table 5.1. It can be seen the 48 pre-calculated LU decompositions corresponding to 48 positions on the moving airgap occupied the entire RAMs of the FPGA.

Each module is placed and routed on the FPGA chip and the floor plan of the implemented design is shown in Fig. 5.10. The RAMs storing the



Table 5.1: Hardware resource utilization

Module	LUT	LUTRAM	FF	BRAM
Movement modeling	2726	135	3527	2.5
Power supply	6097	467	10616	5
TLM source	56407	2894	90752	20
Forward/Backward	14692	118	8657	2064
Next incident	118442	7344	156611	15
Force calculation	14090	889	19864	9
Others	552	20	812	0
Total	213006	11867	290839	2115.5
Percentage	18.02%	2.01%	12.30%	97.94%

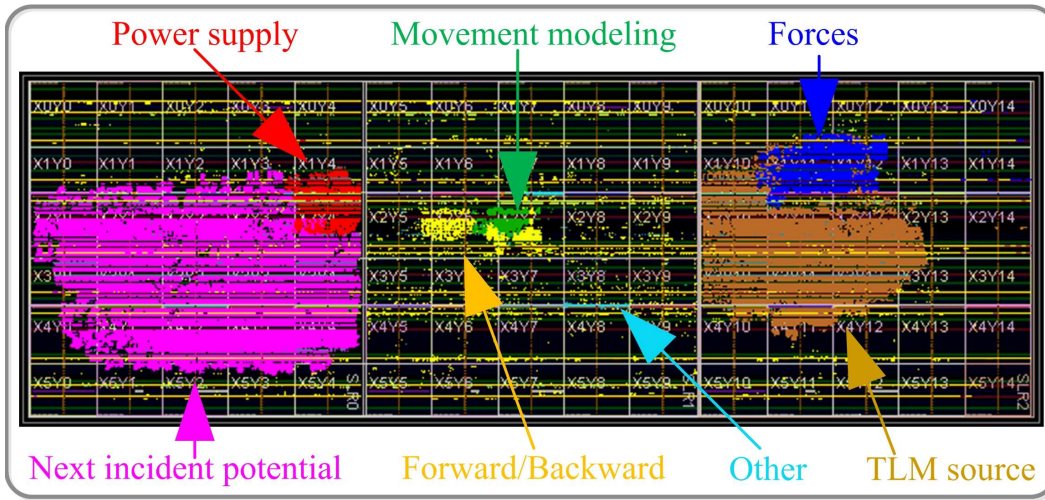


Figure 5.10: Floor plan of the mapped novel RT-TLM FEM design on Virtex UltraScale+ XCVU9P FPGA.

pre-calculated LU decompositions in the forward elimination and backward substitution module are distributed through the entire board surface shown in yellow.

The pre-processing stage can be performed on either another CPU-based unit or on the same FPGA board subject to the availability of hardware resource and pre-processing computational burden. In this chapter, the 48 pre-calculated LU decompositions are performed on CPU and loaded into FPGA BRAMs for simplicity, but it can be also fitted into the targeted FPGA.

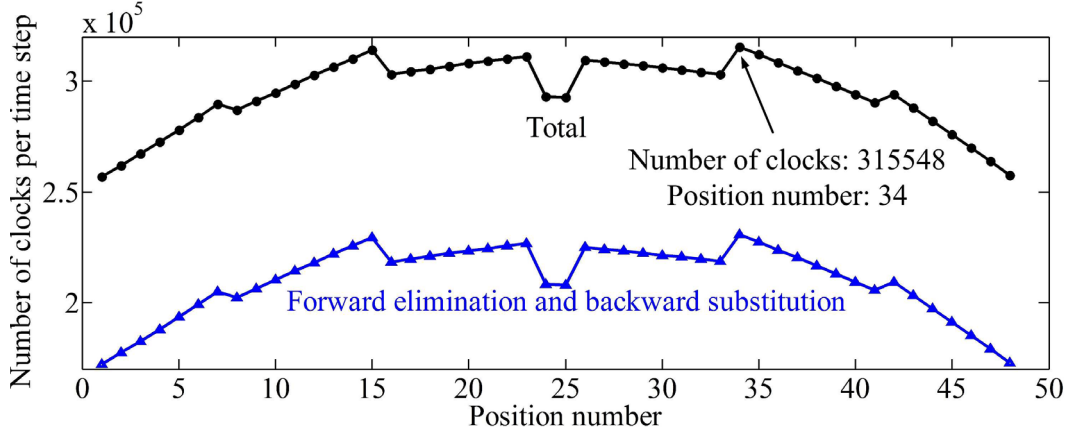


Figure 5.11: Number of clocks per time-step versus the position number.

#### 5.4.4 Timing Analysis

The relative position of the moving part to the stationary part changes the nonzero pattern of the FEM stiffness matrix and accordingly the  $L$  and  $U$  decomposed matrices. Therefore, the number of floating point operations are subject to change affecting the number of clocks per time-step for each relative position. Fig. 5.11 shows the number of clocks per time-step versus the position number for the forward elimination and backward substitution module as well as the total clocks per time-step.

For the case study it is shown the maximum number of clocks per time-step happens in the position number of 34, which should be considered in determination of the minimum achievable time-step of the design. Other position numbers will result in a time slack in real-time simulation.

The TLM lines relativity ( $\nu_{TLM}^{(e)}$ ) of 1000 m/H for nonlinear conductances and the time delay ( $T_{TLM}$ ) of 0.5 ms for all transmission lines are chosen to converge the TLM iteration and decoupled N-R iteration with an acceptable number of 4 and 3 iterations, respectively, both with convergency tolerance of  $10^{-3}$ . Table 5.2 shows the latencies of each hardware module in terms of the number of clocks per time-step with the specified TLM parameters.

Based on the Table 5.2, considering the maximum frequency of 157.77 MHz to drive the FPGA, the minimum time-step of 2 ms can be achieved. Since the dynamics of a linear induction motor as an electromechanical device is slow,



Table 5.2: Latencies of each module

Module	No. of clocks per iteration	No. of iterations per time-step	No. of clocks per time-step
Movement modeling	565	1	565
Power supply	346	1	346
TLM source	17605	$n_{TLM}$	70420
Boundary condition	162	$n_{TLM}$	648
Forward elimination	28849	$n_{TLM}$	115396
Backward substitution	28877	$n_{TLM}$	115508
Next incident potential	1796	$n_{TLM} \times (n_{NR})$	12464
Force calculation	201	1	201
<b>Total</b>	-	-	<b>315548 Clks</b>

the minimum achievable time-step of 2  $ms$  can be considered small enough to model the machine behavior shown by validation of the results with FEM and experiments. From HIL and interfacing prospective, different components on a real-time simulator can have different time-steps corresponding to the component dynamics. For example in the case of an electric machine drive system, the control system and power converter can have time-steps in the range of  $\mu s$  and  $ns$  respectively and efficiently connected to an electric machine with the time-step of 2  $ms$ , however it requires a digital integration of the converter output PWM voltage to supply the machine.

It is worth noting that pre-processing stage is unavoidable to satisfy the timing constraints of real-time simulation, especially for complex and large models including FEM. Furthermore, the small pre-processing stage proposed in the chapter for geometry definition, discretization and pre-calculated LU decompositions needs to be carried out once when the machine structure changes. However, even in machine design optimization procedure, once it is performed, it is used for infinite operating positions and simulation time-steps for performance evaluation.

## 5.5 Summary

FEM-based HIL emulation provides the most accurate and fast prototype platform for real-time design and testing of electric machines in a non-destructive environment. The application of TLM can expeditiously reduce the FEM ex-

ecution time by decoupling the nonlinear elements of the FEM equivalent network using transmission lines to keep the stiffness matrix unchanged through the simulation for static cases. However, in electric machines the TLM method suffers from the change of stiffness matrix in the time-stepped procedure due to movement. Furthermore, time consumption for solution of numerous decoupled nonlinear equations for fairly large number of TLM iterations in comparison with the conventional Newton-Raphson (N-R) method remains a challenge. This chapter proposes a novel RT-TLM method based on finite pre-calculated LU decompositions, and FPGA hardware implementation to exploit TLM parallelism for real-time simulation of magnetodynamics in electric machines. A 2-D FEM simulation of a SLIM is emulated in hardware and the results are validated experimentally and with Jmag-Designer<sup>®</sup> software to show the effectiveness of the proposed method.

# Chapter 6

## Conclusion

Conventionally, off-line simulations were used to evaluate the behavior of an electric machine and the drive system, which suffer from long simulation time as well as inaccuracy due to simplification of the model. As a substitute, hardware-in-the-loop emulation is promising in electric machine prototyping since the simulated electric machine can be connected to actual external devices under test and transfer data in real-time for accurate and fast testing. HIL is fairly straightforward for models such as lumped q-d vector model due to its simplicity. However, simulations with detailed and accurate models, i.e. magnetic equivalent circuit and finite element method models, considering all physical phenomenon including nonlinearity and saturation have high computational burden. Physics-based models, i.e. FEM and MEC, have a similar ability in considering magnetic nonlinearity, geometrical distributed effects and internal machine faults; however, since MEC is represented by lumped permeances of the flux paths, it requires prior knowledge of the flux paths and its accuracy in modeling of local phenomenon in comparison with FEM is sacrificed. So, FEM outperforms MEC in term of detailed and accurate modeling, which can affect the performance evaluation of an electric machine including losses. This thesis proposes the benefit of FPGA paralleled hardware architecture to overcome the computation of the three widely used electric machine models and addresses all challenges and solutions in real-time hardware-in-the-loop simulation of electric machines and drives for electrified transportation systems. In this work, the Virtex-7 XC7VX485T and Virtex

UltraScale+ XCVU9P FPGA devices are used based on the required clock speed and hardware resource for each application.

In this chapter, the contributions of this thesis are presented and the directions for future works are proposed.

## 6.1 Contributions of This Thesis

- In Chapter 2, a complete linear induction motor drive system based on lumped q-d vector model is emulated on FPGA for real-time hardware-in-the-loop testing. The implementation is performed through an evaluation of fixed-point number representation using XSG and floating-point number representation using handwritten VHDL program. It is shown that with floating-point number representation more accurate results can be achieved with the cost of requiring more hardware resource and larger time-step in comparison to fixed-point number representation, when both implementations are performed in massively parallel approach. The minimum time-step of  $2.3 \mu s$  and  $0.8 \mu s$  was achieved for floating-point and fixed-point implementations, respectively. The compatibility of the real-time results with off-line simulations and verification by experiments as well as finite element simulation shows the accuracy and effectiveness of the proposed scheme in high speed execution.
- In Chapter 3, for the first time, the RT-TLM emulation of faulted induction machines is proposed, facing the timing constraint challenges with high band width and non-periodic boundary condition of fault studies. The TLM method is applied to the MEC model in order to keep the coefficient matrix unchanged during each time-step, requiring only one LU decomposition per time-step due to movement. The MEC-based TLM matrix is re-ordered efficiently to keep the majority of the matrix unchanged during the entire simulation to facilitate partial pre-calculation of LU decomposition through the left looking Gilbert-Peierls algorithm. Taking the advantage of FPGAs in parallel processing of the algorithm, the minimum time-step of  $500 \mu s$  with the FPGA clock frequency of

300  $MHz$  is achieved for real-time fault study of electric machines. The results show the accuracy of the proposed method in obtaining the harmonics of the stator winding currents validated by the Jmag-Designer<sup>®</sup> commercial FEM software.

- In Chapter 4, the FPGA-based hardware prototyping of a single-sided linear induction machine in order to resolve the problem of long execution time of finite element analysis is proposed. Thanks to the hardware architecture of FPGAs, the massive finite element computation is accelerated through deeply pipelined and massively paralleled scheme with 32-bit single-precision number representation. A new sparse matrix solver based on Gilbert-Peierls algorithm has been proposed combining the symbolic analysis and matrix fill-in calculation, appropriate for time-stepped, nonlinear and medium-sized matrices of any FEM problem on FPGA by utilization of six auxiliary matrices that store the sparsity pattern. The results of the hardware prototype are validated using JMAG-Designer<sup>®</sup> through magnetic field distribution, propulsion and levitation forces characteristics. It is shown that the hardware prototype verified results can reach the average of 9.73 times speed-up in comparison with the well-known commercial finite element software JMAG-Designer<sup>®</sup>.
- In Chapter 5, for the first time, real-time finite element emulation of an electric machine as a nonlinear magnetodynamic case involving movement is performed for hardware-in-the-loop prototyping on FPGA. Two novel ideas are proposed along with the transmission line modeling method to overcome the massive time consumption of FEM and enforce real-time execution. First, a novel RT-TLM method based on *finite pre-calculated LU decomposition* is proposed to avoid LU decomposition of the stiffness matrix for each time-step. Second, the parallelism of TLM method and the solver algorithms are fully exploited for a massively parallel implementation on the FPGA. It is shown the simulation time-step on the utilized hardware can be as low as 2  $ms$ , and the accuracy of results including the forces vs. speed curves and losses are evaluated experimentally

and with Jmag-Designer<sup>®</sup>. The proposed method is readily applicable for real-time magnetodynamic simulation and analysis of other types of linear and rotary electric machines.

## 6.2 Directions for Future Work

- Most of the current real-time simulators employ multi-core CPUs, including NovaCor<sup>®</sup> developed by RTDS Technologies Inc., HYPERSIM<sup>®</sup> developed by Hydro-Québec, and eMEGAsim<sup>®</sup> developed by Opal-RT Technologies Inc. Although the high clock frequency of multi-core CPUs for sequential operations with limited parallelism is attractive, the massively paralleled and deeply pipelined architecture of FPGAs makes it a superior platform for parallel processing applicable in real-time simulation. The integration of multi-core CPUs and FPGAs as the ideal combination offers the advantage of high clock frequency operation of CPUs for sequential operations, and the massive hardware parallelism of FPGAs for parallel operations of a system at the same time.
- Considering the advancement in the real-time simulation of electric machines by transmission line modeling method proposed in this thesis, it can be notified that the sparse forward elimination and backward substitution as the bottleneck are required to be performed in every TLM iteration. So, focus on reducing the computational burden of the sparse forward elimination and backward substitution, and employing algorithms that can lead to more parallelism can be a topic of interest for future work.
- Real-time simulation of electric machines with computationally intense detailed models still suffers from the large minimum achievable time-step, which affects the efficiency of the HIL scenario. In spite of the fairly slow dynamic of electric machines, their interactions with an actual fast acting power converter, drive systems, and the protection devices are required. Innovative algorithms and methods should be developed to

further exploit parallelism to achieve time-steps as small as a few  $\mu s$  with accurate and reliable machine models.

- Domain decomposition methods divide a large study domain into several smaller sub-domains for potential execution acceleration and parallelism. The efficiency of the domain decomposition methods with FEM computations for an induction motor should be evaluated. In addition, combining the domain decomposition and TLM methods to further facilitate pre-calculation and accelerate the massive FEM computations on FPGA is suggested for future work.
- In this thesis, the main frequency and low order harmonic supply voltages are considered. However, in most cases, the electric machine is supplied with a high frequency PWM voltage. So, for accurate and realistic real-time simulation, the high frequency electric machine models considering parasitic capacitors is suggested for future works.
- In this thesis all types of electrical faults are investigated for real-time simulation. However, mechanical faults including eccentricity and bearing failures that may require 3-D modeling is left for future work.

# Bibliography

- [1] B. C. Mecrow and A. G. Jack, “Efficiency trends in electric machines and drives,” *Energy Policy*, vol. 36, no. 12, pp. 4336–4341, Dec. 2008.
- [2] Y. Cao, R. C. Kroeze, and P. T. Krein, “Multi-timescale parametric electrical battery model for use in dynamic electric vehicle simulations,” *IEEE Trans. Transport. Electrification*, vol. 2, no. 4, pp. 432–442, Dec. 2016.
- [3] S. C. Carpiuc and C. Lazar, “Fast real-time constrained predictive current control in permanent magnet synchronous machine-based automotive traction drives,” *IEEE Trans. Transport. Electrification*, vol. 1, no. 4, pp. 326–335, Dec. 2015.
- [4] K. Itani, A. de Bernardinis, Z. Khatir, A. Jammal, and M. Oueidat, “Extreme conditions regenerative braking modeling, control and simulation of a hybrid energy storage system for an electric vehicle,” *IEEE Trans. Transport. Electrification*, vol. 2, no. 4, pp. 465–479, Dec. 2016.
- [5] H. Yin, W. Zhou, M. Li, C. Ma, and C. Zhao, “An adaptive fuzzy logicbased energy management strategy on battery/ultracapacitor hybrid electric vehicles,” *IEEE Trans. Transport. Electrification*, vol. 2, no. 3, pp. 300–311, Sep. 2016.
- [6] F. Gao, X. Zheng, S. Bozhko, C. I. Hill, and G. Asher, “Modal analysis of a PMSG-based DC electrical power system in the more electric aircraft using eigenvalues sensitivity,” *IEEE Trans. Transport. Electrification*, vol. 1, no. 1, pp. 65–76, Jun. 2015.



- [7] P. C. Krause and C. H. Thomas, “Simulation of symmetrical induction machinery,” *IEEE Trans. Power App. Syst.*, vol. 84, no. 11, pp. 1038–1053, Nov. 1965.
- [8] H. W. Dommel, “Digital computer solution of electromagnetic transients in single- and multiphase networks,” *IEEE Trans. Power App. Syst.*, vol. PAS-88, no. 4, pp. 388–399, Apr. 1969.
- [9] X. Guillaud et al., “Applications of real-time simulation technologies in power and energy systems,” *IEEE Power Energy Technol. Syst. J.*, vol. 2, no. 3, pp. 103–115, Sep. 2015.
- [10] M. W. Naouar, E. Monmasson, A. A. Naassani, I. Slama-Belkhodja, and N. Patin, “FPGA-based current controllers for AC machine drives—A review,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1907–1925, Aug. 2007.
- [11] V. Jalili-Marandi, L. F. Pak, and V. Dinavahi, “Real-time simulation of grid-connected wind farms using physical aggregation,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 9, pp. 3010–3021, Sep. 2010.
- [12] R. Champagne, L. A. Dessaint, H. Fortin-Blanchette, and G. Sybille, “Analysis and validation of a real-time AC drive simulator,” *IEEE Trans. Power Electron.*, vol. 19, no. 2, pp. 336–345, Mar. 2004.
- [13] C. Dufour, S. Abourida, and J. Bélanger, “Hardware-in-the-loop simulation of power drives with RT-LAB,” in *Proc. Int. Conf. Power Electron. Drives Syst.*, Dec. 2005, pp. 1646–1651.
- [14] S. Abourida, C. Dufour, and J. Bélanger, “Real-time and hardware-in-the-loop simulation of electric drives and power electronics: Process, problems and solutions,” in *Proc. Int. Power Electron. Conf.*, Niigata, Japan, Apr. 2005, pp. 1908–1913.
- [15] M. Harakawa, H. Yamasaki, T. Nagano, S. Abourida, C. Dufour, and J. Bélanger, “Real-time HIL simulation of a complete PMSM drive at 10

- micro sec time step,” in *Proc. Int. Power Electron. Conf.*, Sep. 2005, pp. 1–8.
- [16] M. S. Vekic, S. U. Grabic, D. P. Majstorovic, I. L. Celanovic, N. L. Celanovic, and V. A. Katic, “Ultralow latency HIL platform for rapid development of complex power electronics systems,” *IEEE Trans. Power Electron.*, vol. 27, no. 11, pp. 4436–4444, Nov. 2012.
- [17] R. E. Crosbie, “Advances in high-speed real-time simulation,” in *Proc. Eur. Symp. Comput. Modeling Simulation*, Sep. 2008, pp. 2–8.
- [18] A. Bouscayrol, “Different types of hardware-in-the-loop simulation for electric drives,” in *Proc. IEEE Int. Symp. Ind. Electron.*, Jun. 2008, pp. 2146–2151.
- [19] J.-N. Paquin, J. Moyon, G. Dumur, and V. Lapointe, “Real-time and offline simulation of a detailed wind farm model connected to a multi-bus network,” in *Proc. IEEE Elect. Power Conf.*, Oct. 2007, pp. 145–152.
- [20] J. Bélanger, V. Lapointe, C. Dufour, and L. Schoen, “eMEGAsim: An open high-performance distributed real-time power grid simulator. Architecture and specification,” in *Proc. Int. Conf. Power Syst.*, Dec. 2007, pp. 12–24.
- [21] F. M. Uriarte and K. L. Butler-Purry, “Multicore simulation of an AC-radial shipboard power system,” in *Proc. IEEE Power Energy Soc. General Meeting*, Jul. 2010, pp. 1–8.
- [22] C. Dufour, S. Cense, V. Jalili-Marandi, and J. Bélanger, “Review of state-of-the-art solver solutions for HIL simulation of power systems, power electronic and motor drives,” in *Proc. 15th Eur. Conf. Power Electron. Appl.* Sep. 2013, pp. 1–12.
- [23] E. Monmasson, L. Idkhajine, and M. W. Naouar, “FPGA-based controllers,” *IEEE Ind. Electron. Mag.*, vol. 5, no. 1, pp. 14–26, Mar. 2011.

- [24] C. Dufour and J. Bélanger, “A real-time simulator for doubly fed induction generator based wind turbine applications,” in *Proc. IEEE 35th Annu. Power Electron. Specialists Conf.*, vol. 5. Jun. 2004, pp. 3597–3603.
- [25] J. Rose, A. El Gamal, and A. Sangiovanni-Vincentelli, “Architecture of field-programmable gate arrays,” *Proc. IEEE*, vol. 81, no. 7, pp. 1013–1029, Jul. 1993.
- [26] E. Monmasson and M. N. Cirstea, “FPGA design methodology for industrial control systems—A review,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, Aug. 2007.
- [27] W. Ren et al., “Interfacing issues in real-time digital simulators,” *IEEE Trans. Power Del.*, vol. 26, no. 2, pp. 1221–1230, Apr. 2011.
- [28] S. Gdaim, A. Mtibaa, and M. F. Mimouni, “Design and implementation of Direct Torque Control of induction machine on FPGA,” in *Proc. 5th Int. Conf. Modeling, Simulation Appl. Optim.*, Apr. 2013, pp. 1–6.
- [29] L. Wang, J. Jatskevich, and H. W. Dommel, “Re-examination of synchronous machine modeling techniques for electromagnetic transient simulations,” *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1221–1230, Aug. 2007.
- [30] M. Moallem and G. E. Dawson, “An improved magnetic equivalent circuit method for predicting the characteristics of highly saturated electromagnetic devices,” *IEEE Trans. Magn.*, vol. 34, no. 5, pp. 3632–3635, Sep. 1998.
- [31] R. Qin and M. A. Rahman, “Magnetic equivalent circuit of PM hysteresis synchronous motor,” *IEEE Trans. Magn.*, vol. 39, no. 5, pp. 2998–3000, Sep. 2003.
- [32] W. Kemmetmüller, D. Faustner, and A. Kugi, “Modeling of a permanent magnet synchronous machine with internal magnets using magnetic

- equivalent circuits,” *IEEE Trans. Magn.*, vol. 50, no. 6, pp. 1–14, Jun. 2014.
- [33] M. Yilmaz and P. T. Krein, “Capabilities of finite element analysis and magnetic equivalent circuits for electrical machine analysis and design,” in *Proc. IEEE Power Electron. Specialists Conf.*, Jun. 2008, pp. 4027–4033.
- [34] P. C. Krause, O. Wasynczuk, S. D. Sudhoff, and S. Pekarek, *Analysis of Electric Machinery and Drive Systems*. Hoboken, NJ, USA: Wiley, 2013.
- [35] R. Kerkman, “Steady-state and transient analysis of an induction machine with saturation of the magnetizing branch,” *IEEE Trans. on Ind. Appl.*, vol. IA-21, pp. 226–234, Jan./Feb. 1985.
- [36] R. D. Lorenz and D. W. Novotny, “Saturation effects in field-oriented induction machines,” *IEEE Transactions on Industry Applications*, vol. 26, pp. 283–289, March/April 1987.
- [37] C. R. Sullivan, C. Kao, B. M. Acker, and S. R. Sullivan, “Control systems for induction machines with magnetic saturation,” *IEEE Transactions on Industrial Electronics*, vol. 43, pp. 142–152, August 1996.
- [38] C. J. Carpenter, “Magnetic equivalent circuits,” *Proc. Inst. Elect. Eng.*, vol. 115, no. 10, pp. 1503–1511, Oct. 1968.
- [39] E. C. Cherry, “The duality between interlinked electric and magnetic circuits and the formation of transformer equivalent circuits,” *Proc. Phys. Soc. Sec. B*, vol. 62, no. 2, pp. 101–111, 1949.
- [40] M. Amrhein and P. T. Krein, “Magnetic equivalent circuit simulations of electrical machines for design purposes,” in *Proc. IEEE Electr. Ship Technol. Symp.*, May 2007, pp. 254–260.
- [41] S. A. Saied, K. Abbaszadeh, and M. Fadaie, “Reduced order model of developed magnetic equivalent circuit in electrical machine modeling,” *IEEE Trans. Magn.*, vol. 46, no. 7, pp. 2649–2655, Jul. 2010.

- [42] M. Amrhein, and P. T. Krein, "Induction machine modeling approach based on 3-D magnetic equivalent circuit framework," *IEEE Trans. Energy Convers.*, vol. 25, no. 2, pp. 339-347, Jun. 2010.
- [43] M. Amrhein and P. T. Krein, "3-D magnetic equivalent circuit framework for modeling electromechanical devices," *IEEE Trans. Energy Convers.*, vol. 24, no. 2, pp. 397-405, Jun. 2009.
- [44] B. N. Cassimere and S. D. Sudhoff, "Population-based design of surface-mounted permanent-magnet synchronous machines," *IEEE Trans. Energy Convers.*, vol. 24, no. 2, pp. 338-346, Jun. 2009.
- [45] M. L. Bash and S. D. Pekarek, "Modeling of salient-pole wound rotor synchronous machines for population-based design," *IEEE Trans. Energy Convers.*, vol. 26, no. 2, pp. 381-392, Jun. 2011.
- [46] W. N. Fu, S. L. Ho, H. L. Li, and H. C. Wong, "An effective method to reduce the computing time of nonlinear time-stepping finite-element magnetic field computation," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 441-444, Mar. 2002.
- [47] G. Y. Sizov, D. M. Ionel, and N. A. O. Demerdash, "Modeling and parametric design of permanent-magnet AC machines using computationally efficient finite-element analysis," *IEEE Trans. Ind. Electron.*, vol. 59, no. 6, pp. 2403-2413, Jun. 2012.
- [48] K. Zhou, J. Pries, and H. Hofmann, "Computationally efficient 3-D finite-element-based dynamic thermal models of electric machines," *IEEE Trans. Transport. Electrific.*, vol. 1, no. 2, pp. 138-149, Aug. 2015.
- [49] W. Wang, Z. Shen, and V. Dinavahi, "Physics-based device-level power electronic circuit hardware emulation on FPGA," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2166-2179, Nov. 2014.
- [50] K. V. Iyer and N. Mohan, "Modulation and commutation of a single stage isolated asymmetrical multilevel converter for the integration of renewables

- and battery energy storage system in ships,” *IEEE Trans. Transport. Electrification.*, vol. 2, no. 4, pp. 580–596, Dec. 2016.
- [51] M. Rakotozafy, P. Poure, S. Saadate, C. Bordas, and L. Leclere, “Real-time digital simulation of power electronics systems with neutral point piloted multilevel inverter using FPGA,” *Electr. Power Syst. Res.*, vol. 81, no. 2, pp. 687–698, Feb. 2011.
- [52] M. Matar and R. Iravani, “FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients,” *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 852–860, Apr. 2010.
- [53] A. Myaing and V. Dinavahi, “FPGA-based real-time emulation of power electronic systems with detailed representation of device characteristics,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 1, pp. 358–368, Jan. 2011.
- [54] H. F. Blanchette, T. Ould-Bachir, and J. P. David, “A state-space modeling approach for the fpga-based real-time simulation of high switching frequency power converters,” *IEEE Trans. Ind. Electron.*, vol. 59, no. 12, pp. 4555–4567, Dec. 2012.
- [55] Ó. Lucia, I. Urriza, L. A. Barragan, D. Navarro, Ó. Jimenez, and J. M. Burdio, “Real-time FPGA-based hardware-in-the-loop simulation test bench applied to multiple-output power converters,” *IEEE Trans. Ind. Appl.*, vol. 47, no. 2, pp. 853–860, Mar. 2011.
- [56] E. J. Bueno, A. Hernandez, F. J. Rodriguez, C. Giron, R. Mateos, and S. Cobrecas, “A DSP- and FPGA-based industrial control with high-speed communication interfaces for grid converters applied to distributed power generation systems,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 654–669, Mar. 2009.
- [57] M. O. Faruque, and V. Dinavahi, “Hardware-in-the-loop simulation of power electronic systems using adaptive discretization,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1254–1261, Apr. 2010.

- [58] W. Wang, Z. Shen and V. Dinavahi, "Physics-based device-level power electronic circuit hardware emulation on FPGA," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2166-2197, Nov. 2014.
- [59] Z. Shen, V. Dinavahi, "Real-time device-level transient electro-thermal model of the modular multilevel converter on FPGA," *IEEE Trans. Power Electron.*, vol. PP, no. PP, pp. 1-14, 2015.
- [60] Y. Chen and V. Dinavahi, "FPGA-based real-time EMTP," *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 892-902, Apr. 2009.
- [61] Y. Chen and V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1300-1309, Feb. 2012.
- [62] Y. Chen and V. Dinavahi, "Hardware emulation building blocks for real-time simulation of large-scale power grids," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 373-381, Feb. 2014.
- [63] Y. Chen and V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA," *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2547-2555, Jun. 2011.
- [64] 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 *Proc. 17th IEEE Symp. Field Program. Custom Comput. Mach.*, 2009, pp. 227-230.
- [65] Y. Chen, V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA," *IEEE Trans. on Ind. Electron.*, vol. 58, no. 6, pp. 2547-2555, Jun. 2011.
- [66] Y. Chen, V. Dinavahi, "Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems", *IET Gen., Trans. & Distrib.*, vol. 7, no. 5, pp. 451-463, May 2013.

- [67] Y. Chen and V. Dinavahi, "Hardware emulation building blocks for real-time simulation of large-scale power grids," *IEEE Trans. on Ind. Informat.*, vol. 10, no. 1, pp. 373-381, Feb. 2014.
- [68] H. Chen, S. Sun, D. C. Aliprantis, and J. Zambreno, "Dynamic simulation of electric machines on FPGA boards," in *Proc. IEEE Int. Electr. Mach. Drives Conf.*, May 2009, pp. 1523-1528.
- [69] N. R. Tavana and V. Dinavahi, "A general framework for FPGA-based real-time emulation of electrical machines for HIL applications," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2041-2053, Apr. 2015.
- [70] G. G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," *IEEE Trans. Power Del.*, vol. 22, no. 2, pp. 1235-1246, Apr. 2007.
- [71] S. Usenmez, R. A. Dilan, M. Dolen, and A. B. Koku, "Real-time hardware-in-the-loop simulation of electrical machine systems using FPGAs," in *Proc. Int. Conf. Elect. Mach. Syst.*, Nov. 2009, pp. 1-6.
- [72] M. Matar and R. Iravani, "Massively parallel implementation of AC machine models for FPGA-based real-time simulation of electromagnetic transients," *IEEE Trans. Power Del.*, vol. 26, no. 2, pp. 830-840, Apr. 2011.
- [73] T. Sutikno, N. R. N. Idris, A. Jidin, and M. N. Cirstea, "An improved FPGA implementation of direct torque control for induction machines," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1280-1290, Aug. 2013.
- [74] D. Zhang and H. Li, "A stochastic-based FPGA controller for an induction motor drive with integrated neural network algorithms," *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 551-561, Feb. 2008.
- [75] A. Schmitt, J. Richter, U. Jurkewitz, and M. Braun, "FPGA-based real-time simulation of nonlinear permanent magnet synchronous machines for power hardware-in-the-loop emulation systems," in *Proc. 40th Annu. Conf. IEEE Ind. Electron. Soc.*, Nov. 2014, pp. 3763-3769.



- [76] A. Battiston, E.-H. Miliiani, S. Pierfederici, and F. Meibody-Tabar, “Efficiency improvement of a quasi-Z-source inverter-fed permanentmagnet synchronous machine-based electric vehicle,” *IEEE Trans. Transport. Electrific.*, vol. 2, no. 1, pp. 14–23, Mar. 2016.
- [77] T. Ould-Bachir, C. Dufour, J. Bélanger, J. Mahseredjian, and J.-P. David, “Effective floating-point calculation engines intended for the FPGA-based HIL simulation,” in *Proc. IEEE Int. Symp. Ind. Electron.*, May 2012, pp. 1363–1368.
- [78] S. Amornwongpeeti, M. Ekpanyapong, N. Chayopitak, J. L. Monteiro, J. S. Martins, and J. L. Afonso, “A single chip FPGA-based solution for controlling of multi-unit PMSM motor with time-division multiplexing scheme,” *Microprocess. Microsyst.*, vol. 39, no. 8, pp. 621–633, Nov. 2015.
- [79] E. Monmasson, L. Idkhajine, M. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, “FPGAs in industrial control applications,” *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, May 2011.
- [80] M. Dagbagi, L. Idkhajine, E. Monmasson, L. Charaabi, and I. Slama-Belkhodja, “FPGA implementation of a synchronous motor real-time emulator based on delta operator,” in *Proc. IEEE Int. Symp. Ind. Electron.*, Jun. 2011, pp. 1581–1586.
- [81] L. Idkhajine, E. Monmasson, and A. Maalouf, “Fully FPGA-based sensorless control for synchronous AC drive using an extended Kalman filter,” *IEEE Trans. Ind. Electron.*, vol. 59, no. 10, pp. 3908–3918, Oct. 2012.
- [82] R. Gregor, G. Valenzano, J. Rodríguez-Piñeiro, and J. Rodas, “FPGA-based real-time simulation of a dual three-phase induction machine,” in *Proc. 16th Eur. Conf. Power Electron. Appl.*, Aug. 2014, pp. 1–8.
- [83] P. Ponce, L. Ibarra, A. Molina, and B. MacCleery, “Real time simulation for DC and AC motors based on LabVIEW FPGAs,” *IFAC Proc. Vols.*, vol. 45, no. 6, pp. 1777–1784, May 2012.

- [84] P. A. Rajne and V. Ramanarayanan, "Programming an FPGA to emulate the dynamics of DC machines," in *Proc. Int. Conf. Power Electron.*, Dec. 2006, pp. 120–124.
- [85] Z. Liu, O. A. Mohammed, and S. Liu, "Equivalent hardware representation of PM synchronous motors from the physics-based phase variable model obtained through FE computation," *IEEE Trans. Magn.*, vol. 45, no. 3, pp. 1450–1453, Mar. 2009.
- [86] J. Gao, S. Song, Y. Huang, and S. Huang, "Implementation and test for the semi-physical real-time simulation of IPMSM based on 3-D inductance table," in *Proc. IEEE Conf. Expo Transp. Electrification Asia-Pacific (ITEC Asia-Pacific)*, Aug. 2014, pp. 1–5.
- [87] R. C. Okonkwo, A. Dehkordi, A. M. Gole, and R. Hanitsch, "Permanent magnet DC linear machine model for real time simulation," in *Proc. Can. Conf. Elect. Comput. Eng.*, May 2005, pp. 1509–1512.
- [88] A. B. Dehkordi, P. Neti, A. M. Gole, and T. L. Maguire, "Development and validation of a comprehensive synchronous machine model for a real-time environment," *IEEE Trans. Energy Convers.*, vol. 25, no. 1, pp. 34–48, Mar. 2010.
- [89] Y. Inaba, S. Cense, T. O. Bachir, H. Yamashita, and C. Dufour, "A dual high-speed PMSM motor drive emulator with finite element analysis on FPGA chip with full fault testing capability," in *Proc. 14th Eur. Conf. Power Electron. Appl.*, Aug. 2011, pp. 1–10.
- [90] C. Dufour, S. Cense, T. Yamada, R. Imamura, and J. Bélanger, "FPGA permanent magnet synchronous motor floating-point models with variable-DQ and spatial harmonic finite-element analysis solvers," in *Proc. 15th Int. Power Electron. Motion Control Conf.*, Sep. 2012, pp. LS6b.2–1–LS6b.2–10.
- [91] A. Griffo, D. Salt, R. Wrobel, and D. Drury, "Computationally efficient modelling of permanent magnet synchronous motor drives for real-time

- hardware-in-the-loop simulation,” in *Proc. 39th Annu. Conf. IEEE Ind. Electron. Soc.*, Nov. 2013, pp. 5368–5373.
- [92] Z. Bo and J. Xu, “Interpolation method in hardware-in-the-loop simulation for brushless DC motor,” in *Proc. 11th World Congr. Intell. Control Autom.*, Jun. 2014, pp. 5852–5857.
- [93] A. Sarikhani and O. A. Mohammed, “HIL-based finite-element design optimization process for the computational prototyping of electric motor drives,” *IEEE Trans. Energy Convers.*, vol. 27, no. 3, pp. 737–746, Sep. 2012.
- [94] C. Dufour, J. Bélanger, S. Abourida, and V. Lapointe, “FPGA-based real-time simulation of finite-element analysis permanent magnet synchronous machine drives,” in *Proc. IEEE Conf. Power Electron. Specialist*, Jun. 2007, pp. 909–915.
- [95] S. Abourida, C. Dufour, J. Bélanger, T. Yamada, and T. Arasawa, “Hardware-in-the-loop simulation of finite-element based motor drives with RT-LAB and JMAG,” in *Proc. IEEE Int. Symp. Ind. Electron.*, Jul. 2006, pp. 2462–2466.
- [96] L. Herrera, C. Li, X. Yao, and J. Wang, “FPGA-based detailed real-time simulation of power converters and electric machines for EV HIL applications,” *IEEE Trans. Ind. Appl.*, vol. 51, no. 2, pp. 1702–1712, Mar. 2015.
- [97] N. R. Tavana and V. Dinavahi, “Real-time FPGA-based analytical space harmonic model of permanent magnet machines for hardware-in-the-loop simulation,” *IEEE Trans. Magn.*, vol. 51, no. 8, pp. 1–9, Aug. 2015.
- [98] J. Liu, and V. Dinavahi, “A real-time nonlinear hysteretic power transformer transient model on FPGA,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3587–3597, Jul. 2014.

- [99] J. Liu, and V. Dinavahi, "Nonlinear magnetic equivalent circuit-based real-time Sen transformer electromagnetic transient model on FPGA for HIL emulation," *IEEE Trans. Power Del.*, vol. 31, no. 6, pp. 2483–2493, Dec. 2016.
- [100] P. Liu, and V. Dinavahi, "Real-Time Finite-Element Simulation of Electromagnetic Transients of Transformer on FPGA," *IEEE Trans. Power Del.*, DOI: 10.1109/TPWRD.2018.2812753.
- [101] B. Asghari, and V. Dinavahi, "Novel transmission line modeling method for nonlinear permeance network based simulation of induction machines," *IEEE Trans. Magn.*, vol. 47, no. 8, pp. 2100-2108, Aug. 2011.
- [102] B. Asghari, and V. Dinavahi, "Experimental validation of a geometrical nonlinear permeance network based real-time induction machine model," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4049-4062, Nov. 2012.
- [103] F. Fleming, and C. Edrington, "Real-time emulation of switched reluctance machines via magnetic equivalent circuits," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3366-3376, Jun. 2016.
- [104] N. R. Tavana, and V. Dinavahi, "Real-Time nonlinear magnetic equivalent circuit model of induction machine on FPGA for Hardware-in-the-Loop simulation," *IEEE Trans. Energy Convers.*, vol. 31, no. 2, pp. 520-530, Jun. 2016.
- [105] G. V. Sadler, and A. W. Davey, "Applications of linear induction motors in industry," *IEE Proc.*, vol. 118, no. 6, pp. 765-776, Jun. 1971.
- [106] J. S. Choi, and Y. S. Baek, "Magnetically-levitated steel-plate conveyance system using electromagnets and a linear induction motor," *IEEE Trans. Magn.*, vol. 44, no. 11, pp. 4171-4174, Nov. 2008.
- [107] Y. Cho, and J. Lee, "An investigation on the characteristics of a single-sided linear induction motor at standstill for maglev vehicle," *IEEE Trans. Magn.*, vol. 33, no. 2, pp. 2073-2076, Mar. 1997.

- [108] U. Hasirci, A. Balikci, Z. Zabar, and L. Birenbaum, “A novel magnetic-levitation system: design, implementation, and nonlinear control,” *IEEE Trans. Plasma Sci.*, vol. 39, no. 1, pp. 492-497, Jul. 2010.
- [109] E. R. Laithwaite, “Linear electric machines – a personal view”, *Proc. IEEE*, vol. 63, no 2, pp. 250–290, 1975.
- [110] E. R. Laithwaite, and S. A. Nasar, “Linear-motion electrical machines,” *Proc. IEEE*, vol. 58, no. 4, pp. 531–542, Apr. 1970.
- [111] M. R. Doyle, D. J. Samuel, T. Conway, and R. R. Klimowski, “Electromagnetic aircraft launch system-EMALS,” *IEEE Trans. Magn.*, vol. 31, no. 1, pp. 528-533, Jan. 1995.
- [112] H. Chen, and Q. Wang, “Modeling of switched reluctance linear launcher,” *IEEE Trans. Plasma Sci.*, vol. 41, no. 5, pp. 1123-1130, Mar. 2013.
- [113] J. Duncan, “Linear induction motor-equivalent circuit model,” *IEE Proc. Electr. Power Appl.*, vol. 130, no. 1, pp. 51-57, Jan. 1983.
- [114] W. Xu, J. Zhu, Y. Zhang, Z. Li, Y. Li, Y. Wang, Y. Guo, and Y. J. Li, “Equivalent circuits for single-sided linear induction motors,” *IEEE Trans. Ind. Appl.*, vol. 46, no. 6, pp. 2410-2423, Sept. 2010.
- [115] J. Y. Lu, and W. M. Ma, “Research on end effect of linear induction machine for high-speed industrial transportation,” *IEEE Trans. Plasma Sci.*, vol. 39, no. 1, pp. 116-120, Jan. 2011.
- [116] W. Xu, G. Sun, G.Wen, Z.Wu, and P. K. Chu, “Equivalent circuit derivation and performance analysis of a single-sided linear induction motor based on winding function theory,” *IEEE Trans. Veh. Techol.*, vol. 61, no. 4, pp. 515-525, Jan. 2012.
- [117] G. Kang, and K. Nam, “Field-oriented control scheme for linear induction motor with the end effect,” *IEE Proc. Elect. Power Appl.*, vol. 152, no. 6, pp. 1565-1572, Nov. 2005.

- [118] D. Hall, J. Kapinski, M. Krefta, and O. Christianson, "Transient electromechanical modeling for short secondary linear induction machines," *IEEE Trans. Energy Convers.*, vol. 23, no. 3, pp. 789-795, Sept. 2008.
- [119] B. I. Kwon, K. I. Woo, and S. Kim, "Finite element analysis of direct thrust controlled linear induction motor," *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 1306-1309, May 1999.
- [120] T. Yamaguchi, Y. Kawase, M. Yoshida, Y. Saito, and Y. Ohdachi, "3-D finite element analysis of a linear induction motor," *IEEE Trans. Magn.*, vol. 37, no. 5, pp. 3668-3671, Sept. 2001.
- [121] M. Poloujadoff, and H. El Khashab, "A finite difference model of linear induction motors, taking into account the finite length of iron," *IEEE Trans. Power App. Syst.*, vol. 101, no. 8, pp. 2966-2974, Aug. 1982.
- [122] S.-M. Jang, Y.-S. Park, S.-Y. Sung, K.-B. Lee, H.-W. Cho, and D.-J. You, "Dynamic characteristic of a linear induction motor for predicting operating performance of magnetic levitation vehicles based on electromagnetic field theory," *IEEE Trans. Magn.*, vol. 47, no. 10, pp. 3673-3676, Oct. 2011.
- [123] B.-J. Lee, D.-H. Koo, and Y.-H. Cho, "Investigation of linear induction motor according to secondary conductor structure," *IEEE Trans. Magn.*, vol. 45, no. 6, pp. 2839-2842, Jun. 2009.
- [124] Q. Lu, Y. Li, Y. Ye, and Z. Q. Zhu, "Investigation of forces in linear induction motor under different slip frequency for low-speed maglev application," *IEEE Trans. Energy Convers.*, vol. 28, no. 1, pp. 145-153, Mar. 2013.
- [125] Y. Zhang, W. Ma, J. Lu, Z. Sun, and J. Xu, "A new approach to research the transverse edge effect in linear induction motor considering the edge fringing flux," *IEEE Trans. Magn.*, vol. 47, no. 11, pp. 4660-4668, Nov. 2011.

- [126] A. Z. Bazghaleh, M. R. Naghashan, and M. R. Meshkatoddini, "Optimum design of single-sided linear induction motors for improved motor performance," *IEEE Trans. Magn.*, vol. 46, no. 11, pp. 3939-3947, Nov. 2010.
- [127] T. Yang, L. Zhou, and L. Li, "Influence of design parameters on end effect in long primary double-sided linear induction motor," *IEEE Trans. Plasma Sci.*, vol. 39, no. 1, pp. 192-197, Jan. 2011.
- [128] A. Shiri, and A. Shoulaie, "Design optimization and analysis of single-Sided linear induction motor, considering all phenomena," *IEEE Trans. Energy Convers.*, vol. 27, no. 2, pp. 516-525, Apr. 2012.
- [129] S. G. Lee, and H.-W. Lee, "Influence of the construction of secondary reaction plate on the transverse edge effect in linear induction motor," *IEEE Trans. Magn.*, vol. 45, no. 9, pp. 2815-2818, Jun. 2009.
- [130] I. Boldea, and S.A. Nasar, *Linear electric actuators and generators*, 1997.
- [131] K.R. Davey, "The equivalent T circuit of the induction motor: Its nonuniqueness and use to the magnetic field analyst," *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1745-1748, Apr. 2007.
- [132] I. Munteanu, A. I. Bratcu, S. Bacha, D. Roye, and J. Guiraud, "Hardware-in- the-loop-based simulator for a class of variable-speed wind energy conversion systems: design and performance assessment," *IEEE Trans. Energy Convers.*, vol. 25, no. 2, pp. 564-576, Jun. 2010.
- [133] F. Gao, B. Blunier, M. G. Simoes, and A. Miraoui, "PEM fuel cell stack modeling for real-time emulation in hardware-in-the-loop applications," *IEEE Trans. Energy Convers.*, vol. 26, no. 1, pp. 184-194, Mar. 2011.
- [134] O. Vodyakho, M. Steurer, C. S. Edrington, and F. Fleming, "An induction machine emulator for high-power applications utilizing advanced simulation tools With graphical user interfaces," *IEEE Trans. Energy Convers.*, vol. 27, no. 1, pp. 160-172, Mar. 2012.

- [135] S. Mojlish, N. Erdogan, D. Levine, and A. Davoudi, "Review of hardware platforms for real-time simulation of electric machines," *IEEE Trans. Transport. Electrification.*, vol. 3, no. 1, pp. 130-146, Mar. 2017.
- [136] T. Garbiec, "Fast computation of performance characteristics for solid-rotor induction motors with electrically inhomogeneous rotors", *IEEE Trans. Energy Convers.*, vol. 31, no. 4, pp. 1688-1696, Dec. 2016.
- [137] M. L. Bash, J. M. Williams, and S. D. Pekarek, "Incorporating motion in mesh-based magnetic equivalent circuits," *IEEE Trans. Energy Convers.*, vol. 25, no. 2, pp. 329-338, Jun. 2010.
- [138] H. Gorginpour, H. Oraee, and R. A. McMahon, "A novel modeling approach for design studies of brushless doubly fed induction generator based on magnetic equivalent circuit," *IEEE Trans. Energy Convers.*, vol. 28, no. 4, pp. 902-912, Dec. 2013.
- [139] S. D. Sudhoff, G. M. Shane, and H. Suryanarayana, "Magnetic equivalent circuit- based scaling laws for low-frequency magnetic devices," *IEEE Trans. Energy Convers.*, vol. 28, no. 3, pp. 746-755, Sep. 2013.
- [140] S. D. Sudhoff, B. T. Kuhn, K. A. Corzine, and B. T. Branecky, "Magnetic equivalent circuit modeling of induction motors," *IEEE Trans. Energy Convers.*, vol. 22, no. 2, pp. 259-270, Jun. 2007.
- [141] G. Y. Sizov, A. Sayed-Ahmed, Y. Chia-Chou, and N. A. O. Demerdash, "Analysis and diagnostics of adjacent and nonadjacent broken-rotor-bar faults in squirrel-cage induction machines," *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4627-4641, Nov. 2009.
- [142] A. Gandhi, T. Corrigan, and L. Parsa, "Recent advances in modeling and online detection of stator interturn faults in electrical motors," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1564-1575, May 2011.
- [143] J. Faiz, S. M. M. Moosavi, M. B. Abadi, and S. M. A. Cruz, "Magnetic equivalent circuit modeling of doubly-fed induction generator with assess-



- ment of rotor inter-turn short-circuit fault indices,” *IET Renew. Power Gener.*, vol. 10, no. 9, pp. 1431-1440, Oct. 2016.
- [144] P. Naderi, and A. Shiri, “Rotor/Stator inter-turn short circuit fault detection for saturable wound-rotor induction machine by modified magnetic equivalent circuit approach,” *IEEE Trans. Magn.*, DOI: 10.1109/TMAG.2017.2672924.
- [145] O. Deblecker, J. Lobry, and C. Broche, “Use of the transmission line modeling method in FEM for solution of nonlinear eddy-current problems,” *Proc. IEE, Sci. Meas. Technol.*, vol. 145, no. 1, pp. 31-38, Jan. 1998.
- [146] A. M. Knight, “Time-stepped eddy-current analysis of induction machines with transmission line modeling and domain decomposition,” *IEEE Trans. Magn.*, vol. 39, no. 4, pp. 2030-2035, Jul. 2003.
- [147] P. C. Krause, O. Wasynczuk, and S. Sudhoff, *Analysis of Electric Machinery and Drive Systems*. Piscataway, NJ, USA: IEEE Press, 1994.
- [148] R. J. Wai, J. D. Lee, and K. L. Chuang, “Real-time PID control strategy for maglev transportation system via particle swarm optimization,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 2, pp. 629-646, Feb. 2011.
- [149] G. Lv, D. Zeng, T. Zhou, and Z. Liu, “Investigation of forces and secondary losses in linear induction motor with the solid and laminated back iron secondary for metro,” *IEEE Trans. Ind. Electron.*, DOI: 10.1109/TIE.2016.2565442.
- [150] F. Eastham, T. Cox, P. Leonard, and J. Proverbs, “Linear induction motors with modular winding primaries and wound rotor secondaries,” *IEEE Trans. Magn.*, vol. 44, no. 11, pp. 4033-4036, Nov. 2008.
- [151] G. Kang, J. Kim and K. Nam, “Parameter estimation scheme for low-speed linear induction motors having different leakage inductances,” *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 708-716, Jul. 2003.

- [152] F. J. Lin, P. K. Huang, and W. D. Chou, "Recurrent-fuzzy-neural-network-controlled linear induction motor servo drive using genetic algorithms," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1449-1461, Apr. 2007.
- [153] F. J. Lin, P. K. Huang, and W. D. Chou, "Motion control of linear induction motor via petri fuzzy neural network," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 281-295, Feb. 2007.
- [154] J. H. Choi, S. Kim, D. S. Yoo, and K. H. Kim, "A diagnostic method of simultaneous open-switch faults in inverter-fed linear induction motor drive for reliability enhancement," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4065-4077, Jul. 2015.
- [155] B. Kou, J. Luo, X. Yang, and L. Zhang, "Modeling and analysis of a novel transverse-flux flux-reversal linear motor for long-stroke application," *IEEE Trans. Ind. Electron.*, vol. 63, no. 10, pp. 6238-6248, Oct. 2016.
- [156] K. N. Gyftakis, J. A. Antonino-Daviu, R. Garcia-Hernandez, M. D. McCulloch, D. A. Howey, and A. J. Marques Cardoso, "Comparative experimental investigation of broken bar fault detectability in induction motors," *IEEE Trans. Ind. Appl.*, vol. 52, no. 2, pp. 1452-1459, Mar. 2016.
- [157] T. Davis, *Direct methods for sparse linear systems. vol. 2.*, Soc. for Industrial Math., 2006.
- [158] T. Davis, I. Duff, P. Amestoy, J. Gilbert, S. Larimore, E. Natarajan, Y. Chen, W. Hager, and S. Rajamanickam, "Suite sparse: a suite of sparse matrix packages," 2013.
- [159] J. Gilbert and T. Peierls, "Sparse spatial pivoting in time proportional to arithmetic operations," *SIAM J. Sci. and Stat. Compu.*, vol. 9, no. 5, pp. 862-874, 1988.
- [160] X. Chen, W. Wu, Y. Wang, H. Yu, and H. Yang, "An escheduler based data dependence analysis and task scheduling for parallel circuit simu-

- lation,” *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 58, no. 10, pp. 702-706, Oct. 2011.
- [161] X. Chen, Y. Wang, and H. Yang, “NICSLU: An adaptive sparse matrix solver for parallel circuit simulation,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 2, pp. 261-274, Feb. 2013.
- [162] K. He, S. X. D. Tan, H. Wang, and G. Shi, “GPU-accelerated parallel sparse LU factorization method for fast circuit analysis,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3, pp. 1140-1150, Mar. 2016.
- [163] T. Nechma, and M. Zwolinski, “Parallel sparse matrix solution for circuit simulation on FPGAs,” *IEEE Trans. Comps.*, vol. 64, no. 4, pp. 1090-1103, Apr. 2015.
- [164] M. O. Karsavuran, K. Akbudak, and C. Aykanat, “Locality-aware parallel sparse matrix-vector and matrix-transpose-vector multiplication on many-core processors,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 6, pp. 1713-1726, Jun. 2016.
- [165] J. Hu, S. F. Quigley, and A. Chan, “An element-by-element preconditioned conjugate gradient solver of 3D tetrahedral finite elements on an FPGA coprocessor,” *International Conference on Field Programmable Logic and Applications*, pp. 575-578, 2008.
- [166] R. Mafi, S. Sirouspour, B. Mahdavihah, B. Moody, K. Elizeh, A. B. Kinsman, and N. Nicolici, “A parallel computing platform for real-time haptic interaction with deformable bodies,” *IEEE Trans. Hapt.*, vol. 3, no. 3, pp. 211-223, Sept. 2010.
- [167] J. Zhang, M. Zhang, H. He, and Q. Song, “Accelerating the finite element method using FPGA for electromagnetic field computation,” *The 5th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, pp. 1763-1768, 2015.

- [168] Y. El-Kurdi, D. Giannacopoulos, and W. J. Gross, "Hardware acceleration for finite-element electromagnetics: efficient sparse matrix floating-point computations with FPGAs," *IEEE Trans. Magn.*, vol. 43, no. 4, Apr. 2007.
- [169] J. P. A. Bastos, and N. Sadowski, "Electromagnetic modeling by finite element methods," CRC press, 2003.
- [170] Q. Lu, Y. Li, Y. Ye, and Z. Q. Zhu, "Investigation of forces in linear induction motor under different slip frequency for low-speed maglev application," *IEEE Trans. Energy Convers.*, vol. 28, no. 1, pp. 145-153, Nov. 2012.
- [171] I. Boldea, "Linear electric machines, drives, and MAGLEVs handbook," CRC press, 2013.
- [172] D. Dyck, T. Rahman, and C. Dufour, "Internally consistent nonlinear behavioral model of a PM synchronous machine for hardware-in-the-loop simulation," *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 1-4, Feb. 2014.
- [173] J. Guzinski, and H. Abu-Rub, "Speed sensorless induction motor drive with predictive current controller," *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 699-709, Feb. 2013.
- [174] A. Hasanzadeh, C. Edrington, N. Stroupe, and T. Bevis, "Real-time emulation of a high-speed microturbine permanent-magnet synchronous generator using multiplatform hardware-in-the-loop realization," *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 3109-3118, Jun. 2014.
- [175] L. Herrera, C. Li, X. Yao, and J. Wang, "FPGA-based detailed real-time simulation of power converters and electric machines for EV HIL applications," *IEEE Trans. Ind. Appl.*, vol. 51, no. 2, pp. 1702-1712, Mar. 2015.

- [176] S. Mojlish, N. Erdogan, D. Levine, and A. Davoudi, "Review of hardware platforms for real-time simulation of electric machines," *IEEE Trans. Transport. Electrification.*, vol. 3, no. 1, pp. 130-146, Mar. 2017.
- [177] F. Alvarez-Gonzalez, A. Griffo, B. Sen, and J. Wang, "Real-time hardware-in-the-loop simulation of permanent magnet synchronous motor drives under stator faults," *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 6960-6969, Sep. 2017.
- [178] L. Wu, and Z. Q. Zhu, "Analytical modeling of surface-mounted PM machines accounting for magnet shaping and varied magnet property distribution," *IEEE Trans. Magn.*, vol. 50, no. 7, pp. 1-11, Jul. 2014
- [179] W. J. R. Hofer, "The transmission-line matrix method - Theory and applications," *IEEE Trans. Microw. Theory Tech.*, vol. 30, no. 10. pp. 882-893, Oct. 1985.
- [180] S. Y. R. Hui, K. Fung, M. Q. Zhang, and C. Christopoulos, "Variable time step technique for transmission line modelling," *IEE Proc. A*, vol. 140, no.4, pp. 299-302, Jul. 1993.
- [181] J. Lobry, J. Trecat, and C. Broche, "The transmission line modelling method as a new iterative technique in nonlinear magnetostatics," *IEEE Trans. Magn.*, vol. 32, no. 2 ,pp. 559-566, Mar. 1996.
- [182] O. Deblecker, J. Lobry, and C. Broche, "Use of the transmission-line modeling method in FEM for solution of nonlinear eddy-current problems," *Proc. IEE, Sci. Meas. Technol.*, vol. 145, no. 1, pp. 31-38, Jan. 1998.
- [183] O. Deblecker, J. Lobry, and C. Broche, "Novel algorithm based on transmission-line modeling in the finite-element method for nonlinear quasi-static field analysis," *IEEE Trans. Magn.*, vol. 39, no. 1, pp. 529-538, Jan. 2003.

# Appendix A

## Device Specifications

The specifications of the single-sided linear induction motor of American Maglev Technology (AMT) used in Chapter 2, 4, and 5 are presented in A.1.

Table A.1: Specifications of the studied LIM.

Length	3.33 m
Width	26.67 cm
Weight	1136 kg
Air gap	1 cm
Number of poles	21
Apparent power	$0.762+j1.187$ MVA
Power factor	0.54
Efficiency	78%
Aluminum sheet loss	87.1 kW
Copper loss	59.98 kW
Stray load loss	0.87 kW
Aluminum conductivity	$3.25 \times 10^7$ S/m

The specifications of the squirrel-cage rotary induction motor used in Chapter 3 are presented in A.2.

The hardware resource of the Xilinx Virtex FPGA boards used in this thesis is presented in A.3

Table A.2: Specifications of the studied rotary induction motor.

Stator outer diameter	195.38 (mm)
Rotor outer diameter	114.9 (mm)
Stator inner diameter	115.54 (mm)
Rotor inner diameter	36.5 (mm)
Stator slot depth	21.1 (mm)
Stator tooth width	5.7 (mm)
Rotor slot depth	22.1 (mm)
Rotor tooth width	6.2 (mm)
Stator tooth face width	7.4 (mm)
Rotor tooth face width	12.7 (mm)
Motor length	107.95 (mm)
Stator tooth flange thickness	1.2 (mm)
Rotor tooth flange thickness	1.8 (mm)
Mechanical air-gap length	0.31 (mm)
Number of stator slots	36
Number of rotor slots	28
Rated voltage	230 (V)
Frequency	60 (Hz)
Number of poles	4
Rotor inertia	0.025 $Kg.m^2$
Winding connections	Wye
Rotor type	Squirrel cage

Table A.3: Hardware resource utilization

Resource	Virtex-7 XC7VX485T	Virtex UltraScale+ XCVU9P
System Logic Cells (K)	485	2,586
DSP Slices	2,800	6,840
Memory (Mb)	37	345.9
I/O Pins	700	832