**2D Hybrid Magnetic Field Model Performance Optimization for Linear Induction Motor**

By

**Michael Thamm**

A Thesis

Submitted to the Faculty of Graduate Studies

through the Department of Electrical & Computer Engineering

in Partial Fulfillment of the Requirements for

the Degree of Master of Applied Science

at the University of Windsor

Windsor, Ontario, Canada

2022

© 2022 Michael Thamm

2D Hybrid Magnetic Field Model Performance Optimization for Linear Induction Motors

by

**Michael Thamm**

APPROVED BY:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

J. Urbanic

Department of Mechanical, Automotive & Materials Engineering

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

M. S. Toulabi

Department of Electrical & Computer Engineering

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

N. C. Kar, Advisor

Department of Electrical & Computer Engineering

May TBD, 2022

# DECLARATION OF CO-AUTHORSHIP/PREVIOUS PUBLICATION

I hereby declare that this thesis incorporates material thatis result of joint research, as follows:

This thesis contains the outcomes of publications which include the contributions of co-authors who were/are post-doctoral fellows, graduate students or associate professors under the supervision of Dr. Narayan C. Kar. In all cases, only my primary contributions towards these publications are included in this thesis. The contribution of co-authors was primarily with respect to refinement and editing process. In Chapter 2, I was the co-author in which I was actively part of experimental testing and assisted in data analysis. The model developed by the primary author, A. Fatima, in this publication is used by the proposed method and is therefore described in this chapter. Chapter 5, I was the co-author in which I applied the proposed method to predict dynamic performance characteristics. Only the sections with my personal contribution are included in this thesis to analyze the performance of the proposed method described in this thesis.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis. I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work. This thesis includes three original papers that have been previously published/submitted to journals for publication, as follows:

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis. I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

# ABSTRACT

This paper analyzes the domain of double-layer, single-sided, 3-phase, integral slot winding, linear induction motors. Motor meta-parameters such as slots and poles are difficult to optimize since they drastically effect the configuration of the motor and require general optimization implementations.

The NSGAII multi-objective optimization algorithm within the Platypus-Opt Python library proved to be robust, yet flexible, while maximizing thrust and minimizing mass of motor iterations. Each iteration was accurately modelled using the HAM producing the necessary performance parameters for the NSGAII’s objective function. Field plotting capability of the processed HAM allowed for the feasibility check on post-processing constraints increasing the robustness of the optimization.

Validation between HAM to FEA and HAM to the baseline proved the accuracy of the modelling algorithm within the objective function. The optimization concluded that the optimal motor had 36 slots and 4 poles within the domain of [12 slots, 54 slots] & [4 poles, 12 poles], where 9 feasible motors were objectively compared.

Using this automated optimization tool saved significant time and effort while generating reproducible results within a constrained domain. The entire optimization completed in 5 minutes whereas the total time for configuring all motors within the domain took 4.5 hours, proving its worth.

# DEDICATION

I would like to express sincere gratitude towards Dr. Narayan Kar who contributed in many ways to “get the monkey off my back” and complete this Ma.Sc. journey. Whether I required financial or academic support Dr. Kar was always optimistic and willing to progress research within the lab. This created a competitive and supportive lab environment within CHARGE labs that I was lucky enough to take part in. The critical feedback I received from my thesis committee members M. S. Toulabi and Jill Urbanic was constructive in nature and challenged my thesis at its core. I am grateful to have worked with such cooperative and supportive committee members.

Throughout the countless hours of work, I grew as an academic and a teammate among many talented and supportive lab members who have helped to make this journey possible. I would like to thank Shruti M, Hima D, and Aida M. for being the first few lab members to drive my initial understanding of electric motor theory and simulation in a masters setting. Lastly, I would like to thank the lab members that were there for me throughout most of my journey Brad S, Tim S, and Areej F. Without these supportive people I would not be publishing this paper and I am forever grateful for these amazing friends.

# ACKNOWLEDGEMENTS

The rollercoaster of emotions and stress that comes with the thesis journey become worthwhile only if shared with the most important people in your life. Solange, Isabelle, Heike, Uwe I love you all so much for the patience and openness you have shown me throughout the long years of my work and MaSc career. I am proud to have such amazing family on my side and I will always return the favour without hesitation.

# TABLE OF CONTENTS

[DECLARATION OF CO-AUTHORSHIP/PREVIOUS PUBLICATION iii](#_Toc124946827)

[ABSTRACT v](#_Toc124946828)

[DEDICATION vi](#_Toc124946829)

[ACKNOWLEDGEMENTS vii](#_Toc124946830)

[TABLE OF CONTENTS viii](#_Toc124946831)

[LIST OF TABLES xi](#_Toc124946832)

[LIST OF FIGURES xiv](#_Toc124946833)

[LIST OF ABBREVIATIONS xvii](#_Toc124946834)

[NOMENCLATURE xviii](#_Toc124946835)

[CHAPTER 1 Introduction 20](#_Toc124946836)

[1.1. Electric Vehicles–A Green Alternative 20](#_Toc124946837)

[1.2. Motor Slot and Pole Count - Winding Configurations 21](#_Toc124946838)

[1.3. Literature Survey on Motor Modelling 23](#_Toc124946839)

[1.4. Induction Motor Optimization 25](#_Toc124946840)

[1.5. Research Motivations 26](#_Toc124946841)

[1.6. Research Objectives 27](#_Toc124946842)

[1.7. Research Contribution and Deliverables 28](#_Toc124946843)

[1.8. Organization of Thesis 29](#_Toc124946844)

[CHAPTER 2 Hybrid Analytical Model 31](#_Toc124946845)

[2.1. Base Model 31](#_Toc124946846)

[2.2. Model Relationships 35](#_Toc124946847)

[2.3. Winding Distribution Table 37](#_Toc124946848)

[2.4. Hybrid Analytical Model Structure 40](#_Toc124946849)

[2.5. System of Linear Equations 49](#_Toc124946850)

[2.6. Processed Model 53](#_Toc124946851)

[CHAPTER 3 Optimization Algorithm 56](#_Toc124946852)

[3.1. Genetic Algorithm 56](#_Toc124946853)

[3.2. Particle Swarm Optimization 58](#_Toc124946854)

[3.3. Schwefel Function Minimization Case Study 60](#_Toc124946855)

[3.4. NSGAII Configuration 70](#_Toc124946856)

[3.4.1. Solver Selection 71](#_Toc124946857)

[3.4.2. Solver Variation 72](#_Toc124946858)

[CHAPTER 4 Model Optimization Integration 75](#_Toc124946859)

[4.1. Optimization Constants 76](#_Toc124946860)

[4.2. Motor Feasibility 77](#_Toc124946861)

[4.3. Compute Fitness 80](#_Toc124946862)

[4.4. Baseline Validation 82](#_Toc124946863)

[4.4.1. Air Gap B field and Force Validation 82](#_Toc124946864)

[4.4.2. Motor Core B Field Plot Validation 84](#_Toc124946865)

[4.5. Solver Configuration 86](#_Toc124946866)

[4.6. Optimal Motor 90](#_Toc124946867)

[CHAPTER 5 Research Summary 91](#_Toc124946868)

[5.1. Future Research on HAM and LIM Optimization 92](#_Toc124946869)

[5.1.1. Constrained Optimization Domain 92](#_Toc124946870)

[5.1.2. Winding feasibility using WDT 93](#_Toc124946871)

[5.1.3. ANSYS Implementations 93](#_Toc124946872)

[5.2. Conclusion 94](#_Toc124946873)

[REFERENCES 95](#_Toc124946874)

[VITA AUCTORIS 100](#_Toc124946875)

# LIST OF TABLES

[Table 1.1 21](#_Toc124946876)

[Slot and Pole Trend Decision-Making 21](#_Toc124946877)

[Table 1.2 23](#_Toc124946878)

[Modelling Algorithm Comparison 23](#_Toc124946879)

[Table 2.1 32](#_Toc124946880)

[Baseline Spatial Motor Parameters 32](#_Toc124946881)

[Table 2.2 33](#_Toc124946882)

[Baseline Electrical And Material Motor Parameters 33](#_Toc124946883)

[Table 2.3 34](#_Toc124946884)

[Baseline Model Parameters 34](#_Toc124946885)

[Table 2.4 38](#_Toc124946886)

[Order of the WDT Elements 38](#_Toc124946887)

[Table 2.5 39](#_Toc124946888)

[WDT of Sample Motor 39](#_Toc124946889)

[Table 2.6 42](#_Toc124946890)

[Node Index Continuity 42](#_Toc124946891)

[Table 2.7 51](#_Toc124946892)

[System of Linear Equations Solving for Unknown Variables 51](#_Toc124946893)

[Table 2.8 53](#_Toc124946894)

[Boundary Condition Summary 53](#_Toc124946895)

[Table 3.1 60](#_Toc124946896)

[PSO Velocity and Position Coefficients 60](#_Toc124946897)

[Table 3.2 63](#_Toc124946898)

[Optimization Algorithm Configuration 63](#_Toc124946899)

[Table 3.3 63](#_Toc124946900)

[Average Optimization Algorithm Results 63](#_Toc124946901)

[Table 3.4 66](#_Toc124946902)

[Algorithm Convergence Visualization 66](#_Toc124946903)

[Table 4.1 76](#_Toc124946904)

[Constant Motor Parameters 76](#_Toc124946905)

[Table 4.2 77](#_Toc124946906)

[Model Parameters 77](#_Toc124946907)

[Table 4.4 77](#_Toc124946908)

[Motor Feasibility Constraints 77](#_Toc124946909)

[Table 4.5 81](#_Toc124946910)

[Mass Equations 81](#_Toc124946911)

[Table 4.6 86](#_Toc124946912)

[Optimization Configuration 86](#_Toc124946913)

[Table 4.7 89](#_Toc124946914)

[Pareto Front Analysis 89](#_Toc124946915)

[Table 4.8 90](#_Toc124946916)

[Optimal Motor Parameters 90](#_Toc124946917)

[Table 4.9 91](#_Toc124946918)

[Optimal Model Parameters 91](#_Toc124946919)

# LIST OF FIGURES

[Fig. 2.1. Magnetic flux density in the middle of the airgap in the normal direction comparing the proposed model results against FEA using ANSYS Electronics (Ip = 10 A, v = 0 m/s, f = 100 Hz) [20]. 34](#_Toc124946920)

[Fig. 2.2. Magnetic flux density in the middle of the airgap in the tangential direction comparing the proposed model results against FEA using ANSYS Electronics (Ip = 10 A, v = 0 m/s, f = 100 Hz) [20]. 35](#_Toc124946921)

[Fig. 2.3. Meshed motor model containing boundary conditions. 41](#_Toc124946922)

[Fig. 2.4. Single MEC node element at index . 43](#_Toc124946923)

[Fig. 2.5. MEC modelling of 4 arbitrary neighbouring nodes within 3 different arbitrary materials (yellow, blue, red). 44](#_Toc124946924)

[Fig. 2.6. MMF scaling factor per node in the y-direction of a coil for a) coil in upper slot and b) coil in lower slot of the primary core [20]. 46](#_Toc124946925)

[Fig. 2.7. Complex Fourier Series approximation of the discrete B field at the center of the airgap for various harmonics. 55](#_Toc124946926)

[Fig. 3.1. Layout of a genetic algorithm with an arbitrary number of chromosomes and genes per population. 56](#_Toc124946927)

[Fig. 3.2. Layout of a genetic algorithm execution loop. 57](#_Toc124946928)

[Fig. 3.3. Layout of a particle swarm optimization algorithm optimization loop. 59](#_Toc124946929)

[Fig. 3.4. Surface plot of the Schwefel function on the input range. 61](#_Toc124946930)

[Fig. 3.5. Contour plot of the Schwefel function on the input range highlighting the global minimum with a red cross. 62](#_Toc124946931)

[Fig. 3.6. Comparison of the average solver execution time between GA and PSO until 25 stall iterations are achieved using the Schwefel test function at different artificial objective function execution times. 70](#_Toc124946932)

[Fig. 3.7. Layout of a Tournament selection algorithm using arbitrary objective values to highlight the winning decision based on a minimization problem. 72](#_Toc124946933)

[Fig. 3.8. Visualization of crossover between two parent variables to produce two child variables governed by the crossover point. 73](#_Toc124946934)

[Fig. 3.9. Layout of a genetic algorithm with an arbitrary number of chromosomes and genes per population. 74](#_Toc124946935)

[Fig. 4.1. Motor optimization algorithm state chart for hybrid analytical modelling. 75](#_Toc124946936)

[Fig. 4.2. Skin depth (blue) in the secondary aluminum plate, with increased resistivity to account for the transverse end-effects, including the plate thickness (orange). 79](#_Toc124946937)

[Fig. 4.3. Layout of the motor optimization algorithm inputs and the resultant multi-objectives. 80](#_Toc124946938)

[Fig. 4.4. Tangential B field in the center of the air gap comparison between Ansys Electronics FEA (light blue), reference paper results (red), and the HAM produced in this paper (dark blue). 82](#_Toc124946939)

[Fig. 4.5. Normal B field in the center of the air gap comparison between Ansys Electronics FEA (light blue), reference paper results (red), and the HAM produced in this paper (dark blue). 83](#_Toc124946940)

[Fig. 4.6. Magnetic flux density (B) field plot in the motor core of the HAM simulation for the baseline motor to validate accuracy. 84](#_Toc124946941)

[Fig. 4.7. Magnetic flux density (B) field plot in the motor core of the Ansys Electronics simulation for the baseline motor to validate accuracy. 85](#_Toc124946942)

[Fig. 4.8. Transient tangential (Fx) and normal (Fy) force plot of all motors evaluated during the optimization process calculated as a steady state average between the time interval of 48 to 60 ms. 87](#_Toc124946943)

[Fig. 4.9. Average steady state thrust plot with data point annotations in the format: (Ns, Np) grouped by the x-axis into columns of q values. 88](#_Toc124946944)

[Fig. 4.10. Pareto plot of the objectives: motor mass (x-axis) and average steady state thrust (y-axis), with data point annotations in the format: (Ns, Np) note the legend colours. 89](#_Toc124946945)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| LEM | Linear Electric Motor |
| REM | Rotary Electric Motor |
| LIM | Linear Induction Motor |
| LSM | Linear Synchronous Motor |
| WDT | Winding Distribution Table |
| HAM | Hybrid Analytical Model |
| HM | Harmonic Model |
| MEC | Magnetic Equivalent Circuit |
| ECM | Equivalent Circuit Model |
| FEA | Finite Element Analysis |
| OA | Optimization Algorithm |
| EA | Evolutionary Algorithm |
| NN | Neural Network |
| PSO | Particle Swarm Optimization |
| OMOPSO | Optimized Multi-Objective Particle Swarm Optimization |
| GA | Genetic Algorithm |
| NSGAII | Non-Dominated Sorting Genetic Algorithm II |
|  |  |
|  |  |
|  |  |
|  |  |

# NOMENCLATURE

|  |  |
| --- | --- |
| **Variable** | **Description** |
|  | Number of slots |
|  | Number of poles |
|  | Number of phases |
|  | Synchronous velocity |
|  | Electrical frequency |
|  | Peak current |
|  | Phase current |
|  | Number of turns per coil |
|  | Magnetomotive force scaling factor |
|  | Number of nodes in the x-direction for a single coil |
|  | Slots/poles/phase |
|  | Conductivity |
|  | Vacuum permeability |
|  | Relative permeability of a node in the HM region |
|  | Relative permeability of a node in the MEC region |
|  | Slot height |
|  | Yoke height |
|  | Tooth width |
|  | Slot width |
|  | Slot pitch |
|  | Airgap |
|  | Aluminum thickness |
|  | Back iron thickness |
|  | Primary length |
|  | Primary height |
|  | Primary depth |
|  | Periodical length of model |
|  | Space harmonic |
|  | Number of space harmonics |
|  | Spatial frequency for nth space harmonic |
| *,* | Complex harmonic analysis unknowns for nth space harmonic |
|  | Spatial position in the x-direction |
|  | Spatial position in the y-direction |
|  | Number of harmonic model regions in the model |
|  | Number of magnetic equivalent circuit regions in the model |
|  | y-index of a node in the magnetic equivalent circuit region |
|  | x-index of a node in the magnetic equivalent circuit region |
|  | Number of rows in a magnetic equivalent circuit region |
|  | Number of columns in a magnetic equivalent circuit region |
|  | Total nodes in a magnetic equivalent circuit region |
|  | Reluctance |
|  | Magnetomotive force |
|  | Flux |
|  | Complex scalar potential |
|  | Surface Area |
|  | Magnetic flux density |
|  | Magnetic field |
|  | Skin depth of magnetic field in material |
|  | Number of columns in a WDT table |
|  | Coil pitch of a double-layer winding |
|  | Pole pitch of the motor |

# Introduction

## Electric Vehicles–A Green Alternative

In 2009 the European commission for science and environmental policy [1] stated that the world must not exceed the 1 trillion carbon budget to avoid a 2 degree rise in the world’s average climate. Through a combined effort of all countries across the world human civilization has introduced 6.36 trillion tonnes of emissions from the late 1800s to 2020 according to [2]. The impacts of this accelerated rate of output has serious implications on the health of the world’s ecosystems and is quickly becoming a concern across economic and geopolitical conversations. Since global transportation accounts for 37% of CO2 emissions from end‐use sectors [3], it is imperative that there be an initiative which can alleviate some of this contribution. According to the government of Canada [4] the efficiency of energy conversion from on-board storage to turning the wheels is nearly five times greater for electricity than gasoline, at approximately 76% and 16%, respectively. If this data were scaled to the number of combustion engines that exist globally, the lost potential and the environmental impact becomes clear. One of the most efficient ways to travel is via high-speed electric train due to ride sharing and efficiency which could drastically reduce the global carbon footprint of transportation if it were the primary means for transportation.

The Siemens Velaro D (DB Class 407) high-speed electric train [5] is designed for operation at 320 km/h with an output power of 8 MW. Since these trains can span hundreds of meters, 16 motors were distributed across the train cars each producing 500 kW – 600 kW. The class 407 trains were first operational in 2013 and are setting an example for the importance of efficient high-speed electric motors. In contrast the Swissloop team produced a double-sided linear induction motor electric train prototype that can achieve a top speed of 252 km/h at 250kW of output power [6]. Since rotary electric motor applications require mechanical traction, they experience mechanical losses and complexity which are not applicable to LEMs. Therefore, it is optimal to select LEMs when a linear force is required and REMs when a torque is required due to the minimization of lost energy during energy transfer. LEMs are commonly used in precise, high-acceleration applications like actuators and in high-speed, low-acceleration systems like electric trains. With careful design considerations, a combination of speed, thrust, and efficiency can be achieved to meet the application’s design objectives.

## Motor Slot and Pole Count - Winding Configurations

The linear electric motor’s slot and pole count is an important electromagnetic relationship that determines the resultant magnetic field waveform found in the airgap of the motor application. To achieve greater efficiency, the generated wave in the primary field shall approximate a sin wave. This approach is called distributed winding and is achieved through different slot and pole combinations in the primary.

Table 1.1

Slot and Pole Trend Decision-Making

|  |  |  |
| --- | --- | --- |
| **Increase Slots** | Advantage | * Better primary field sin wave approximation resulting in improved efficiency * Reduced mass due to the metal core being denser than the combination of copper and insulation in the slots |
| Disadvantage | * Complex to manufacture and wind the coils * Localized saturation of the primary core if the primary tooth width is too low |
| **Increase Poles** | Advantage | * Improved operating efficiency * Reduced mass |
| Disadvantage | * Increased eddy current losses * Reduced maximum speed * Reduced thrust force |

The fewer pole pairs a motor has, the less drag the motor experiences and therefore the more thrust it generates [7], [8]. As the slot count of the motor increases, the discretization effect of the resultant magnetic field waveform found in the airgap decreases, approaching a sine wave resemblance. However, the effect is lost if the slot count is excessively increased causing the field to approach a triangular waveform. To combat this, the winding in each slot can be split into two different coils, which may carry a different electrical phase, and then one layer of coils is shifted. This allows for the waveform to retain its curvature like that of the sine wave.

There are many combinations of winding patterns when considering various slot and pole counts. It is important that there be a governing set of rules which all winding configurations abide by to ensure that each implementation is feasible and effective. A winding distribution table (WDT) serves this purpose which is detailed in [9] providing a robust definition of its approach. The general principle is that the WDT balances the slot EMFs over the phases, creating symmetrical windings.

## Literature Survey on Motor Modelling

When choosing a suitable modelling workflow for a motor optimization problem, many constraint considerations must be made. Considering optimization efficiency, robustness, integration complexity, and flexibility when choosing the best modelling algorithms within a workflow proves to be challenging. Within table 1.2 a comparison between FEA, MEC, and HM techniques is provided to aid in the decision-making process.

Table 1.2

Modelling Algorithm Comparison

|  |  |  |
| --- | --- | --- |
| **FEA** | **Advantage** | * Modular modelling capability which can be extended to many fields of physics like electromagnetics, thermodynamics, and mechanics * Efficient machine code and modelling techniques produces the most accuracy within reasonable time * Accurately models both magnetic and electrical losses under transient conditions |
| **Disadvantage** | * High computation demand for dense-mesh, transient simulations * The freedom in designing and optimizing a model may be limited to the software’s capability * Difficult to automate an optimization workflow due to lack of customization |
| **MEC** | **Advantage** | * Flexible modelling methodology which can equate a large range of models within a domain * Accurately models complex geometries with a relatively dense mesh |
| **Disadvantage** | * Discretization of the spatial domain requires dense meshing to produce an accurate solution * Errors in the solution occur near abrupt changes in source potential |
| **HM** | Advantage | * The computation intensity only scales with harmonics, not the number of nodes in the mesh of the region * Since the region does not require discretization, the solution can be calculated at any spatial position within the region rather than at the center of a node |
| Disadvantage | * Inaccurately models complex geometries which are common in motor applications such as motor teeth and windings * Requires many harmonics to accurately predict the waveform |

Although FEA is often known as the ultimate modelling application due to its accurate modelling ability, it is often not an efficient medium for custom optimization problems. This is due to the lack of access to the back-end code resulting in the user having to conform to the functionality provided by the application itself. When creating custom modelling algorithms, this constraint is relieved and is often preferred when the optimization is in the intermediary development phases. Once the custom algorithm has narrowed the design space via its convergence on an optimal solution, then it is beneficial for modelling final designs in FEA. Hybrid analytical models [HAM] are a merger of multiple modelling techniques which utilize the advantages of each individual technique in regions within the model. For example, when merging MEC modelling with HM the resulting model achieves the advantages of each found in table 1.2. This allows complex geometries to be accurately modelled using MEC regions and simple geometries with HM regions to achieve greater efficiency and accuracy.

## Induction Motor Optimization

The classification of optimization algorithms can be summarized into 3 distinct categories: Gradient based, evolutionary, and neural network. Two elite algorithms within each subset of the categories are highlighted in figure x which serve unique purposes in optimization workflows. The simplest of the 3 is the gradient-based algorithm which requires function evaluations to determine rate of improvement towards the objective. This often leads to finding local minima and maxima rather than the global counterparts since the algorithm will return to the local minimum/maximum once the gradient direction becomes negative.

Diagram

Description automatically generated

Alternatively, evolutionary algorithms and neural networks are heuristics which means they weigh solutions based on their prediction of the solution’s future performance. Although neural networks are the most robust, they tend to be computationally intensive in addition to the difficulty of deciphering how the algorithm determines its solution. Evolutionary algorithms find an equilibrium between the performances of gradient-based algorithms and neural networks which makes them an attractive prospect for intermediate optimization problems.

In addition to custom implementations of optimization algorithms, there exists modern software applications which can achieve many of the same things. Examples of these are Optimetrics optimization [10] and PyAEDT [11] integration within ANSYS Electronics (a modern FEA software useful for electric motor modelling). The benefit with these existing applications is that they act as algorithms wrapped around the ANSYS modelling software. The main problem with Optimetrics is that iterative slot-pole combination changes the geometry of the motor model, making it hard to implement such a general application. Alternatively, “PyAEDT is a Python library that interacts directly with the AEDT API to make scripting simpler for the end user” [11]. This greatly increases the flexibility of the application since python has many modules for programming applications. Although this seems like the ideal implementation framework, it takes extensive effort to takes time to thoroughly investigate the feasibility of integrating a complex solution with an API like PyAEDT. Additionally, the productivity of the project would be dependent on the availability and robustness of the code base which is non-existent for a custom implementation like the one in this paper. Lastly, APIs often have an associated cost to make calls which makes a custom implementation desirable.

## Research Motivations

Many works have implemented motor optimization for different design goals for example: primary weight [12], maximizing the thrust and power to weight ratio [13], optimal winding design of LIM [14] and [15]. In other research, efficiency and power factor was maximized [16] and [17], in addition to the imperialist competitive algorithm implemented for SLIM design [18]. The proposed solution in [19] attempts to improve upon the papers referenced above via multi-objective, genetic algorithm optimization. Although these research references improved their designs via narrow optimization, they often do not consider enough of the optimization domain.

To effectively improve motor performance, it is important to optimize motor meta parameters before tuning lower-level parameter optimizations. Meta parameters are the first design considerations when designing a new motor which have a rippling effect on other motor parameters e.g., motor slot and pole counts are possible meta parameters. From the performance trends found in table 1.1 it is not intuitive to predict an optimal slot-pole combination due to the advantages and disadvantages carrying similar weight. It is known that a greater slot count improves the efficiency and reducing the poles improves the thrust force. An intuitive guess would predict that a 36 slot, 2 pole motor produces the best motor performance. Although the solver may trend towards a solution like this, there are disadvantages to a design at this extreme of the motor domain. For this reason, it is important to introduce an iterative evolutionary algorithm which may predict better designs than a manual design workflow can produce. After simulating hundreds of feasible motor models, the solver shall navigate the motor domain enough to record correlations between the slot-pole combination and the performance parameters.

## Research Objectives

The base model found in [20] acts as ground truth for the HAM model reproduced in this paper. Since the motor was modeled using HAM and its results were within 1.7% compared to FEA and were verified by static measurements, it is an acceptable model for the application of this thesis. Optimizing across an entire sub-category of linear induction motors is achievable using HAM and an optimization workflow described in chapters 2 and 3 respectively. It is not sufficient to produce a motor that outperforms its alternatives in one performance parameter while sacrificing other performance parameters. Therefore Pareto-optimal solutions that outperform the alternative motors in most performance parameters, if not all, are required. To determine that motors produced via the optimization process are feasible in all aspects, including consequential constraints such as saturation effect, robust feasibility checks are added to the solver (detailed in chapter 4.2). To ensure the accuracy of the model, the results are objectively compared to FEA via transient electromagnetic analysis in the software ANSYS Electronics.

## Research Contribution and Deliverables

From previous linear induction motor optimization research highlighted in section 1.5, it is feasible to produce significant improvement in one performance parameter such as weight, thrust, efficiency, and power factor. However, when conducting narrow optimization there is unrealized potential. To account for this, additional narrow optimizations can be conducted further optimizing the motor. This will likely reduce other performance parameters by conflicting with other optimizations if they are not mutually exclusive. ***In this thesis a novel holistic model is proposed which includes flexible motor modelling, modified constraints, multi-objective optimization, and field plotting to serve as a design tool to automate producing optimal motor designs within their constrained domain***. The expected deliverables include:

* A custom optimization workflow that is configurable and expandable for future optimization studies.
* Produce the optimal motor within the subclass of double-layer, single-sided, 3-phase, integral slot winding, linear induction motors.
* Less than 5% HAM modelling error compared to ANSYS Electronics FEA simulation results.

The modelling technique for HAM is complex to integrate and program while achieving efficient execution times of the program written to optimize this motor problem. When using Python 3, the extensive libraries related to optimization and data collection make it a prime medium for tying all pieces of the optimization workflow together. Using Scipy for the system of linear equations, Platypus-opt for the optimization algorithm, and the graphical framework Tkinter for data visualization allows for field plots and transient responses to be visualized within the application.

## Organization of Thesis

The major sections of this thesis are as follows:

1. Chapter 1 provides an overview of LEMs and the use of OAs with induction machines, demonstrating the motivations, challenges and objectives associated with the proposed method from a vehicle level to the motor level and the incorporation of the algorithm level.
2. The baseline double-layer single-sided LIM considered for optimization is introduced in chapter 2, outlining its performance parameters and constraint considerations. The modelling methodology to be implemented on the motor will be discussed in detail within this chapter.
3. This chapter includes a case study on the proposed optimization algorithms within the class of EAs, conducted to determine the optimal multi-objective algorithm for implementation on the modelling algorithm described in chapter 2.
4. Chapter 4 serves to elaborate on the integration of the chosen optimization algorithm from chapter 3 with the modelling algorithm from chapter 2. Efficiency and robustness of the solver as well as solver configuration constraints will be discussed.
5. Chapter 5 summarizes the results generated through the proposed method and identifies the future scope of the proposed research and developed method in IMs and algorithm-based IM optimization.

# Hybrid Analytical Model

The concept of modelling domains to predict the behaviour of materials and waveforms has drastically improved one’s ability to rapid prototype designs with a significant reduction in cost. To achieve this efficiency in the design phase of any project, a modelling algorithm is required that can accurately and timely predict the domain through a system of equations. Finite element analysis is generally a good application of modelling and has been implemented across topics like fluid dynamics, wave propagation, thermal transport, and generally anything that can be governed by a system of mathematical equations. Although FEA is the standard for accuracy, it can be time and computationally intensive, leaving researchers with optimization strategies needing more specific solutions. To achieve this capability, it is important to understand the fundamentals of the boundaries that constrain the domain and the equations that govern the space, defined by the problem. This section will define the baseline motor used as a reference to compare future motor solutions against and elaborate on the HAM functionalities required to achieve a processed solution after solving the system of linear equations defined by the pre-processed model.

## Base Model

For any optimization procedure it is important to define a reference model that acts as the previous standard for the given problem. For optimizing electric machines this is often called a baseline motor and will be classified as a linear induction motor. A double-layer single-sided LIM was modelled, manufactured, and tested in [20], which will serve as the baseline motor for the optimization problem defined within this paper. The specific mechanical, electric, and material properties of the baseline motor are tabulated in tables 2.1, 2.2, and 2.3 respectively. To account for transverse end-effects in the motor, the conductivity of the aluminum plate was reduced accordingly in table 2.2. The number of node elements contained in rows and columns presented in table 2.3 allows the magnetic field in the primary of the motor and in the surrounding air to be accurately modelled.

Table 2.1

Baseline Spatial Motor Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Spatial** | Pole pitch (mm) |  | 45 |
| Slot height (mm) |  | 20 |
| Yoke height (mm) |  | 6.5 |
| Tooth width (mm) |  | 6 |
| End Tooth width (mm) |  | 10 |
| Slot width (mm) |  | 10 |
| Slot pitch (mm) |  | 16 |
| Airgap (mm) |  | 2.7 |
| Aluminum thickness (mm) |  | 2 |
| Back iron thickness (mm) |  | 8 |
| Primary length (mm) |  | 270 |
| Primary height (mm) |  | 26.5 |
| Primary depth (mm) |  | 50 |
| Periodical length of model (mm) |  | 525 |

Table 2.2

Baseline Electrical And Material Motor Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Electrical** | Number of slots |  | 16 |
| Number of poles |  | 6 |
| Number of phases |  | 3 |
| Synchronous velocity (m/s) |  | 0 |
| Electrical frequency (Hz) |  | 100 |
| Peak current (A) |  | 10 |
| Number of turns per coil |  | 57 |
| **Material** | Aluminum Conductivity |  |  |
| Iron Conductivity |  |  |
| Relative Permeability of Iron |  | 1000 |

Table 2.3

Baseline Model Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | Number of space harmonics |  | 100 |
| Number of HM regions in the model |  | 5 |
| Number of MEC regions in the model |  | 1 |
| Number of rows in a magnetic equivalent circuit region |  | 53 |
| Number of columns in a magnetic equivalent circuit region |  | 576 |

To validate a model’s accuracy, the resulting magnetic flux density in the normal and in longitudinal direction will be plotted against the steady state FEA solution. This will be done in ANSYS Electronics using the same configuration found in tables 2.1, 2.2, 2.3.



Fig. 2.1. Magnetic flux density in the middle of the airgap in the normal direction comparing the proposed model results against FEA using ANSYS Electronics (Ip = 10 A, v = 0 m/s, f = 100 Hz) [20].



Fig. 2.2. Magnetic flux density in the middle of the airgap in the tangential direction comparing the proposed model results against FEA using ANSYS Electronics (Ip = 10 A, v = 0 m/s, f = 100 Hz) [20].

## Model Relationships

There are many important relationships between the motor parameters which can be utilized to assign ratios between variables. Creating relationships between variables allows for more flexibility in the model which tends to produce feasible motor designs. This is an important step which constrains the complexity of the optimization space while improving its robustness. A summation of the individual slot and tooth lengths produces equation (1), to calculate the length of the LIM primary. Since the length of the motor primary is a constant design parameter, it is useful to determine the slot and tooth widths from a given length input with a varying slot input.

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

One important relationship is between the slot and tooth width of the primary core. Since saturation degrades the motor performance in teeth that do not provide enough volume for the flux, the tooth width must not be too small in a motor design. Alternatively, the slot width should not be made too small producing unrealized potential. A relationship where the tooth width is 60% the width of a slot produces a ratio that will work for a large range of motors that vary in their slot and pole combination.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Increasing the width of the end teeth helps alleviate some of the end-effects by capturing more of the magnetic field. Consequently, the overall thrust produced increases due to the addition of more active surface area to the end teeth, which tend to saturate faster than internal motor teeth. Using the relationship in equation (4), the end tooth width is equal to the slot width.

|  |  |  |
| --- | --- | --- |
|  |  | () |

The substitution of equations (2) - (4) into equation (1) results in only one unknown variable, . After factoring and isolation, the slot width is solved in equation (5) which can then be substituted back into the other equations to solve for tooth width, slot pitch, and end tooth width.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Through this approach the modelling of the motor maintains a constant primary length while varying the motor configuration for varying slot counts. Although these relationships are necessary for the model implementation, they are not optimal in the final motor design. Further low-level motor optimization must take place to fine tune the LIM primary geometry which will maximize the flux in the core without producing saturation.

Following the robust relationship between slot count and motor geometry, the relationship between magnetic poles and the motor performance is defined in equation (6). For a constant frequency the velocity is proportional to the pole pitch which is approximated in equation (7) for a given motor primary length. Since the motor application demands a high operational velocity the pole pitch shall be maximized.

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

All motors windings considered in this paper are double-layer demanding a flexible relationship for the coil pitch. In most cases the coil pitch is defined by the relationship to the pole pitch defined in equation (8) for a shortened pitch employed for the limitation of both the 5-th and 7-th harmonics).

|  |  |  |
| --- | --- | --- |
|  |  | () |

The coil pitch is implemented within the WDT which is detailed in section 2.3.

## Winding Distribution Table

Upon every iteration of the optimization loop, a slot-pole combination will be provided to the objective function which is subject to the model relationships within section 2.2 and then assigned to the HAM to solve for performance parameters. To handle a wide variety of integral slot winding patterns and maintain geometric integrity, the WDT [9] is implemented as the formulation for the winding configuration.

|  |  |  |
| --- | --- | --- |
|  |  | () |

In summary, the WDT is composed of rows equal to the number of phases and columns equal to the number of slots divided by phases. In case the winding has multiple layers, each cell in the table has their opposite negative (or positive) coil terminal defined by the coil pitch .

Table 2.4

Order of the WDT Elements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Col 1 | Col. 2 | … | Col. |
| Row 1 | 1 | 2 | … |  |
| Row 2 |  |  | … | 2 |
| Row 3 |  |  | … | 3 |
| ⁝ | ⁝ | ⁝ |  | ⁝ |
| Row m |  |  | … | m |

To implement a motor winding on the generic formulation described in table 2.4, a few more variables must be defined for a double-layer, 3-phase, integral slot winding pattern. Since this optimization problem is constrained to non-reduced motors, the rows on the right half of the table are shifted up by resultant value of equation (10).

|  |  |  |
| --- | --- | --- |
|  |  | () |

This can be seen in practice for a sample 3-phase, double-layer winding (=12, =4, =3) in table 2.5. The procedure is as follows:

1. m rows and columns are defined for the upper layer winding.
2. The first slot is assigned to the first cell of the table and the second slot is assigned cells away from the first. This process repeats, incrementing the slot number, until the table is full.
3. The upper layer columns are divided in half. The left half contains all positive (+) terminal directions, and the right half contains the negative (-) terminal directions.
4. The right half is then shifted downwards by rows defined in equation (10), highlighted in brown.
5. columns are then appended to the table for the lower layer terminals.
6. For each terminal slot number in the upper layer columns, the opposite terminal direction is assigned for the lower layer columns different by slots.

Table 2.5

WDT of Sample Motor

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Upper Layer | | | | Lower Layer | | | |
| Coil Direction | + | + | - | - | - | - | + | + |
| Phase A | 1 | 7 | 2 | 8 | (4) | (10) | (5) | (11) |
| Phase B | 3 | 9 | 4 | 10 | (6) | (12) | (7) | (1) |
| Phase C | 5 | 11 | 6 | 12 | (8) | (2) | (9) | (3) |

Rotary motor winding patterns rarely require empty slots due to the continuity of the motor core along the radial direction. This is not true for linear motors due to the linear core. When a linear motor is wound like a rotary motor, there is often a coil which has coil terminals on each end of the motor that produces an imbalance in the per-phase winding resistance which must be avoided. The WDT allows for empty slots to be included in the formulation of the winding pattern but is omitted in this paper to reduce the complexity required to produce winding patterns for an entire domain of motors.

## Hybrid Analytical Model Structure

Linear motors that have a flat primary core are naturally formulated in rectangular coordinate systems due to their rectangular-like shape. Before a motor model can be solved, a mesh of rectangular nodes is initialized for the motor geometry by discretizing the model and prioritizing the motor core geometry. Since the slot and coil geometries are generally the most complex, the mesh density in the x and y direction is proportional to the complexity of the core shape. The HHAM is an optimal application for this mesh complexity as it merges the benefits of both MEC and harmonic modelling. Within figure 2.3 the division of the model into unique regions through continuous and non-continuous boundaries is realized. For the MEC region, variable will be used for the node index in the x-direction while is the node index in the y-direction. The finite index limits for these two index vectors are defined as and , where their product results in the total number of node indexes .

Graphical user interface

Description automatically generatedFig. .. Meshed motor model containing boundary conditions.

The lengths of a node in the x and y direction are and respectively which defines the dimensions of the rectangle nodes throughout the mesh i.e., the value of in the MEC region is constant throughout all other regions along the -direction with a constant and vice versa. The left and right boundary coordinates in the x-direction are assigned to and respectively. To maintain periodicity in the x-direction, the nodes on the x-boundaries where = 1 and = L are coupled which is elaborated in table 2.6.

Table 2.6

Node Index Continuity

| ***current* node x index** | ***left* node x index** | ***right* node x index** |
| --- | --- | --- |
| = 1 |  |  |
| = |  |  |
| 1 < < |  |  |

Chart, box and whisker chart

Description automatically generated

Fig. 2.4. Single MEC node element at index .

Now that the size and density of the mesh has been defined, it is important to define the properties of each individual node within the mesh. Each node has a reluctance, flux, and MMF component which is defined by the material the node encloses. To avoid cluttering the image of figure 2.5, the index annotations (seen in figure 2.4) were omitted.

Chart, box and whisker chart

Description automatically generated

Fig. 2.5. MEC modelling of 4 arbitrary neighbouring nodes within 3 different arbitrary materials (yellow, blue, red).

The relative permeability (, the vacuum permeability (, and the cross-sectional area ( are all required to define the reluctance of a node:

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

Since the material in the node is homogenous, the reluctance near the positive boundary is equal to the reluctance near the negative boundary for x and y directions. The conservation of flux is maintained in equation (13) stating that all flux entering one potential node should be equal to the magnetic flux leaving the node.

|  |  |  |
| --- | --- | --- |
|  |  | () |

The flux contained in each node at a given time is written as:

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

Variables , , , and are the indices of the neighbouring nodes. The source terms producing the flux are the MMFs generated by the coils defined as contained in a node and defined in equation (18). The MMF in a coil is calculated based on the current excitation (, the number of turns in a coil (), the number of nodes in the x-direction for a single coil (, and the y-position of the node in the coil (.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Since the motor domain is constrained to 3-phase, equations (19) – (21) define the phase rotation for the windings.

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

The y-position of the node in a coil determines the contribution of the node’s MMF to length of the flux path through the airgap. Nodes that are positioned near the yoke of core will enclose the whole area of the coil and produce a longer flux path, leading to an increase in scaling factor, .

Chart

Description automatically generated

Fig. 2.6. MMF scaling factor per node in the y-direction of a coil for a) coil in upper slot and b) coil in lower slot of the primary core [20].

The scenario of coils in both lower and upper sections of the slot is accounted for using superposition of parts (a) and (b) of figure 2.6. The figure is limited to a coil modelled with 1 node in the y-direction and is determined at the center y-position of the node due to the discretization of the MEC region. When increasing the number of nodes in the y-direction, increases proportionally per additional node.

Due to the merger of MEC and HM, the unknown variable for the potential of the node arises in the flux equations above. This value is calculated in equation X:

|  |  |  |
| --- | --- | --- |
|  |  | () |

The unknown potential can be broken down into its time and space dependent parts. The time dependent complex exponential is defined by the frequency (), the time (, and the complex notation . Alternatively, the space dependent part is an unknown variable that requires a system of linear equations to solve which is discussed in detail within chapter 2.5. To quantify the HM regions, the equations for magnetic flux density and magnetic field strength materialize in the form of a complex Fourier series. The equation parameters change from in the MEC equations to since the HM does not require discretized points and is solvable for any coordinate in the model. Since the MEC model determines the mesh density, the HM model follows suite and will be calculated at the center of a node for a processed solution. To solve the magnetostatic field distribution in a region, the magnetic flux density, , can be written in terms of the magnetic vector potential, , via the diffusion in equation (23).

|  |  |  |
| --- | --- | --- |
|  |  | () |

The vector potential is defined as a complex Fourier series in the form of:

|  |  |  |
| --- | --- | --- |
|  |  | () |

To couple to MEC regions, the HM region has unknown variables and which are solved in a system of linear equations like the unknown MEC term . The relationship between the magnetic flux density and the vector potential is defined in equation (24) and allows for the solution of the tangential and normal component of the magnetic flux density equations:

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

Where:

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

These equations produce the solution of the magnetic flux density for one periodical length and N spatial harmonics, where one space harmonic is defined as . Since the HM region contains the same material throughout the region, the values , , and are independent of a node index within the region. The relative velocity between the primary and secondary is defined as . The unknown variables and are like of the node in the MEC.

## System of Linear Equations

Now that the required mesh parameters have been defined, the construction of the system of linear equations relating the unknown variables can begin. The boundary condition between two neighbouring regions can be between two HM regions, between MEC and HM regions, or it can be non-continuous. This classification defines which unknown variables are included in the equation. Since sources cannot be infinite in magnitude and the air surrounding the model theoretically extends to infinity, the Dirichlet boundary condition applies. The [21] forcing all the field components to vanish at the boundary. This equation applies to regions , and is defined as:

|  |  |  |
| --- | --- | --- |
|  |  | () |

For continuous boundaries, the normal and tangential components of each neighbouring region must be conserved. This is true for HM-HM boundaries as well as HM-MEC boundaries. Where is the lower region index at the boundary positioned at .

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

Where:

|  |  |  |
| --- | --- | --- |
|  |  | () |

The HM-MEC boundary must be expanded upon to couple the Fourier and MEC solutions. Unlike the MEC region, the HM regions do not produce a source. This means that the transfer of energy into the HM region is conserved at the HM-MEC boundary. Equation (31) can be implemented at the boundary using Equation (13) to produce the equations:

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

The flux in the normal direction in equations (34) is then substituted with equation (38) which explains that the magnetic flux is equal to the average flux density times the cross-sectional area at the boundary. The equation for the flux with the depth of the domain is defined as:

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

The solution for was solved using the same steps as in equations (36) – (38) by substituting with instead of . While the equations above define the normal boundary condition, the equations below define the tangential boundary condition. For this boundary, equation (32) can be expanded as:

|  |  |  |
| --- | --- | --- |
|  |  | () |

Both sides of the equation are in the form of a complex Fourier series as seen by the summation across harmonics. Discretizing the coils into nodes of a mesh creates a staircase shaped waveform which indicates that the Fourier series needs to be modified for a piece-wise continuous function value. This concept is shown in equation (40) which expands on the Complex Fourier Series.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Some of the variables that help solve for the function value depend on the position of the node at index (. The tangential magnetic flux density of a node is equal to the average flux in the x-direction divided by the cross-sectional area of the flux direction (x-direction):

|  |  |  |
| --- | --- | --- |
|  |  | () |

To produce a processed mesh model, the equations for each boundary condition are separated into a matrix of coefficients , a matrix of unknown variables , and a matrix of constants . Table x below expands on the matrix equation :

Table 2.7

System of Linear Equations Solving for Unknown Variables

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | = |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | . |  |
| . | . | . |
| . | . | . |
|  |  |  |

Where the dimensions of the square matrix are , where is the total number of HM regions in the model and is the number of MEC regions in the model. M is defined as the number of nodes in the MEC region and N is defined as the number of harmonics in the waveform approximation. The dimensions of the column vectors and are . To optimize the system of linear equations, the equations and coefficients that are solvable in the pre-processing stage can be removed. In the Dirichlet equations, an infinite position drives the unknown coefficients to and 0 for and respectively.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Alternatively,

|  |  |  |
| --- | --- | --- |
|  |  | () |

These equations can now be removed from the equation set along with the and unknown variables. The removal of 2N equations and 2N unknown variables maintains a square matrix A which has the new dimensions of . The system of linear equations is then solved using lower-upper-decomposition to produce the unknown variables of the HM and MEC regions.

Table 2.8

Boundary Condition Summary

|  |  |  |
| --- | --- | --- |
| HM-HM | HM-MEC | Non-continuous |
| (31), (32) | (40) | (42), (43) |

## Processed Model

Now that the unknown variables for the HM and MEC regions are solved, their values can be substituted into the model equations from section 2.4 to solve for any processed mesh parameters. An important performance parameter used in the GA objective function is the thrust of the motor. The force on the primary of the motor has a normal and tangential component which can be calculated with the equations below:

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

These equations were derived from the Maxwell stress tensor [22] in the airgap where the complex conjugate of a complex variable is denoted with a \* in the superscript. The conduction loss in the secondary of the motor can be calculated using the Poynting vector, applied in the air gap.

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

These post processing parameters are useful to create field plots on the mesh of the motor to highlight the performance at various operating conditions. For example, saturation effect can be determined by plotting the B field on the motor core and determining if the magnetic flux density has exceeded the capability of the motor core material.

The equations described in this chapter depend on the number of harmonics considered in the Complex Fourier Series approximation MEC discrete waveform at the boundary conditions.

Diagram

Description automatically generated

Fig. 2.7. Complex Fourier Series approximation of the discrete B field at the center of the airgap for various harmonics.

To highlight this effect, figure 2.5 shows the discrete plot of the B field in the air gap in black. Two Complex Fourier Series approximations are overlayed on the base plot to show the improvement in approximation accuracy when more harmonics are considered. Visually the orange plot that contains 70 harmonics in the series can capture the peaks of the waveform much better than that of the blue plot that contains only 20 harmonics. Although an increase in harmonics correlates with approximation accuracy, it dramatically increases the computation complexity and should therefore appropriately be chosen.

# Optimization Algorithm

Within the scope of evolutionary algorithms, GA and PSO are the dominant algorithms when the problem demands robustness and performance. With the overarching objective of integrating the optimization algorithm with the HAM, the comparison between PSO and GA must be carefully considered to ensure that the chosen solver can meet the unique demand of having the HAM as its objective function. In this section the core functionality of each algorithm will be discussed and then compared against one another in a case study to statistically determine the optimal solver for the problem.

## Genetic Algorithm

The GA is a kind of evolutionary algorithm that mimics the general concept of evolution. Natural selection is often mentioned in the context of evolution since it is the strong individuals that survive in each environment. Being the strongest is a generalization that is defined by the objective function applied to the optimization problem. The structure of a population subject to the GA is visualized in Figure 3.1 encapsulating a fixed number of chromosomes, which themselves encapsulate genes.

A screenshot of a video game

Description automatically generated with medium confidence

Fig. 3.1. Layout of a genetic algorithm with an arbitrary number of chromosomes and genes per population.

To understand the function of a gene, the application of the algorithm must be defined since the genes are merely input variables to the model that requires solving. If the optimization problem were a 2-dimensional surface plot minimization, the inputs to the model would be an arbitrary 2-dimension coordinate. Each dimension of this coordinate is considered a gene using the nomenclature of the GA.

Diagram

Description automatically generated

Fig. 3.2. Layout of a genetic algorithm execution loop.

Throughout each iteration of the solver a new population is produced via selection, crossover, and mutation. This iterative loop ensures that the algorithm favors the desirable solutions while maintaining robustness through some degree of randomized search throughout the optimization domain.

## Particle Swarm Optimization

Like GA, the PSO mimics the natural phenomenon of the power of a collective. This is often seen in swarms of insects such as bees which constantly communicate with one another to determine the optimal direction of the entire swarm. If the swarm’s objective were to find a new location to establish a hive, each bee plays a critical role to gather information and relay it throughout the swarm so that the collective can weigh the signals and converge on decisions in real time. Instead of the population, chromosomes, genes, and offspring nomenclature, the PSO uses swarm size, particles, and leaders.

Diagram

Description automatically generated

Fig. 3.3. Layout of a particle swarm optimization algorithm optimization loop.

The optimization loop of the PSO shows the process of updating velocities and positions per particle in the swarm as elaborated in equations (52) and (53).

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

The current and successive iterations are denoted as and respectively, where the local and global best solutions are determined prior to updating positions and velocities . The inertial weight coefficient, local weight coefficients and , and global weight coefficients and are integral in determining the relative influence the swarm has on the particle and vice versa.

Table 3.1

PSO Velocity and Position Coefficients

|  |  |
| --- | --- |
| **Constant** | **Range** |
| **R** |  |
| **C** |  |
| **W** |  |

Referring to the optimization loop, the final step before calculating the objective function on the updated particles is to subject each particle to a mutation algorithm with a designated probability that the mutation executes. This varies the swarm and increases the robustness of the solver to avoid convergence on local minima and maxima.

## Schwefel Function Minimization Case Study

A case study was conducted to determine the optimal optimization algorithm among the subset of EAs through the Schwefel test function. A test function is used to test the ability of an optimization algorithm to converge on a solution that is the global maximum or minimum rather than the function’s local maxima or minima. The Schwefel function was chosen since it has a plethora of local maxima and minima which can stall solvers prior to converging on the solution. The function is defined as:

|  |  |  |
| --- | --- | --- |
|  |  | () |

where is the number of input dimensions and is the function input per dimension . The global minimum is located at inside of the hypercube for all

Chart, surface chart

Description automatically generated

Fig. 3.4. Surface plot of the Schwefel function on the input range.

Background pattern

Description automatically generated

Fig. 3.5. Contour plot of the Schwefel function on the input range highlighting the global minimum with a red cross.

To couple a solver to this test function, a new input is generated by the solver per iteration. These inputs are used to calculate and minimize the objective value through the Schwefel function until convergence on a solution. To ensure that each optimization algorithm is fairly compared in this case study, common solver parameters are used to configure each algorithm which can be found in Table 3.2. Every algorithm will iterate over its population or swarm with the only solver termination criteria being the max number of stall iterations reached. Other solver termination criteria like reaching objective tolerance, timeout, and maximum iterations were omitted in this case study to isolate each solver through a consistent test domain. Additionally, the optimization process is conducted 5 times per algorithm to determine the average performance to ensure that an outlier does not significantly impact the decision making. Table 3.3 compares the EAs: PSO and GA through performance parameters like execution time and error. The solver robustness is the principal performance parameter, while the solver time holds less value as a performance parameter.

Table 3.2

Optimization Algorithm Configuration

|  |  |  |  |
| --- | --- | --- | --- |
| **PSO** | | **GA** | |
| **Population/Swarm Size** | 200 | **Population Size** | 200 |
| **Max Leader Size** | 100 | **Offspring Size** | 100 |
| **Comparator Key** | Objective Value | **Crossover Percentage** | 30% |
| **Mutation Percentage** | 10% | **Mutation Percentage** | 10% |
| **Algorithm Stall Iterations** | 25 | **Algorithm Stall Iterations** | 25 |
| **Global Upper Bound** | [500, 500] | **Global Upper Bound** | [500, 500] |
| **Global Lower Bound** | [-500, -500] | **Global Lower Bound** | [-500, -500] |

Table 3.3

Average Optimization Algorithm Results

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **PSO** | **GA** |
| **Time (s)** | 1.5246 | 1.4546 |
| **Objective Function Executions** | 10760 | 4171 |
| **Solver Iterations** | 57 | 116 |
| **Value of X1 Solution** | 420.9728 | 420.9522 |
| **Error of X1 Solution (%)** | 0.5053 | 1.6543 |
| **Value of X2 Solution** | 420.9669 | 420.9729 |
| **Error of**  **X2 Solution (%)** | 0.1810 | 0.4172 |
| **Value of Final Objective** | 0.0001 | 0.0002 |
| **Error of Final Objective (%)** | 0.0052 | 0.0195 |

From the data found in Table 3.3 it is evident that both the GA and PSO converge on the global minimum across 5 trials. The error in the final coordinate and the error in the resulting objective value at the coordinate were considerably low, although GA was not able to search the peaks as well as PSO. The characteristics of the algorithm’s ability to search the space can be visualized by plotting the swarm or population for a given solver iteration. A comparison between GA and PSO searching the space on the contour plot shown in Figure 3.5 is achieved by selecting the early iterations of each solver. This comparison is found in Table 3.4 which highlights the meta differences between GA and PSO. The GA tends to cluster in the minima that it finds after the first iteration and spawn offspring that allows it to search those minima further. This continues until the population produces enough generations at better minima, reducing the number of offspring centered around the local minima. Contrasting this with PSO, the swarm finds the global optimal solution after the first iteration and begins to orient the velocities towards the swarm’s global minimum (different from the domain’s global minimum). When particles find other local minima, they will slightly affect the swarm’s orientation unless it is the swarm’s new global minimum, in this case the swarm begins to reorient towards this point with much greater influence.

Table 3.4

Algorithm Convergence Visualization

|  |  |  |
| --- | --- | --- |
| **Solver** | **Best Objective** | **Iteration 1** |
| **GA** | Value: 166.0198  Coordinate: (-284.4012, 428.5994) | Diagram, background pattern  Description automatically generated |
| **PSO** | Value: 128.4705  Coordinate: (-296.7353, 427.7533) |  |
| **Solver** | **Best Objective** | **Iteration 3** |
| **GA** | Value: 102.4799  Coordinate: (395.6167, 406.8422) | Background pattern  Description automatically generated |
| **PSO** | Value: 119.4965  Coordinate: (-303.3086, 423.7552) |  |
| **Solver** | **Best Objective** | **Iteration 5** |
| **GA** | Value: 102.4799  Coordinate: (395.6167, 406.8422) |  |
| **PSO** | Value: 119.1884  Coordinate: (-303.1026, 423.3363) |  |

The time of termination, after 25 stall iterations were reached, for each algorithm was approximately the same due to the lack of computation intensity this optimization problem requires. However, the number of solver iterations i.e., the number of new swarms or populations produced, were much greater in the GA although this is not a concern. The method in which the swarms and populations are produced are time efficient and only significantly hinder computation time when the swarm or population are significant in size. When comparing the objective function executions required to converge on a solution, this is where there is a clear difference between the GA and PSO. The number of function executions is more than double that of the GA which is not intuitively a problem. The visualization of the problem is introduced in Figure 3.6 showing the divergence of the GA and PSO solver times when the objective function execution time increases.

Chart, line chart

Description automatically generated

Fig. 3.6. Comparison of the average solver execution time between GA and PSO until 25 stall iterations are achieved using the Schwefel test function at different artificial objective function execution times.

The Schwefel function was artificially slowed down from the original 0.0086ms to 100ms in steps shown in the plot. Solving the Schwefel test function at each of these steps and logging the time it takes each algorithm to converge proves that the GA is much more efficient for slower objective functions. This is a very important decision variable when choosing between GA and PSO since the HAM will need to be solved multiple times per iteration of the motor optimization problem. If the data in the plot were extrapolated to seconds or even minutes in duration, then the difference in solver execution time between GA and PSO would be much more apparent. In summary, the GA is chosen as the optimal optimization algorithm for the Schwefel test function which will act as the foundation for the solver in the motor optimization problem.

## NSGAII Configuration

Without modification, the GA cannot optimize multi-objective problems and requires a modified implementation that produces non-dominated solutions. The non-dominated sorting genetic algorithm II [NSGAII] is a modified implementation of the GA which will be implemented for the motor optimization problem. There are many core functionalities that are required for the NSGAII to successfully navigate a problem’s constrained space and optimize towards a solution. This is no simple task and a misconfiguration of just one core function can result in an instable solver. The classification of NSGAII’s core functionality can be segregated into selection of dominant parents and variation for searching the domain in a robust manor. These functionalities will be discussed in detail within the following sub-sections.

### Solver Selection

Selection is a core solver function that identifies the strongest parents among the population through comparison of performance. This identification process is achieved with a fitness function, which is application specific, coupled via a maximization or minimization definition. Since the population size must remain constant, the weakest parents are removed from the current population and discarded. The remaining parents are then subject to variation which will be discussed in the next section. There are many robust selection algorithms that will find the highest performing parents such as Roulette Wheel and Rank selection [23], although in this paper the focus will be on Tournament selection. The basic principle is that a sample of parents are selected to compete against one another in a tournament-style comparison of their objective values. The likelihood of a parent being selected is dependent on the selection pressure which is a probabilistic measure of a candidate’s likelihood of participation in a tournament. This parameter is an indicator of a solvers ability to converge since higher selection pressure relates to a higher convergence rate.

A picture containing text, electronics

Description automatically generated

Fig. 3.7. Layout of a Tournament selection algorithm using arbitrary objective values to highlight the winning decision based on a minimization problem.

During the case study, optimal parameter ranges for Tournament selection were determined and will be used in the HAM optimization problem. Since tuning the selection configuration is specific to the optimization problem, these parameters will have to be slightly modified, serving as a benchmark for optimization effectiveness.

### Solver Variation

The NSGAII has core functions that are appropriately named after events in the natural process of evolution. Crossover is one of these functions. It allows parents to exchange their qualities and produce children while the remaining qualities are subject to some form of randomized initialization. The number of variables that are subject to be overwritten is defined by a crossover point as visualized in Figure 3.8. Note that the values of the variables were limited to binary for simplicity, but the true values can contain other formats such as integers and real numbers. The crossover point determines the percentage of variables shared among parents, it is important to not choose too small or large of a ratio due to solver robustness. If a small percentage of variables from the parents were crossed over then the solver may become stuck in local minima or maxima rather than the desired global alternative. Alternatively, a large percentage of variables crossed over between parents will have large variations in the solution and can cause an instability in the solver.

Graphical user interface

Description automatically generated with low confidence

Fig. 3.8. Visualization of crossover between two parent variables to produce two child variables governed by the crossover point.

The frequency that the crossover is applied is also an important configuration consideration. This is defined as the probability that crossover will occur between parents and is integral in the solver’s robustness. Like the crossover point, if the probability of crossover is set too high then the parents will often share variables when producing children which is susceptible to finding local minima or maxima rather than the desired global alternative. Contrasting this with a low probability of crossover between parents, the solver may become unstable. This is due to the children’s variable initialization relying on randomized initialization which will resist solver convergence.

Mutation is another important function of EAs which is responsible for manipulating the values of randomly selected variables within a parent. The probability for mutating a parent’s variables shall remain low to maintain solver robustness rather than introducing instability. The general concept of mutation is visualized in Figure 3.9, which highlights the variables that were randomly selected for mutation within the parent.

A screenshot of a cell phone

Description automatically generated with medium confidence

Fig. 3.9. Layout of a genetic algorithm with an arbitrary number of chromosomes and genes per population.

Note that the values of the variables were limited to binary for simplicity, but the true values can contain any format such as integers and real numbers. Similar to the selection section, tuning the variation configuration is specific to the optimization problem, these parameters will have to be slightly modified, serving as a benchmark for optimization effectiveness.

# Model Optimization Integration

Due to the size and complexity required to build a HAM it is important to simplify the model into smaller procedures. Figure 4.1 highlights the state transitions made by the model to produce a pre-processed motor, solve the system of linear equations, and produce a processed motor model. The motor’s performance parameters are then used to compute the GA objective function value and compare it to a desired solver tolerance.

Diagram

Description automatically generated

Fig. 4.1. Motor optimization algorithm state chart for hybrid analytical modelling.

The *Build Motor* and *Compute HAM* states were defined in chapter 2. The *Variation*, *Selection*, *Solver Termination*, and *Compute Fitness* state structures were discussed in chapter 3 using the Schwefel test function for visualization. In this chapter, the integration of these states with HAM will be discussed in more detail.

## Optimization Constants

The baseline motor parameters were detailed in chapter 2. The tables in this section detail the motor parameters which were kept constant during the optimization process. The linear induction motor secondary was kept the same as the baseline since the focus of the optimization was on the primary and the parameters are mutually exclusive.

Table 4.1

Constant Motor Parameters

|  |  |  |
| --- | --- | --- |
| **Description** | **Variable** | **Value** |
| Yoke height (mm) |  | 6.5 |
| Airgap (mm) |  | 2.7 |
| Aluminum thickness (mm) |  | 2 |
| Back iron thickness (mm) |  | 8 |
| Primary length (mm) |  | 270 |
| Primary depth (mm) |  | 50 |
| Periodical length of model (mm) |  | 525 |
| Number of phases |  | 3 |
| Synchronous velocity (m/s) |  | 0 |
| Electrical frequency (Hz) |  | 100 |
| Peak current (A) |  | 10 |
| Aluminum Conductivity |  |  |
| Iron Conductivity |  |  |
| Relative Permeability of Iron |  | 1000 |

Table 4.2

Model Parameters

|  |  |  |
| --- | --- | --- |
| **Description** | **Variable** | **Value** |
| Number of space harmonics |  | 100 |
| Number of HM regions in the model |  | 5 |
| Number of MEC regions in the model |  | 1 |

## Motor Feasibility

To ensure that all motors produced by the optimization algorithm are feasible, the model abides by the rules in table 4.3. If every solver iteration produces a feasible design, then there is no wasted computation. Alternatively, even if a small percentage of the produced motors are infeasible the resulting computation intensity can be costly in a complex optimization problem such as the one in this paper.

Table 4.3

Motor Feasibility Constraints

|  |  |
| --- | --- |
| **Rule** | **Explanation** |
|  | Monopoles cannot exist |
|  | Rotating/moving fields require minimum 2 pole pairs. |
| Ns % m = 0  &  q % 1 = 0 | All motor slots must be filled with a coil and only integral slot windings are considered. |
| B < 1.7T | The primary core material saturates at 1.7T which must be avoided by limiting the current density, of a coil terminal. |
|  | The limited current density avoids saturation. |
|  | Motor core teeth produced past this threshold are considered mechanically fragile. |
|  | The required frequency at high-speed operation cannot produce a skin depth, deeper than the thickness of the aluminum. |

It is important to consider the skin effect if the motor application demands high speeds since the mechanical speed is directly proportional to the primary electrical frequency from equation (55). The skin depth is calculated assuming low frequencies [24] as:

|  |  |  |
| --- | --- | --- |
|  |  | () |

Where and are the resistivity and permeability of the aluminum plate respectively. When plotting the skin depth across frequencies, the feasible frequency range is realized in figure 4.2.

Shape

Description automatically generated

Fig. 4.2. Skin depth (blue) in the secondary aluminum plate, with increased resistivity to account for the transverse end-effects, including the plate thickness (orange).

The plate thickness and the skin depth curve do not intersect within the 1kHz electrical frequency range meaning that the secondary back iron is always coupled with the primary. This is an important check for motors used by the optimization algorithm otherwise their operating frequency may be constrained by the secondary design. If the dimensions or material properties of the aluminum sheet were different from the values considered in this paper, this constraint may be applicable.

From the maximum frequency of operation in figure 4.2, the theoretical top speed of the motor can be solved using equation (56).

|  |  |  |
| --- | --- | --- |
|  |  | () |

This is a theoretical top speed due to the required input voltage to overcome the equivalent impedance and resistance of the motor to continue driving enough current to produce sufficient force to overcome mechanical losses to accelerate towards this speed. The frequency of operation and the slip are contributing factors to the equivalent impedance and resistance which can help predict the required voltage at rated operating conditions. Once the supply voltage to produce this top speed is deemed feasible, the true top speed of the motor is determined.

## Compute Fitness

After solving the *Compute HAM* state in figure 4.1, the performance parameters can be gathered and then maximized, minimized, or trended towards a bias. The performance parameters chosen for this optimization problem are defined in figure 4.3 which are outputs of each motor produced by the HAM. By optimizing for thrust and mass the optimal motors will have a larger thrust-weight ratio. Theoretically, more performance parameters can be added to the multi objective optimization although adding more objectives results in less non-dominated solutions being produced. This results in more HAM executions without producing an improved motor which is undesirable especially when computation considerations are a key focus of this optimization problem.

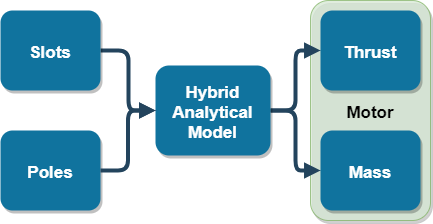


Fig. 4.3. Layout of the motor optimization algorithm inputs and the resultant multi-objectives.

Since all objectives are relatively important performance parameters it is important that the solver produces pareto-optimal solutions, meaning the solution equally satisfies the fitness function criteria. A solution that is not pareto-optimal will still optimize every multi-objective variable but with an inequal emphasis.

The thrust calculation of the motor is described in equation (45) and is a product of the HAM equations. To calculate the mass of the motor, only the primary is considered since the secondary is a fixed design constraint. The primary mass is a summation of the core, winding, and insulation masses defined by their respective volumes times material density which are found in table 4.4 for each material.

Table 4.4

Mass Equations

|  |  |  |  |
| --- | --- | --- | --- |
| **Region** | **Material** | **Volume Equation** | **Density ()** |
| Core | Iron |  | 7.8 |
| Winding | Copper |  | 8.96 |
| Insulation | Plastic |  | 1.4 |

Section 4.2 discusses the criteria for feasible motor designs which must be reflected in the fitness function. If a motor is considered not feasible then the solver discards this iteration to save computation effort and intrinsically avoid future motor combinations like it. This is done by setting a constraint condition within the NSGAII problem definition to only allow solutions returning “True” to the feasibility in question.

## Baseline Validation

### Air Gap B field and Force Validation

Prior to executing an optimization workflow, the HAM model proposed in this paper must be validated so that future resultant motors can be trusted throughout the optimization loop. The definition of the baseline motor was discussed in section 2.1 which will serve as ground truth data since it includes FEM, experimental, and HAM data validation. The HAM model produced in this paper is a reproduction of the one within the reference paper, so it is natural to use its data for validation.

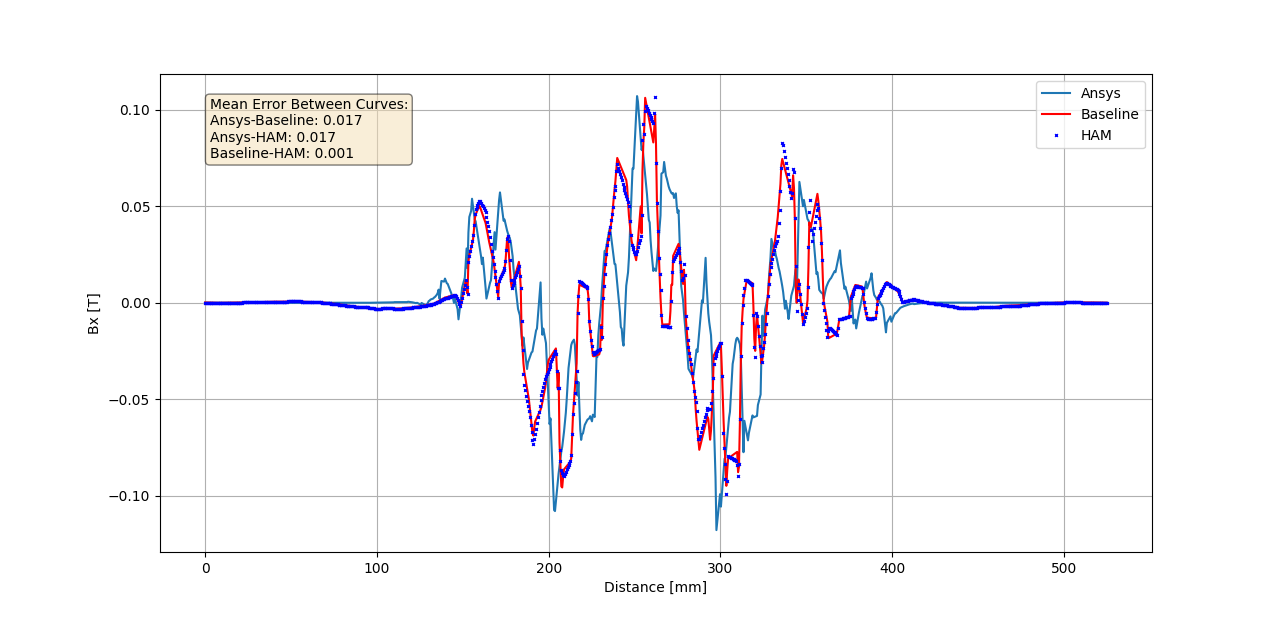


Fig. 4.4. Tangential B field in the center of the air gap comparison between Ansys Electronics FEA (light blue), reference paper results (red), and the HAM produced in this paper (dark blue).

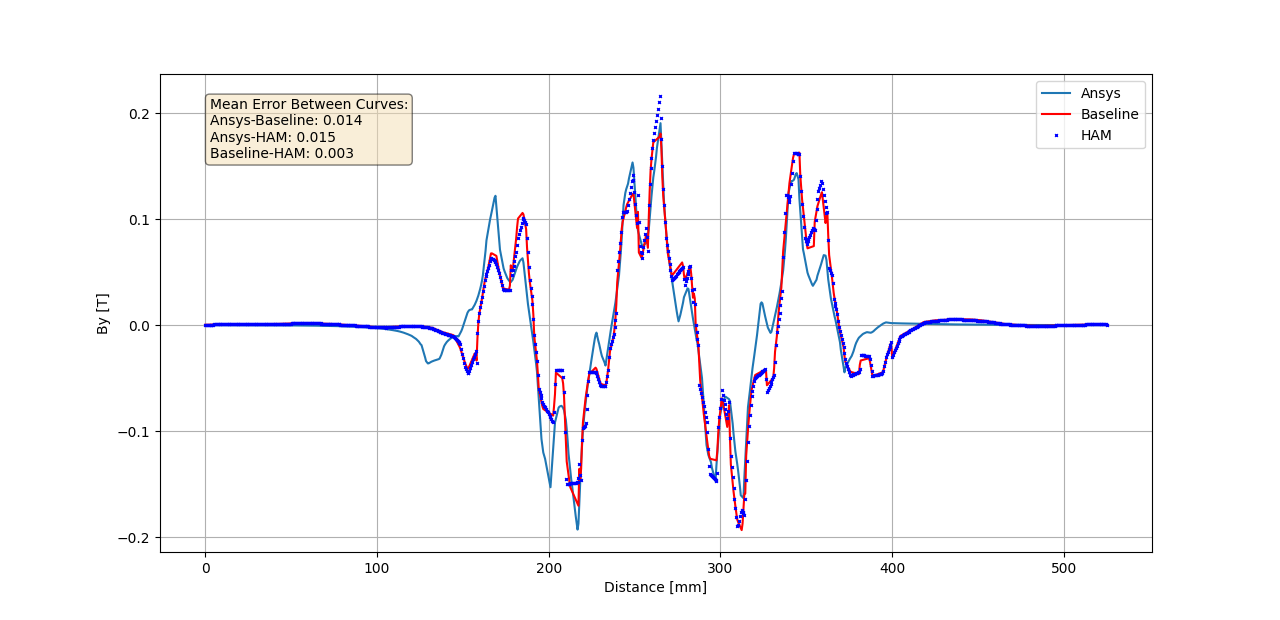


Fig. 4.5. Normal B field in the center of the air gap comparison between Ansys Electronics FEA (light blue), reference paper results (red), and the HAM produced in this paper (dark blue).

Within the upper-left text box in figures 4.4 and 4.5 the mean error between curves is highlighted. This mean error between two curves is calculated as:

|  |  |  |
| --- | --- | --- |
|  |  | () |

Where and are the two curves which are dependent on the independent coordinate x. The mean error for both the normal and tangential B field plots of the air gap is low which confirms the validity of the HAM produced in this paper. The small misalignment and error between the Ansys Electronics results comes from the difficulty in choosing the correct steady state time step to match the reference paper since this was not provided. The air gap plots were simulated at 60ms in Ansys Electronics which was enough time for the simulation to reach steady state and had the best correlation to the baseline results. This time step could be slightly modified to align with the baseline results better but the mean error between the Ansys Electronics, baseline, and HAM curves is low and can be considered accurate.

### Motor Core B Field Plot Validation

In section 4.2, table 4.3 addressed the saturation constraint which must exist for all motor solutions to avoid degrading the motor core material and performance. The HAM is powerful enough to produce field plots for any post-processing model parameters such as B field. The field plot needs to be filtered to only contain B field data within the geometry of the motor core. A comparison between the field plotting results of the HAM proposed in this paper and Ansys Electronics motor simulation software are visualized in figures 4.6 and 4.7 respectively. The filtering functionality in the HAM model is in effect on the right side of the image which has the magnitude of the B field per node mapped to the color bar on the left. The transparent, light blue overlay is helpful in visualizing the region where the filter is applied. Alternatively, on the left side of the figure, the original model is visible without the filter applied to highlight the flexibility of the filtering implementation.

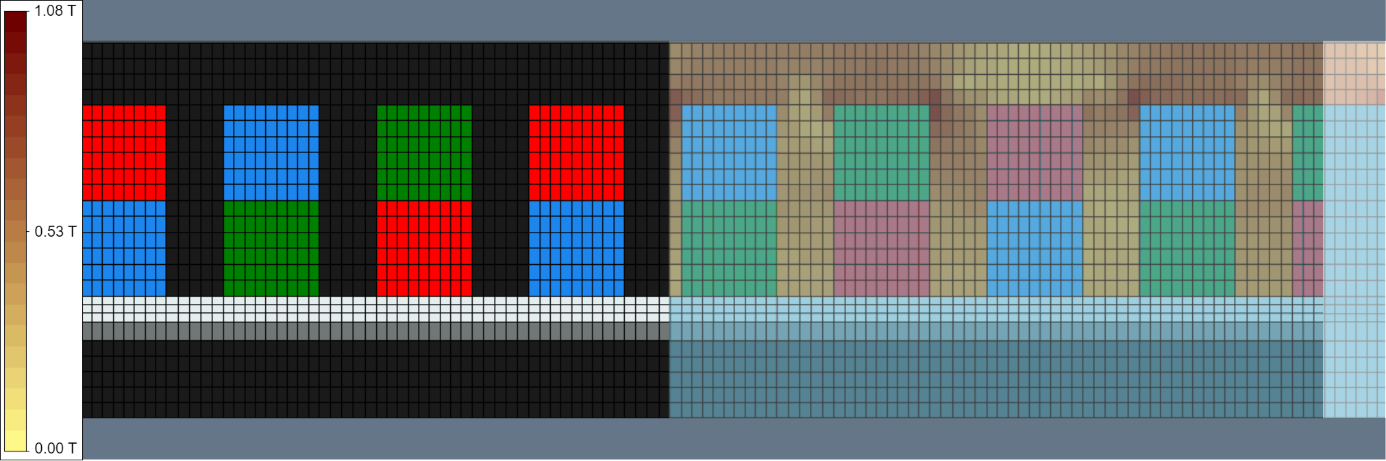


Fig. 4.6. Magnetic flux density (B) field plot in the motor core of the HAM simulation for the baseline motor to validate accuracy.

Chart

Description automatically generated with low confidence

Fig. 4.7. Magnetic flux density (B) field plot in the motor core of the Ansys Electronics simulation for the baseline motor to validate accuracy.

Visually the results between Ansys and HAM are quite similar but vary slightly due to the difficulty in choosing the perfect time slice for the plotting to match without error. More importantly, the bounds of the color bar for B field within the core of the motor for both plots have a percent error of 1.3% which is due to these factors:

* The meshing density of HAM is less than that of ANSYS in this example (polygons vs rectangle mesh).
* The time slice chosen for the simulation between a) and b) is not the same.

Given that the error is relatively low, this proves the accuracy of the HAM modelling algorithm and provides a robust method for creating feasible motors that do not saturate.

## Solver Configuration

Table 4.5

Optimization Configuration

|  |  |
| --- | --- |
| **Configuration** | **Value** |
| Average mesh elements | 5,002,020 |
| Termination condition | stall |
| Iterations |  |
| Optimization time |  |
| Motors modelled | 9 |
| Ns (min, max) | (12, 54) |
| Np (min, max) | (4, 12) |

The total coil terminal area for MMF to be produced in a motor with 12 slots versus a motor with 36 slots only has a 2.5% increase but the thrust is much higher as well.

Chart, line chart

Description automatically generated

Fig. 4.8. Transient tangential (Fx) and normal (Fy) force plot of all motors evaluated during the optimization process calculated as a steady state average between the time interval of 48 to 60 ms.

Only 9 motors were considered in the optimization process where slots.

A picture containing table

Description automatically generated

Fig. 4.9. Average steady state thrust plot with data point annotations in the format: (Ns, Np) grouped by the x-axis into columns of q values.

Chart, scatter chart

Description automatically generated

Fig. 4.10. Pareto plot of the objectives: motor mass (x-axis) and average steady state thrust (y-axis), with data point annotations in the format: (Ns, Np) note the legend colours.

Pareto plots, like the one in figure 4.10, are ideal for analyzing a multi-objective optimization algorithm’s resulting solution. In most cases the algorithm will choose the optimal solution for the application unless the application does not equally consider the objectives. For example, the two best-performing solutions presented by the algorithm are (Ns=36, Np=4) and (Ns=54, Np=6) which are compared in table 4.6.

Table 4.6

Pareto Front Analysis

|  |  |  |
| --- | --- | --- |
| **Motor (Ns, Np)** | **Mass [Kg], Fx [N]** | **Δ Objectives Motor 1 to 2** |
| 1: (36, 4) | 1.80, 44.58 | Mass: (+8%)  Fx: (+12%) |
| 2: (54, 6) | 1.66, 39.69 |

When the objectives are considered without bias, the best motor is (Ns=36, Np=4). However, if the application held a bias towards lower mass, then the objectives between both motors are close enough to argue that (Ns=54, Np=6) is the optimal motor.

## Optimal Motor

Like the tables in Chapter 2, table 4.7 highlights the final values determined for the parameters that were variable throughout the optimization process.

Table 4.7

Optimal Motor Parameters

|  |  |  |
| --- | --- | --- |
| Pole pitch (mm) |  |  |
| Slot height (mm) |  |  |
| Tooth width (mm) |  |  |
| End Tooth width (mm) |  |  |
| Slot width (mm) |  |  |
| Slot pitch (mm) |  |  |
| Number of slots |  |  |
| Number of poles |  |  |
| Number of turns per coil |  |  |
| Coil pitch |  |  |

Table 4.8

Optimal Model Parameters

|  |  |  |
| --- | --- | --- |
| Number of rows in a magnetic equivalent circuit region |  |  |
| Number of columns in a magnetic equivalent circuit region |  |  |

# Research Summary

The implementation of the procedures in previous chapters creates the foundation for producing a novel optimization algorithm within the subclass of double-layer, single-sided, 3-phase, integral slot winding, linear induction motors. This is a very constrained optimization space required to address the general optimization problem of the slot-pole combination trends in motor performance parameters. A key theme throughout this paper is flexibility which is evident within each chapter. The WDT provides flexibility for extremely complex winding patterns. The HAM allows for complex motor models to be meshed and processed using MEC and HM modelling techniques to maintain computation efficiency. The NSGAII multi-objective optimization algorithm allows for custom objective functions that can be implemented to optimize, virtually any aspect of the HAM. This is extremely valuable considering that modern motor modelling software lacks the capability to perform general optimizations such as slot-pole combinations without some serious back-end coding implementation. To support such claims, it is important that the results produced within this paper have fundamental value and be thoroughly validated.

## Future Research on HAM and LIM Optimization

All software projects originate from an idea to solve a problem, regardless of their scale, and will endure many revisions prior to completion. Each revision serves to build on the previous layer until a final product is produced. It is often the case that the original vision was too vague to consider the intricacies of the final product, introducing inefficiencies in the project structure due to the time sensitivity of the target deliverables. This sub-section will investigate these considerations and highlight the potential for future revisions on their implementations.

### Constrained Optimization Domain

Although this thesis investigates a multi-objective optimization, there are many necessary constraints implemented which reduce the complexity and narrows the focus of motor optimization within its domain. These constants are highlighted in section 4.1 and partially limit the ability to generally search for an optimal motor within a relatively unconstrained domain. To truly find the optimal motor in all aspects of the application many layers of complex optimization are required. A wide range of operating conditions need to be considered. For example, a motor may have an operational frequency range from 50 Hz to 500 Hz which requires simulations at steps within this range. This paper optimizes at a constant frequency and the optimal motor may differ at different operational frequencies. AI or a layered approach to the optimization problem (keeping the top percentile of motors from each optimization layer until one remains) would solve this problem while drastically increasing the complexity.

* Ex, slip=1, freq=100, … if you really wanted to find the best motor within a subclass of motors, you would need to do this optimization for variations of these variables but it is possible since I made the tool possible

### Winding feasibility using WDT

The feasibility of linear motor winding patterns was discussed in section 2.3 which highlights the limitation of producing balanced windings. Due to the time and effort, it would take to implement this feature, it was omitted to reduce the complexity required to produce winding patterns for an entire domain of motors. Future work should improve upon this downfall and utilize the capability of the WDT to include empty slots, improving the effectiveness of the optimization. The current WDT implementation still produces accurate trends in motor performance and the final motor design would require a slight adjustment in the number of empty or partially empty slots to achieve a feasible motor.

### ANSYS Implementations

Using a custom modelling algorithm (HAM) added necessary flexibility to the optimization within this paper. However, using a custom model also introduces error that would be minimized if the optimization was coupled directly with ANSYS motor model results. This is theoretically possible using PyAEDT to interact with the AEDT API and pipe those results into a custom optimization algorithm all within a single python application. Further investigation on the feasibility and effort to implement this solution would be required but serves as a viable alternative to the implementation conducted in this paper.

## Conclusion

The baseline motor “had a discrepancy within 1.7% compared to FEA and were verified by static measurements” [20]. The B field plot in the center of the airgap had a maximum mean error of 0.003 between the reference paper and the HAM implementation in this paper. When comparing ANSYS and HAM, the maximum mean error was 0.017. The observed percent error of the magnetic flux density in the motor core between ANSYS and HAM was 1.3%. The source for these errors were discussed in section 4.4 and the observed errors are within the acceptable tolerance, proving the effectiveness of the motor modelling algorithm.

TALK ABOUT THE OPTIMAL MOTOR CHOSEN

Modern optimization applications provide convenience via effective solutions while reducing the time complexity for the end user. Ironically, this is not the case when training and validating such an application. At the completion of optimizing within the problem’s domain, 9 feasible motors were modeled and iteratively compared via multiple objectives. The optimization completed in 5 minutes whereas the average time for configuring and simulating a motor in ANSYS takes 30 minutes. This is relatively fast since a template ANSYS model reduced the configuration time significantly.

The true potential of the HAM optimization application is realized once onboarding an optimization task to a new lab member or employee who might lack the knowledge of optimizing within a target motor domain. This would prompt them to begin building their own template from scratch using their motor modelling software of choice. Evidently, a task such as this is extremely time intensive and is exaggerated when multiple people are onboarded with this task. By producing a configurable motor optimization tool, the end user only needs to be onboarded on the configuration of the tool rather than all the other aspects of the motor optimization process.

# REFERENCES

[1] “Total greenhouse gas emissions.” https://ourworldindata.org/grapher/total-ghg-emissions?tab=chart&country=~CAN (accessed Apr. 20, 2022).

[2] <https://environment.ec.europa.eu/research-and-innovation/science-environment-policy_en>

[3] <https://www.iea.org/topics/transport>

[4] https://www.nrcan.gc.ca/energy-efficiency/transportation-alternative-fuels/personal-vehicles/choosing-right-vehicle/buying-electric-vehicle/21034

[5] https://en.wikipedia.org/wiki/ICE\_3

[6] https://swissloop.ch/prototypes/claude-nicollier/

[11] <https://aedt.docs.pyansys.com/release/0.6/>

[10] T. M. Masuku, R. -J. Wang, M. C. Botha and S. Gerber, "Design Strategy of Traction Induction Motors," 2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA), 2019, pp. 316-321, doi: 10.1109/RoboMech.2019.8704761.

[Ansys-3] Module 03: Mesh Operations for Rotating Machines

[Trains-1] Pioneering the Application of High Speed Rail Express Trainsets in the United States, Parsons Brinckerhoff 2010 William Barclay Parsons Fellowship, Monograph 26.

[Friends-1] Tim Stachl

[Friends-2] Areej Fatima

[7] J. Pyrhönen, T. Jokinen and V. Hrabovcová, Design of Rotating Electrical Machines, John Wiley & Sons, 2008.

[8] S. Yamamura, Theory of Linear Induction Motors, 1st ed., Wiley, 1972.

[24] W. Hayt and J. Buck, Engineering Electromagnetics, New York: McGraw-Hill, 2012.

[Py-PPT] Finite Elements coding with python: Electromagnetics, Guillaume Verez

[Py-1] A Scaled Mesh/Nodal Formulation of Magnetic Equivalent Circuits with Motion Daniel C. Horvath, Student Member, IEEE, Steven D. Pekarek, Fellow, IEEE and Scott D. Sudhoff, Fellow, IEEE

[Py-2] Using Finite Element Method for Determination of Poles Number in Optimal Design of Linear Motor, Abdolamir Nekoubin

[Py-3] Comparison of DifferentWinding Configurations for Dual Three-Phase Interior PM Machines in Electric Vehicles Shensheng Wang 1, Ziqiang Zhu 1, Adam Pride, Juntao Shi, Rajesh Deodhar, and Chiaki Umemura

[9] A General Mathematical Formulation for Winding Layout Arrangement of Electrical Machines Massimo Caruso, Antonino Oscar Di Tommaso, Fabrizio Marignetti, Rosario Miceli, and Giuseppe Ricco Galluzzo

[Py-5] Timperio, Christopher, Linear Induction Motor (LIM) for Hyperloop Pod Prototypes

[Py-6] Jasmin Smajic, Simulation Based Development of the Hyperloop Innovation Award Winning Linear Motor

[Py-7] Design of a single sided linear induction motor (slim) using a user interactive computer program, Sarveswara Prasad Bhamidi

[Py-8] Equivalent Circuit Model of Linear Induction Motor with Parameters Depending on Secondary Speed for Urban Transportation System Yuichiro Nozaki, Terufumi Yamaguchi and Takafumi Koseki

[Py-9] Force Calculation in 3-D Magnetic Equivalent Circuit Networks With a Maxwell Stress Tensor Marco Amrhein, Member, IEEE, and Philip T. Krein, Fellow, IEEE

[Py-10] The Linear Induction Motor (LIM) Power Factor, Efficiency and Finite Element Considerations, E.B. dos Santos (Dr), J.R. Camacho (PhD), A.A. de Paula (MSc) and C.H. Salerno (Dr)

[2019-1] Duncan, J. Linear induction motor-equivalent-circuit model. IEE Proc. B Electr. Power Appl. 1983, 130, 51–57.

[2019-2] Lu, J.; Ma, W. Research on End Effect of Linear Induction Machine for High-Speed Industrial Transportation. IEEE Trans. Plasma Sci. 2011, 39, 116–120.

[2019-3] Hu, Y.; Cosic, A.; Östlund, S.; Hui, Z. Design and Optimization Procedure of a Single-Sided Linear Induction Motor Applied to an Articulated Funiculator. In Proceedings of the 2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia), Hefei, China, 22–26 May 2016; pp. 3096–3102.

[2019-4] Woronowicz, K.; Safaee, A. A novel linear induction motor equivalent-circuit with optimized end-effect model including partially-filled end slots. In Proceedings of the 2014 IEEE Transportation Electrification Conference and Expo (ITEC), Dearborn, MI, USA, 15–18 June 2014; pp. 1–5.

[2019-5] Boughrara, K.; Dubas, F.; Ibtiouen, R. 2-D Analytical Prediction of Eddy Currents, Circuit Model Parameters, and Steady-State Performances in Solid Rotor Induction Motors. IEEE Trans. Magn. 2014, 50, 1–14.

[2019-6] Gieras, J.F.; Eastham, A.R.; Dawson, G.E. Performance calculation for single-sided linear induction motors with a solid steel reaction plate under constant current excitation. IEE Proc. B Electr. Power Appl. 1985, 132, 185–194.

[2019-7] Boldea, I.; Babescu, M. Multilayer approach to the analysis of single-sided linear induction motors. Proc. Inst. Electr. Eng. 1978, 125, 283–287.

[2019-8] Overboom, T.; Smeets, J.; Jansen, J.; Lomonova, E. Semi-analytical modeling of a linear induction motor including primary slotting. In Proceedings of the 15th International Symposium on Electromagnetic Fields in Mechatronics (ISEF), Funchal, Portugal, 1–3 September 2011; pp. 1–8.

[2019-9] Pluk, K.J.W.; Jansen, J.W.; Lomonova, E.A. Hybrid Analytical Modeling: Fourier Modeling Combined With Mesh-Based Magnetic Equivalent Circuits. IEEE Trans. Magn. 2015, 51, 1–12.

[2019-10] Ouagued, S.; Diriye, A.A.; Amara, Y.; Barakat, G. A General Framework Based on a Hybrid Analytical Model for the Analysis and Design of Permanent Magnet Machines. IEEE Trans. Magn. 2015, 51, 1–4.

[2019-11] Bao, J.; Gysen, B.L.J.; Lomonova, E.A. Hybrid Analytical Modeling of Saturated Linear and Rotary Electrical Machines: Integration of Fourier Modeling and Magnetic Equivalent Circuits. IEEE Trans. Magn. 2018, 54, 1–5.

[2019-12] Ostovic, V. Dynamics of Saturated Electric Machines; Springer: Berlin, Germany, 1989.

[2019-13] Gysen, B.L.J.; Meessen, K.J.; Paulides, J.J.H.; Lomonova, E.A. General Formulation of the Electromagnetic Field Distribution in Machines and Devices Using Fourier Analysis. IEEE Trans. Magn. 2010, 46, 39–52.

[2019-14] Overboom, T.; Smeets, J.; Jansen, J.; Lomonova, E. Decoupled control of thrust and normal force in a double-layer single-sided linear induction motor. Mechatronics 2013, 23, 213–221.

[21] Aleksandrov, S.R.; Overboom, T.T.; Lomonova, E.A. Design Optimization and Performance Comparison of Two Linear Motor Topologies With PM-Less Tracks. IEEE Trans. Magn. 2018, 54, 1–8.

[22] Amrhein, M.; Krein, P.T. Force Calculation in 3-D Magnetic Equivalent Circuit Networks with a Maxwell Stress Tensor. IEEE Trans. Energy Convers. 2009, 24, 587–593.

[2019-17] Poynting, J.H. On the transfer of energy in the electromagnetic field. Philos. Trans. R. Soc. Lond. 1884, 175, 343–361.

[2019-18] Bolton, H. Transverse edge effect in sheet-rotor induction motors. Proc. Inst. Electr. Eng. 1969, 116, 725–731.

[20] Aleksandrov, S.R.; Overboom, T.T.; Lomonova, 2D Hybrid Steady-State Magnetic Field Model for Linear Induction Motors 2019

[2018-1] S. Wang, D. M. Miao, and J. X. Shen, “Optimal design of a linear induction motor for woodworking machine application,” in Proc. 17th Int. Conf. Elect. Mach. Syst. (ICEMS), Oct. 2014, pp. 3633–3637.

[2018-2] D. Hu, W. Xu, and R. Qu, “Electromagnetic design optimization of single-sided linear induction motor for improved drive performance based on linear metro application,” in Proc. Australas. Univ. Power Eng. Conf. (AUPEC), Sep. 2014, pp. 1–6.

[2018-3] S. Nonaka and T. Higuchi, “Design of single-sided linear induction motors for urban transit,” IEEE Trans. Veh. Technol., vol. TVT-37, no. 3, pp. 167–173, Aug. 1988.

[2018-4] S.-B. Yoon, J. Hur, and D.-S. Hyun, “A method of optimal design of single-sided linear induction motor for transit,” IEEE Trans. Magn., vol. 33, no. 5, pp. 4215–4217, Sep. 1997.

[2018-5] A. Shiri and A. Shoulaie, “Design optimization and analysis of singlesided linear induction motor, considering all phenomena,” IEEE Trans. Energy Convers., vol. 27, no. 2, pp. 516–525, Jun. 2012.

[2018-6] C. H. H. M. Custers, T. T. Overboom, J.W. Jansen, and E. A. Lomonova, “2-D semianalytical modeling of eddy currents in segmented structures,” IEEE Trans. Magn., vol. 51, no. 11, pp. 1–4, Nov. 2015.

[2018-7] J. Gieras, Linear Induction Drives (Monographs in Electrical and Electronic Engineering). Oxford, U.K.: Clarendon, 1994.

[2015-1] V. Ostovi´c, Dynamics of Saturated Electric Machines. New York, NY, USA: Springer-Verlag, 1989.

[2015-2] B. Hague, “The principles of electromagnetism, applied to electrical machines,” in Electromagnetic Problems in Electrical Engineering, 1st ed. New York, NY, USA: Dover, 1962.

[2015-3] 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.

[2015-4] J. W. Gibbs, “Fourier series,” Nature, vol. 59, p. 200, 1898.

[XX] Modeling of Flux Switching Permanent Magnet Machines With Fourier Analysis Bart L. J. Gysen, Esin Ilhan, Koen J. Meessen, Johannes J. H. Paulides, and Elena A. Lomonova

[23] Genetic Algorithms, Fernando Lobo University of Algarve

[19] Multi-objective optimal design of low-speed linear induction motor using genetic algorithm, Abbas SHIRI, Abbas SHOULAIE

[12] Osawa S., Wada M., Karita M., Ebihara D., T. Yokoi, Lightweight type linear induction motor and its characteristics, IEEE Trans. Magn, 28(1992), No. 4, 3003-3005.

[13] Kitamura M., Hino N., Nihei H., M. Ito, a direct search shape optimization based on complex expressions of 2-dimentional magnetic fields and forces, IEEE Trans. Magn, 34(1998), No. 5, 2845-2848.

[14] Laporte B., Takorabet N., Vinsard G., An approach to optimize winding design in linear induction motors, IEEE Trans. Magn, 33(1997), No. 2, 1844-1847.

[15] Mishima T., Hiraoka M., Nomura T., A study of the optimum stator winding arrangement of LIM in maglev systems, IEEE Inter. Conf. on Electric Machines Drives, IEMDC, (2005), 1238-1242.

[16] Hassanpour Isfahani A., Lesani H., Ebrahimi B. M., Design Optimization of Linear Induction Motor for Improved Efficiency and Power Factor, IEEE Inter. Conf. on Electric Machines Drives, IEMDC, (2007), 988-991.

[17] Hassanpour Isfahani A., Ebrahimi B. M., Lesani H., Design Optimization of a Low-Speed Single-Sided Linear Induction Motor for Improved Efficiency and Power Factor, IEEE Trans. Magn., 44(2008), No. 2, 266-272.

[18] Lucas C., Nasiri G. Z., Tootoonchian F., Application of an imperialist competitive algorithm to design of linear induction motor, Elsv. Engy. Conv. & Mang, 51(2010), 1407-1411.

[link (google)](https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3)

[Another Good Link For All EA Functionality (google)](https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_mutation.htm)

[I NEED TO IMPLEMENT THIS TO VISUALIZE PSO (google)](https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/)

# VITA AUCTORIS

|  |  |
| --- | --- |
| NAME: | **Michael Thamm** |
| PLACE OF BIRTH: | Dresden, GER |
| YEAR OF BIRTH: | 1997 |
| EDUCATION: | St.Anne High School, Belle River, ON, 2015  University of Windsor, B.A.Sc., Windsor, ON, 2019 |