**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:

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

H. Hu

Department of Mechanical, Automotive & Materials Engineering

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

B. Balasingam

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

# DEDICATION

# ACKNOWLEDGEMENTS

# TABLE OF CONTENTS

[DECLARATION OF CO-AUTHORSHIP/PREVIOUS PUBLICATION iii](#_Toc106915476)

[ABSTRACT v](#_Toc106915477)

[DEDICATION vi](#_Toc106915478)

[ACKNOWLEDGEMENTS vii](#_Toc106915479)

[TABLE OF CONTENTS viii](#_Toc106915480)

[LIST OF TABLES x](#_Toc106915481)

[LIST OF FIGURES xi](#_Toc106915482)

[LIST OF ABBREVIATIONS xiii](#_Toc106915483)

[NOMENCLATURE xiv](#_Toc106915484)

[CHAPTER 1 Introduction 16](#_Toc106915485)

[1.1. Electric Vehicles–A Green Alternative 16](#_Toc106915486)

[1.2. Motor Slot and Pole Count 17](#_Toc106915487)

[1.3. Literature Survey on Motor Modelling 18](#_Toc106915488)

[1.4. Induction Motor Optimization 22](#_Toc106915489)

[1.5. Research Motivations 23](#_Toc106915490)

[1.6. Research Objectives 24](#_Toc106915491)

[1.7. Research Contribution and Deliverables 25](#_Toc106915492)

[1.8. Organization of Thesis 26](#_Toc106915493)

[CHAPTER 2 Hybrid Analytical Model 27](#_Toc106915494)

[2.1. Base Model 27](#_Toc106915495)

[2.2. Model Relationships 31](#_Toc106915496)

[2.3. Hybrid Analytical Model 33](#_Toc106915497)

[2.4. System of Linear Equations 39](#_Toc106915498)

[2.5. Processed Model 44](#_Toc106915499)

[CHAPTER 3 Optimization Algorithm 45](#_Toc106915500)

[3.1. Genetic Algorithm 46](#_Toc106915501)

[3.2. Particle Swarm Optimization 48](#_Toc106915502)

[3.1. Schwefel Function Minimization Case Study 50](#_Toc106915503)

[3.2. NSGAII Configuration 57](#_Toc106915504)

[3.2.1. Selection 58](#_Toc106915505)

[3.2.2. Variation 60](#_Toc106915506)

[CHAPTER 4 Model Optimization Integration 62](#_Toc106915507)

[4.1. Compute Fitness 63](#_Toc106915508)

[4.2. Motor Feasibility 65](#_Toc106915509)

[4.3. Solver Configuration 67](#_Toc106915510)

[CHAPTER 5 Research Summary 67](#_Toc106915511)

[5.1. Conclusions 68](#_Toc106915512)

[5.2. Future Research on HAM and LIM Optimization 68](#_Toc106915513)

[REFERENCES 68](#_Toc106915514)

[VITA AUCTORIS 69](#_Toc106915515)

# LIST OF TABLES

[Table 1.1 18](#_Toc106915516)

[Slot and Pole Trend Decision-Making 18](#_Toc106915517)

[Table 1.2 19](#_Toc106915518)

[Modelling Algorithm Comparison 19](#_Toc106915519)

[Table 2.4 25](#_Toc106915520)

[Solution Deliverables 25](#_Toc106915521)

[Table 2.1 28](#_Toc106915522)

[Baseline Spatial Motor Parameters 28](#_Toc106915523)

[Table 2.2 29](#_Toc106915524)

[Baseline Electrical And Material Motor Parameters 29](#_Toc106915525)

[Table 2.3 29](#_Toc106915526)

[Baseline Model Parameters 29](#_Toc106915527)

[Table 2.4 34](#_Toc106915528)

[Node Index Continuity 34](#_Toc106915529)

[Table 2.5 43](#_Toc106915530)

[System of Linear Equations Solving for Unknown Variables 43](#_Toc106915531)

[Table 2.6 44](#_Toc106915532)

[Boundary Condition Summary 44](#_Toc106915533)

[Table 3.1 50](#_Toc106915534)

[PSO Velocity and Position Coefficients 50](#_Toc106915535)

[Table 3.2 53](#_Toc106915536)

[Optimization Algorithm Configuration 53](#_Toc106915537)

[Table 3.3 53](#_Toc106915538)

[Average Optimization Algorithm Results 53](#_Toc106915539)

[Table 3.4 54](#_Toc106915540)

[Algorithm Convergence Visualization 54](#_Toc106915541)

[Table 4.1 64](#_Toc106915542)

[Mass Equations 64](#_Toc106915543)

[Table 4.2 66](#_Toc106915544)

[Motor Feasibility Constraints 66](#_Toc106915545)

# LIST OF FIGURES

[Fig. 2.1. Fourier series approximation (red) of arbitrary discretized magnetic flux waveform (blue) 21](#_Toc106551891)

[Fig. 2.1. Foundation of MEC modelling between 4 nodes within 3 different materials (yellow, blue, red). 22](#_Toc106551892)

[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). 29](#_Toc106551893)

[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). 30](#_Toc106551894)

[Fig. 2.3. Meshed motor model containing boundary conditions 32](#_Toc106551895)

[Fig. 2.4. Single MEC node element at index 33](#_Toc106551896)

[Fig. 2.4. 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 35](#_Toc106551897)

[Fig. 3.1. Layout of a genetic algorithm with an arbitrary number of chromosomes and genes per population. 44](#_Toc106551898)

[Fig. 3.2. Layout of a genetic algorithm execution loop. 45](#_Toc106551899)

[Fig. 3.3. Layout of a particle swarm optimization algorithm optimization loop. 47](#_Toc106551900)

[Fig. 3.4. Surface plot of the Schwefel function on the input range. 49](#_Toc106551901)

[Fig. 3.5. Contour plot of the Schwefel function on the input range highlighting the global minimum with a red cross. 50](#_Toc106551902)

[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. 54](#_Toc106551903)

[Fig. 3.7. Layout of a Tournament selection algorithm using arbitrary objective values to highlight the winning decision based on a minimization problem. 57](#_Toc106551904)

[Fig. 3.8. Visualization of crossover between two parent variables to produce two child variables governed by the crossover point. 58](#_Toc106551905)

[Fig. 3.9. Layout of a genetic algorithm with an arbitrary number of chromosomes and genes per population. 59](#_Toc106551906)

[Fig. 4.1. Layout of the motor optimization algorithm inputs and the resultant multi-objectives. 61](#_Toc106551907)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| LEM | Linear Electric Motor |
| REM | Rotary Electric Motor |
| LIM | Linear Induction Motor |
| LSM | Linear Synchronous Motor |
| 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 |
|  | Pole pitch |
|  | 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 |
|  | 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 |

# Introduction

## Electric Vehicles–A Green Alternative

In 2009 the European commission for science and environmental policy [[X]](https://ec.europa.eu/environment/integration/research/newsalert/pdf/159na4_en.pdf) 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 [[X]](https://ourworldindata.org/co2-emissions). 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 [[X]](https://www.iea.org/topics/transport), it is imperative that there be an initiative which can alleviate some of this contribution. According to the government of Canada [[X]](https://www.nrcan.gc.ca/energy-efficiency/transportation-alternative-fuels/personal-vehicles/choosing-right-vehicle/buying-electric-vehicle/21034) 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 danger 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 [[X]](https://en.wikipedia.org/wiki/ICE_3) 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 [[X]](https://swissloop.ch/claude-nicollier/). 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

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. The fewer pole pairs a motor has, the less drag the motor experiences and therefore the more thrust it generates [17], [19]. To achieve greater efficiency, the generated wave in the primary field shall approximate a sin wave. This approach is called distributed windings 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 |

Then talk about different winding configurations with a winding diagram.

## 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 choosing the best modelling algorithms within a workflow proves to be challenging. Within table x 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 x. This allows complex geometries to be accurately modelled using MEC regions and simple geometries with HM regions to achieve greater efficiency and accuracy.

HM is a technique used to approximate waveforms with Fourier analysis. The approximation of the waveform is in the form of a summation of N space harmonics, where is the current harmonic number. Each harmonic has its own unknown variables that need to be solved in the matrix equation. The function value at position with a period of defines the function value as a complex Fourier series. Substituting equation X into equation X produces equation X which allows for waveform approximation.

Chart

Description automatically generated with medium confidence

Fig. 1.1. Fourier series approximation (red) of arbitrary discretized magnetic flux waveform (blue)

MEC modelling is achieved by discretizing its regions into a mesh of nodes each of which contain unique spatial and magnetic properties. Hopkinson's law defines the flux, reluctance, and the MMF which have a superficial resemblance to Ohm’s law when solving electric circuits.

The model works on a conservation of flux principle, stating that the amount of flux entering a polygon must also exit the polygon.

Chart, box and whisker chart

Description automatically generated

Fig. 1.2. Foundation of MEC modelling between 4 nodes within 3 different materials (yellow, blue, red).

## 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.

## Research Motivations

In reference [7] the primary weight has been considered for optimization. In other work, the thrust and power to weight ratio are maximized [8]. In [9] and [10] optimum winding design of LIM have been presented. In other research, optimal design of the LIM for having the maximum efficiency and power factor has been done [11, 12]. 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 x 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

Using a base model as a reference for improvement allows for each produced solution from the solver to be objectively compared. The base model found in [[X]](C://GitHub/Masters/SupportingDocs/Papers/HAM_2019.pdf) acts as the current standard to improve upon which 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 the base model in one performance parameter while sacrificing other performance parameters. It is important to create Pareto-optimal solutions that outperform the base model in most performance parameters, if not all. 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. 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 paper a novel holistic model is proposed which includes flexible motor modelling, modified constraints and losses, 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:

Table 2.4

Solution Deliverables

| ***Deliverable*** | ***Amount (%)*** |
| --- | --- |
| Improve efficiency | **+2** |
| Improve thrust | **+5** |
| Improve weight | **-5** |

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 function, 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 as well.
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 [X], 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 X, Y, and Z respectively. To account for transverse end-effects in the motor, the conductivity of the aluminum plate was reduced accordingly in table X. The number of node elements contained in rows and columns presented in table Z 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 X, Y, and Z.



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).



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).

## 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. To calculate the length of the LIM primary, the summation of the individual slot and tooth lengths produce equation X. 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 scalar in equation X, the end tooth width is now related to the slot width.

Through substitution of equations X, Y, and Z into equation A the only unknown variable remaining is . After factoring and isolation, the slot width is solved in equation B 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 X. For a constant frequency the velocity is proportional to the pole pitch which is approximated in equation X for a given motor primary length.

Since the motor application demands a high operational velocity the pole pitch shall be maximized.

As seen from Faraday’s Law, the number of turns, 𝑁, is directly related to the induced voltage across a set of coils. The relationship is rewritten here for clarity.

ℰ = −𝑁𝑑/ 𝑑𝑡 (Φ𝐵)

Here, ℰ is the electromotive force or EMF and it is proportional to an induced voltage. By increasing the number of turns in a coil, more voltage will be induced across its windings

## Hybrid Analytical Model

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 X 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 generated

Fig. 2.3. 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 x.

Table 2.4

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. 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 X 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 x. 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 x, y, and z 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. 2.5. 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

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 x. 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 2.4. 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, , through the diffusion in equation x.

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 x 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.

I CAN ADD THE LOCALIZED MESH DENSITY PICTURES AND ELABORATE HERE

## 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 2018 paper defines this as 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 equation x 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 X(By=By) can be implemented at the boundary using Equation X (flux conservation one) to produce the equations:

The flux in the normal direction in equation X is then substituted with equation X 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 X and Y above 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 X(Hx=Hx) 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 X which expands on the Fourier equations X, Y, Z. The value for is substituted with a summation of nodes in the x-direction for .

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):

The tangential and normal magnetic flux densities in the middle of the air gap, between the primary and secondary, can be plotted to check the piece-wise continuous waveform produced by the coils in the primary. I can prove that we don’t need the c\_0 term in the complex fourier transform by this image: **Graphical user interface, histogram

Description automatically generated Graphical user interface, chart, histogram

Description automatically generated**

The Bx field is piecewise-continuous and is plotted in Blue. The complex Fourier transform was applied to the Bx field and plotted in Red. The accuracy of the complex Fourier transform depends on: # of harmonics, # of x positions, # of nodes in the x-direction of the model

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 2.5

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 2.6

Boundary Condition Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | Continuous | | MEC | Non-continuous |
| Top | Bottom |
|  | X | X | X | X |
|  | X | X | X | X |

## 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 2.3 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 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. With this information a rough estimate on the efficiency of the motor can be produced which doubles as an objective function value for the GA.

# 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 compare them 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 X 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 through 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 through the means of 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 terminology, 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 X and Y.

(1)

(2)

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 allows for variation of the swarm and increasing 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 X. 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 X 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 X 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 X is achieved by selecting the early iterations of each solver. This comparison is found in Table X 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

|  |  |  |
| --- | --- | --- |
|  | GA | PSO |
| Iteration 1 | Diagram, background pattern  Description automatically generated |  |
| Iteration 3 | Background pattern  Description automatically generated |  |
| Iteration 5 |  |  |

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 X 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 fashion.

### 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 with 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, 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.

To determine the best configuration for the Tournament selection, experiment results across many configurations were tabulated in Table X, which optimized the Schwefel test function. The Schwefel test function is defined in section 2.4 and will have a consistent configuration throughout each test in the case study to isolate the effect of the Tournament selection configurations.

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

[Good link for mutation and selection](https://www.fernandolobo.info/ec1920/lectures/GAs-2.pdf)

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

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

### Variation

Like real life, 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 X. 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. Since 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 and the solver may become unstable. This is due to the children’s variable initialization relying on some form of randomized initialization which will resist solver convergence.

Mutation is another important function of the 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 X, 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.

# 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 x 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. 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 whereas in this chapter, the integration of these states with HAM will be discussed in more detail.

Diagram

Description automatically generated

## Compute Fitness

After solving the *Compute HAM* state, 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 x which are outputs of each motor produced by the HAM. By optimizing for thrust, mass, and efficiency the optimal motors will have a larger thrust-weight ratio while ensuring that the energy consumption to accomplish the latter is reasonable. 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.

A picture containing text, electronics

Description automatically generated

Fig. 4.1. 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.

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 volume’s times material density which are found in table x for each material.

Table 4.1

Mass Equations

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

From section 2.5 the definition of the thrust force acting on the primary relative to the secondary is provided as well as the conduction losses inside the secondary. The overall efficiency is calculated as:

Where is the primary input phase voltage, is the primary phase current, and is the phase angle of the total impedance of single-phase equivalent circuit.

I think the angle comes from the impedance angle since there is not capacitor in the circuit. So figure out Z = R + jwL

Text

Description automatically generated

Text

Description automatically generated

BRO THIS IS HUGE – follow this eqn for inductance of square coil:  
<https://www.allaboutcircuits.com/tools/rectangle-loop-inductance-calculator/>

Z = jwL where w = 2\*pi\*freq

## Motor Feasibility

To ensure that all motors produced by the optimization algorithm are feasible, the electromagnetic rules in table x were followed. 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 4.2

Motor Feasibility Constraints

|  |  |
| --- | --- |
| **Rule** | **Explanation** |
|  | Monopoles cannot exist |
| 0.75 < q < 8.0 | The pole count relative to slots should not be too large |
|  | The application demands a positive thrust and a negative normal force |
| B < 1.7T | The primary core material saturates at 1.7T which must be avoided |
|  | The number of coil turns should produce a wire diameter capable of conducting the required amperage while not exceeding the slot area |
|  | 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 x. The skin depth is calculated assuming low frequencies [15] 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 x.

Shape

Description automatically generated

Fig. 4.1. 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 larger motor sizes which may have their operating frequency constrained by the secondary design.

Using the skin depth procedure to determine the maximum frequency of operation, the theoretical top speed of the motor can be solved. One reason this is a theoretical is due to the required input voltage to overcome the equivalent impedance and resistance of the motor to continue driving enough current to produce enough 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. This voltage is important for the calculation of efficiency

## Solver Configuration

* In this section discuss the final solver configuration numbers like in chapter 3 for variation, selection to ensure that the motor domain is properly searched

# Research Summary

A specific field type can be plotted on the mesh to visualize its magnitude throughout the mesh. The magnitude for the range of values for the field type are translated to a colour gradient and plotted on the mesh to define the field. This is important for visualizing the magnetic flux density in the core of the primary to check that the core dimensions are compatible with the localizations of flux density throughout the core.

## Conclusions

## Future Research on HAM and LIM Optimization

# REFERENCES

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

# 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 |