FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Low-Cost MagLev System for Cartesian Robots

Martinho Gomes de Figueiredo

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Maria Teresa Duarte Barroca Delgado

March 28, 2025

Resumo

A rápida evolução das tecnologias de fabricação modernas está a impulsionar a inovação em múltiplos domínios, com a manufatura aditiva a emergir como uma abordagem revolucionária para a produção de componentes. As técnicas aditivas permitem a criação de geometrias complexas que seriam pouco práticas ou impossíveis de fabricar através de métodos convencionais. Esta flexibilidade abre novas oportunidades para o desenvolvimento de dispositivos eletromecânicos inovadores, incluindo motores com configurações inéditas que podem melhorar a eficiência, a compactação e a adaptabilidade. No entanto, apesar do seu grande potencial, a manufatura aditiva ainda é uma tecnologia relativamente dispendiosa e em evolução, especialmente no que diz respeito à fabricação de componentes eletromecânicos funcionais.

No contexto do desenvolvimento de motores elétricos, as técnicas de fabricação convencionais impõem limitações à liberdade de *design*, devido a restrições associadas às propriedades dos materiais, às tolerâncias de produção e à complexidade da montagem. A manufatura aditiva tem o potencial de superar muitos destes desafios ao permitir a produção direta de estruturas motoras complexas, reduzindo o número de peças e possibilitando a personalização de percursos magnéticos e condutivos. No entanto, o estado atual das tecnologias aditivas, particularmente para aplicações de alta precisão, ainda apresenta desafios como limitações de materiais, robustez mecânica e viabilidade económica, impedindo a sua adoção generalizada na produção de motores.

Esta dissertação investiga uma abordagem alternativa, centrando-se no design e desenvolvimento de um motor planar inspirado na cinemática dos robôs delta. Os robôs delta são conhecidos pela sua estrutura cinemática paralela, que permite movimentos rápidos e precisos, e pela sua arquitetura, que apresenta, normalmente, um ângulo de 120º entre os seus três eixos rotativos. Adaptando este conceito a uma configuração de motor planar, este estudo analisa como um sistema deste tipo pode ser aplicado em cenários que exigem movimentos bidirecionais rápidos e precisos, sem depender de atuadores rotativos convencionais.

Dado que as técnicas de manufatura aditiva capazes de fabricar este motor de forma eficiente ainda estão em desenvolvimento, esta investigação explora um método de fabricação alternativo e mais acessível: a construção baseada em tecnologia de PCB. A tecnologia de circuitos impressos (PCB) oferece uma abordagem bem estabelecida, escalável e economicamente viável para a produção de motores, tirando partido de processos avançados de gravação, estruturação em múltiplas camadas e integração de percursos condutores. Ao utilizar esta tecnologia, este trabalho demonstra como um motor planar pode ser concebido com controlo geométrico preciso, mantendo a viabilidade de fabricação com os métodos atuais de produção.

Os resultados desta investigação fornecem uma visão sobre a viabilidade de um motor planar fabricado com tecnologia PCB, destacando tanto as suas vantagens como as suas limitações. Avanços futuros na manufatura aditiva poderão aperfeiçoar ainda mais este conceito, permitindo maior flexibilidade de design e melhor desempenho. Em última análise, este trabalho contribui para a exploração contínua de novas arquiteturas de motores e demonstra como as técnicas de fabricação existentes e emergentes podem ser aproveitadas para expandir os limites do design eletromecânico.

Palavras-Chave: Motor Planar, Motor Baseado em PCB, *Design* Eletromecânico, Manufactura Avançada, Estrutura Multicamadas

Abstract

The rapid evolution of modern manufacturing technologies is driving innovation across multiple fields, with additive manufacturing emerging as a transformative approach to fabrication. Additive techniques enable the creation of complex geometries that were previously impractical or impossible to manufacture using conventional methods. This flexibility opens doors for the development of new electromechanical devices, including motors with novel configurations that could lead to advancements in efficiency, compactness, and adaptability. However, while additive manufacturing holds great promise, it remains a relatively expensive and developing technology, particularly when applied to the fabrication of functional electromechanical components.

In the context of electric motor development, conventional manufacturing techniques impose limitations on design freedom due to constraints related to material properties, production tolerances, and assembly complexity. Additive manufacturing has the potential to overcome many of these challenges by enabling the direct fabrication of intricate motor structures, reducing part count, and allowing for customized magnetic and conductive pathways. Despite these advantages, the current state of additive technologies, particularly for high-precision applications, still presents challenges such as material limitations, mechanical robustness, and cost-effectiveness, preventing widespread adoption in motor production.

This dissertation explores an alternative approach by focusing on the design and development of a planar motor inspired by delta robot kinematics. Delta robots are known for their parallel kinematic structure, which provides high-speed and precise motion control, and their architecture usually features a 120º angle between its three rotary axis. By adapting this concept to a planar motor configuration, this work investigates how such a system could be used for applications requiring high-speed and precise bidirectional movement without relying on traditional rotary actuators.

Given that additive manufacturing techniques capable of efficiently producing such a motor are still evolving, this research focuses on an alternative, more accessible fabrication method: PCB-based construction. Printed Circuit Board (PCB) technology offers a well-established, scalable, and cost-effective approach to motor fabrication, leveraging advanced etching, multilayer structuring, and integrated conductive pathways. By utilizing PCB technology, this work demonstrates how a planar motor can be conceived with precise geometric control, while maintaining manufacturing feasibility with current production methods.

The results of this research provide insights into the feasibility of a PCB-manufactured planar motor, highlighting both its advantages and limitations. Future advancements in additive manufacturing may further refine this concept, allowing for greater design flexibility and improved performance characteristics. Ultimately, this work contributes to the ongoing exploration of novel motor architectures and demonstrates how existing and emerging manufacturing techniques can be leveraged to push the boundaries of electromechanical design.

Keywords: Planar Motor, PCB-Based Motor, Electromechanical Design, Advanced Manufacturing, Multilayer Structure

Agradecimentos

À minha orientadora, por ter aceitado esta proposta e por toda a disponibilidade e ajuda que ofereceu tanto em termos académicos como motivacionais.

Aos meus pais, por me darem a oportunidade de aprender com os meus erros, e por estarem sempre lá para me ajudar a recuperar.

À minha irmã, que me oferece todas a suas perpesctivas do mundo alargando mais a minha.

À minha família, por acreditarem em mim e nas minhas capacidades.

À minha namorada, por se deixar ser enganada durante tanto tempo.

Aos meus avós, espero fazer algo de que se possam orgulhar.

Obrigado

Martinho Figueiredo

United Nations Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) are a set of 17 goals aiming to address pressing global challenges. These were set in 2015 as part of the 2030 Agenda for Sustainable Development, serving as a universal call to action to address critical economic, social, and environmental issues worldwide. [4]

Figure 1: United Nations Sustainable Development Goals (SDG)

This dissertation work focuses on a different approach to planar motor technology, in order to develop a low-cost device that can be employed in a myriad of applications. By developing an affordable and versatile design, this project aligns with and contributes to the advancement of multiple SDGs.

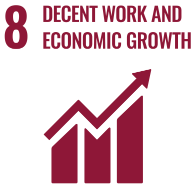


Figure 2: SDG 8 - Decent Work and Economic Growth

Decent Work and Economic Growth

- **Foster Economic Opportunities** — The development of this device can broaden accessibility to advanced technology, encouraging its adoption in industry, small businesses, and research institutions. Due to the open-source nature of the project along with the hardware's reduced cost, this work makes innovation attainable for a wider range of audience, which in turn will foster new markets and economic activities.
- **Support Workforce Development** — The system's affordability and compactness make it an excellent educational and training tool for technical fields such as mechatronics, robotics, and precision manufacturing. This contributes toward the upskilling of the workforce, improving employment opportunities in emerging high-tech industries.
- **Encourage Sustainable Growth** — By offering a cost-effective alternative to traditional technologies, the system enables businesses to optimize their operations with minimal capital investment. This contributes to sustainable economic growth, particularly for small and medium enterprises.

Figure 3: SDG 9 - Industry, Innovation, and Infrastructure

Industry, Innovation, and Infrastructure

- **Promote Innovation** — This system serves as a platform for further technological innovation. Its adaptability for various applications encourages experimentation and the development of new solutions across multiple industries.
- **Enable Sustainable Infrastructure** — MagLev technology has inherent benefits such as reduced friction and wear, leading to longer-lasting and more efficient systems. The miniaturization and affordability of the technology pave the way for its integration into sustainable infrastructure, such as energy-efficient manufacturing processes or micro-scale transportation systems.

- **Strengthen Industrial Capacities** — The research and development of systems like this strengthen local industrial capabilities, especially in the manufacturing and engineering sectors. Additionally, it aligns with the goal of building resilient infrastructure by providing innovative alternatives to traditional, resource-intensive systems.

Figure 4: SDG 12 - Responsible Consumption and Production

Responsible Consumption and Production

- **Reduce Material Waste** — This device's operational principle, based on non-contact levitation, reduces wear and tear, minimizing the need for frequent replacements and, consequently, reducing waste production.
- **Encourage Energy Efficiency** — MagLev systems are energy-efficient by nature, due to the lack of mechanical friction. The system's small scale design further reduces energy requirements, making it a sustainable choice for small-scale applications.
- **Facilitate Sustainable Research and Production** — This work can empower researchers and industries to prototype, test, and deploy sustainable technologies, creating products and systems that align with eco-friendly and resource-efficient principles.

"The difference between screwing around and science is writing it down."

Adam Savage

Contents

List of Figures

List of Tables

Abbreviations and Symbols

BFS	Best-First Search
CAD	Computer-Aided Design
DM3	Diamagnetic Micro Manipulation
DRC	Design Rule Check
EDA	Electronic Design Automation
ERC	Electrical Rule Check
FDTD	Finite-Difference Time-Domain
FEA	Finite Element Analysis
FEM	Finite Element Method
FOSS	Free Open Source Software
FR4	Fire Retardant 4
MAGLEV	Magnetic Levitation
PCB	Printed Circuit Board
PMSM	Permanent Magnet Synchronous Motor
PWM	Pulse-Width Modulation SDG
Sustainable Development Goals	
SRI	Stanford Research Institute
STEP	Standard for the Exchange of Product model data
UWB	University of West Bohemia

Chapter 1

Introduction

In this chapter, the motivation behind undertaking this research is presented, along with the project objectives, highlighting key factors that inspired the work. Lastly, the structure of the dissertation is introduced, providing a roadmap for the remaining chapters.

1.1 Motivation

The motivation for this research stems from the desire to explore innovative applications for printed circuit boards (PCBs) in electromechanical systems, moving beyond their traditional role as passive electrical interconnects.

Initially inspired by the challenge of designing a chessboard capable of moving its pieces autonomously, the idea of employing PCBs as active components in planar motors seemed like a plausible solution and, thus, became the focus point for this study. This concept highlights the potential of repurposing a common and widely used component to serve both electrical and mechanical functions.

By reimagining the role of this ubiquitous and cost-effective technology, this research contributes toward the development of compact, efficient, and versatile systems that integrate electronics and mechanics seamlessly.

1.2 Objectives

This dissertation work focuses on three key objectives that it aims to achieve:

1. **Framework Development** — Development of a program capable of generating different PCB layouts tailored for planar motor applications, while taking into account the necessary design choices to ensure that the designs are manufacturable.

2. **Simulation Development** — To accelerate the development process and reduce the costs associated with physical prototyping, a simulation framework is presented. This framework

models the electromagnetic behavior of the designed PCB planar motors, enabling iterative design improvements and performance predictions.

3. **Application Potential** — Emphasize the versatility of PCB planar motors in automation and compact system design. This research aims to highlight the advantages of this technology, such as reduced system size, increased integration, and cost-effective manufacturing, offering insights into potential real-world implementations.

By addressing these three aspects, this dissertation aims to establish a comprehensive foundation for the adoption and further development of PCB planar motors, bridging the gap between concept and application while contributing to the broader field of electromechanical innovation.

1.3 Dissertation Structure

Following this introductory chapter, this dissertation is organized into four additional chapters.

Chapter 2 traces the evolution of electromagnetic motion, beginning with conventional rotary motors and advancing to linear motors that mitigate mechanical transmission losses for direct motion. The chapter further explores planar motors, which achieve precise two-dimensional control via perpendicular linear actuators with varied configurations and trade-offs. Finally, it presents a design and simulation workflow for PCB-based planar motors using open-source tools like KiCad, FreeCAD, ElmerFEM, and ParaView, thereby enhancing accessibility and reproducibility in research and development.

Chapter 3 outlines a methodology for generating and implementing the winding layout of a PCB-based planar motor for maglev cartesian robots. It introduces a hexagonal grid structure that optimizes space utilization and routing in order to support a scalable, modular design, aligned with a tri-axis control strategy. The process automates winding pattern creation, identifies critical via points for inter-layer connectivity, and assigns electrical nets, while enforcing manufacturability through KiCad's Design Rule Check (DRC) to meet industry standards for trace width, via size, and spacing. Additionally, the integration of the BD6342FV-E2 driver and a caching mechanism for circuit layout streamlines design complexity and export processes, with KiCad-based STEP file exports preparing the model for 3D simulation and analysis.

Chapter 4 outlines the meshing process integral to simulating the PCB planar motor and describes the preliminary testing setup. It details the methodology for generating the mesh and defining the excitation conditions, culminating on a simulation process that offers qualitative insights rather than exact real-world performance predictions.

Chapter 5 concludes that the framework for designing PCB planar motors was successfully developed using open-source tools, offering a structured and accessible approach for the research community and demonstrating manufacturability despite slightly higher costs due to design constraints. It also reveals that while the simulation process provided valuable insights, challenges with mesh complexity and accuracy remain, meaning improvements can be made in the simulation techniques. Looking forward, future work can explore transitioning to a 6-layer PCB in order to enhance performance and routing efficiency, while addressing the trade-offs of increased manufacturing costs and simulation intricacies.

Chapter 2

State of the Art

2.1 Introduction

Since the beginning of the study of electromagnetic phenomenona, one of its crucial properties that has been subject to extensive research is the ability to generate motion over a distance. Throughout the years, this ability to impart force over what appears to be empty space has been crucial for the development of electrical machines, ranging from the birth of the galvanometer, which was the first device created capable of measuring such effects, to the first homopolar electrical motors created by James Faraday.

The first type of motors to be developed and used in industry, were brushed rotary motors. These lack the need for control electronics, since they make use of a commutator ring and brushes to alternate the polarity of the fields created. This format quickly became the standard way to generate motion, however, since this motion would only be put to use after being transmitted through a myriad of mechanical devices, this approach reduced the possible efficiency of a system.

The rotary motor field has evolved a lot managing to develop motors with higher efficiency, however their overall morphology has stayed more or less the same, consisting mainly of a rotor and stator. The stator is composed by a grouping of coils arranged in a specific configuration, so that it creates alternating magnetic fields facing the rotor. These coils are wound around an iron core that serves as a conduit for the magnetic field . The stator in turn, either has a DC constant field source (electromagnet) or a permanent magnet that will be attracted and repelled to the field generated by the rotor coils.

A different approach was developed by Nikola Tesla, in which an induction motor employs alternating current in multiple phases, to create a rotating magnetic field that induces a current in the rotor. This creates an opposing field to the stator's, thus, achieving magnetic coupling between the rotor and the stator, generating the rotation motion intended. This in particular was a significant development in the industry, solving a lot of the problems required for the widespread adoption of electrical motors and electricity in general.

Due to the usefulness of these rotary electrical machines, they were introduced in mechanical devices as a driving force for a multitude of jobs. Generally, in order to be able to transfer the motion generated to the place where it is actually needed, these machines would employ transmission mechanisms, which convert rotary motion into linear motion. This leads to a logical question — if the motion generated is to be linear, why not create a motor that has the necessary morphology to achieve it without any assistance of an external mechanism.

2.2 Linear Motors

In parallel with Faraday's exploration into rotary motors, Charles Wheatstone developed a linear motor prototype. This marks the beginning of linear motors and constituted a great innovation, even though his first prototype lacked efficiency for any real-world applications. In opposition to its counterpart, this motor type translates its field along an axis instead of rotating it.

Despite being heavily used in many industrial and transport applications such as manufacturing, MAGLEV trains and others [7][8][15], this technology hasn't found its way onto the consumer market and most people have seen very few examples of this type of motor in their daily lives.

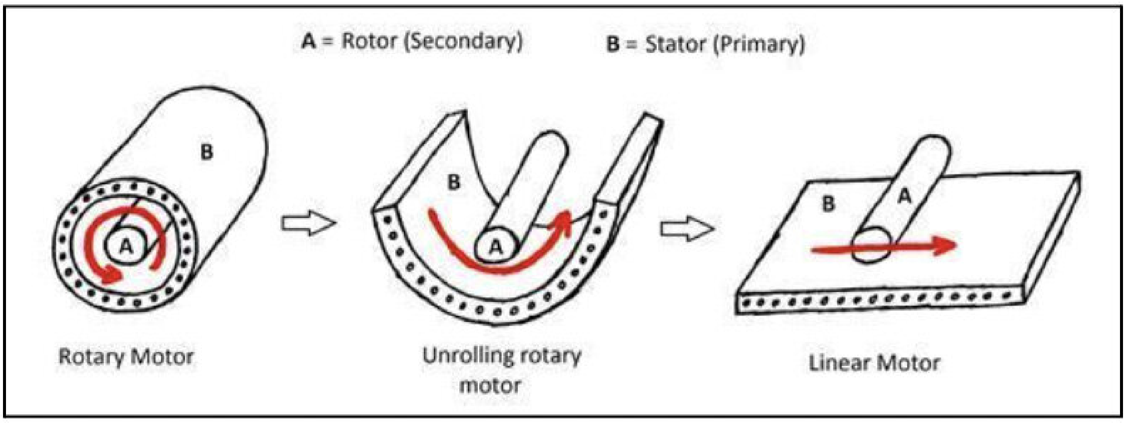


Figure 2.1: Conceptual Transformation from Rotary to Linear Actuator [5]

The recent increase in demand for technologies like 3D printing has brought more attention toward these types of motors, as shown by the creation of the Peopoly Magneto X, one of the first machines on the consumer level employing two linear motors in its main gantry.

2.3 Planar Motors

The interest in developing planar motors stems from the industry's necessity for precise machine control in a cartesian plane. In literature, the use of two linear motors with a 90º offset between them is the most prevalent solution. This technology has evolved to accommodate a myriad of sizes and shapes, each with different capabilities, such as differing carrying capacity, accelerations and footprints. [19]

As depicted in Figure 2.2, there are three major configurations for planar actuators:

- **Type A** — In this configuration, the actuator coils are positioned outside the operational area (arena) and exert forces to push or pull the actuated body within it. This setup is suboptimal when independent movement of multiple bodies is required, as external coils cannot provide precise individual control over each body.
- **Type B** — This arrangement involves a tessellation of coils whose normals are parallel to the normal of the arena plane. While this design allows for improved control — assuming each coil can be individually manipulated — it limits the resolution of the actuated body's movement to the size of the smallest coil. Essentially, the smallest coil determines the finest level of control achievable, which may hinder precision.
- **Type C** — This configuration utilizes two or more planar coils that are offset along the z-axis and typically rotated 90 degrees relative to each other to achieve movement along two axes. Although this setup enables multi-axis control, it still has limitations in providing precise, independent movement, especially when dealing with multiple bodies or requiring high-resolution actuation.

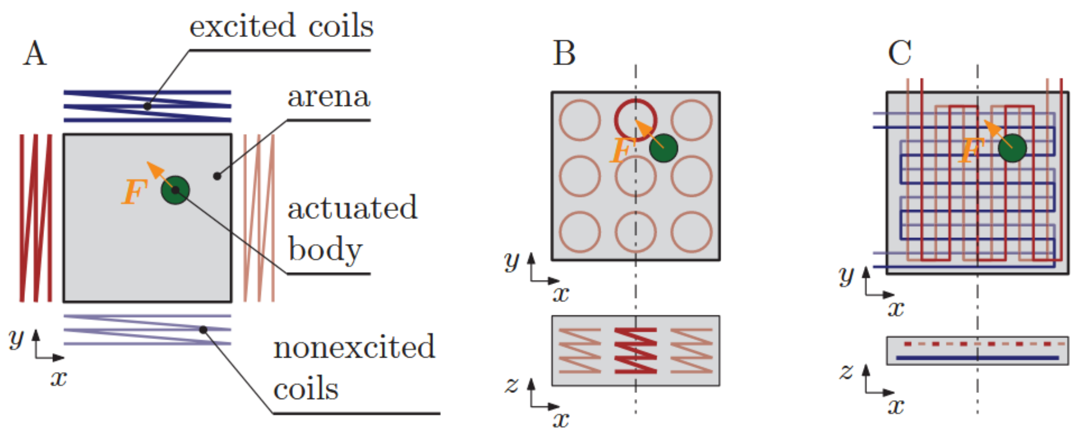


Figure 2.2: Archetype of Planar Actuators [14]

2.4 Printed Circuit Boards

Printed circuit boards (PCB) are a common way to develop a physical object that contains all the wire interconnects between components, needed for the proper assembly of a circuit. These mainly consist of a fiber glass substrate (FR4 is commonly used) where planes of copper are deposited and then milled or etched to achieve the shape required for a particular circuit.

A lot of effort has been made to create tools that simulate the behavior of a circuit after its is assembled in a PCB, ensuring the layout does not interfere with the operation of the circuit, like in crosstalk scenarios. Hence, there are several software packages available that are capable of estimating device parameters such as the field at a certain point or plane in space, when given the proper input signals.

2.4.1 Examples of Planar PCB Motors

2.4.1.1 MicroFactory

SRI International has developed micro-robots controlled by magnetic fields generated by printed circuit board (PCB) substrates, utilizing Diamagnetic Micro Manipulation (DM3) technology in a multi-agent system context. These robots operate at small dimensions and achieve movement with two degrees of freedom (x and y axes) by switching currents through specific traces on the PCB. A diamagnetic graphite layer on the PCB surface creates a passive levitation force by repelling the robot's ferromagnetic body, effectively eliminating or reducing friction depending on the thickness of the graphite layer — 0.5 mm thickness enables levitation, while 25 to 100 µm allows for diamagnetic sliding. [18]

Parallel actuation is accomplished by dividing the workspace into multiple independently controlled segments. Each segment, equipped with four coils, can move one robot separately in x and y directions. If multiple robots are present in a single segment, they move simultaneously in the same direction until one transitions into another segment. This system enables the operation of hundreds of robots simultaneously, greatly enhancing manufacturing capabilities beyond what is possible with single-robot systems. [18]

The micro-robots are notably fast, reaching speeds up to 55 cm/s, and can maneuver in any orientation, including climbing vertically or navigating flexible circuits. By attaching simple end effectors like metal wires, these robots can perform practical tasks such as transporting thin carbon rods for assembling high-strength, lightweight truss structures. SRI has successfully created truss structures up to 29 cm long that can support weights of up to 1 kg, demonstrating the robots' potential for efficient and scalable manufacturing applications. [18]

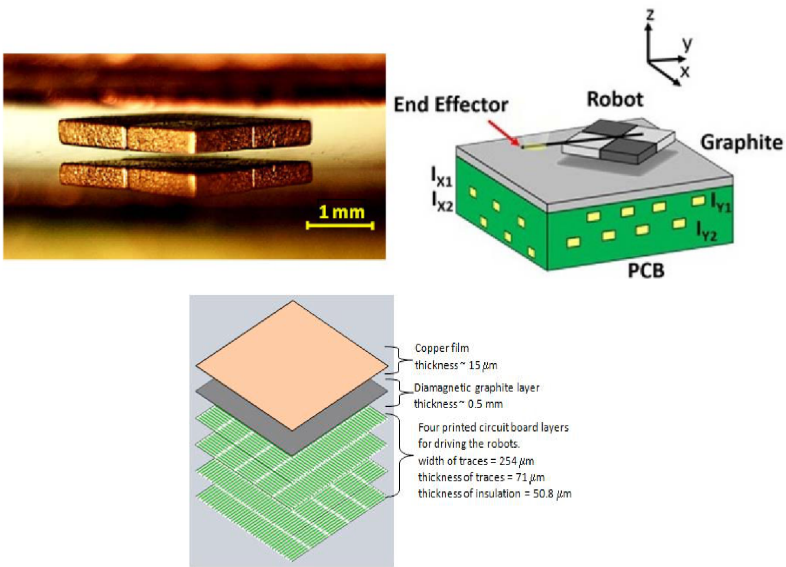
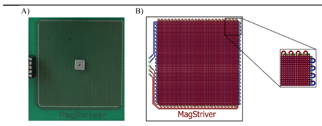
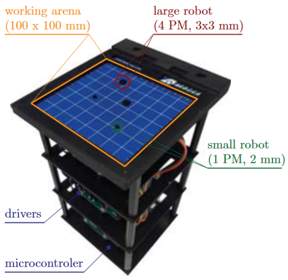
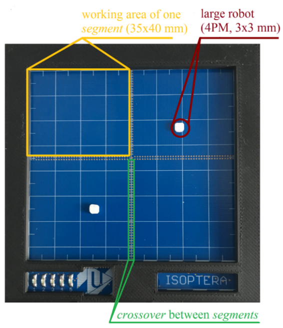
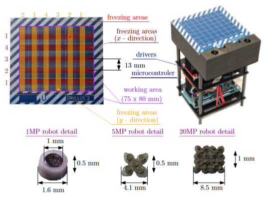


Figure 2.3: DM3 System [17] [18]

2.4.1.2 University of West Bohemia - CIMRA

The University of West Bohemia, through the CIMRA, has developed a series of different prototypes of the type C configuration (Figure 2.2), with different levels of control.

Table 2.1: List of Prototypes Developed at UWB

Image	**Name**	**Description**
	MagStriver	First basic implementation of the concept with 2 planar coils in each direction. [9][10]
	Scarabeus	Optimizing on the previous iteration, with the goal of minimizing the traces while maintaining performance. [9]
	Isoptera	This is iteration focuses on independent control by creating 4 separate zones. However, inside of each zone all robots will behave identically. [9]
	Gryllus[2]	The lastest version solves the problem of independent actuation by creating another set of planar coils responsible for "freezing" the motions of the robot. Selecting with which sections to freeze in sequence allows this platform to achieve independent control. [9]

2.4.1.3 Wuhan University

Another recent prototype comes from the university of Wuhan where Zou et al. [16] developed a system of the type B configuration (Figure 2.2). This was achieved with a 16-layer PCB, using the last layer to connect the driving circuit for each coil, as well as connecting them together in a matrix.

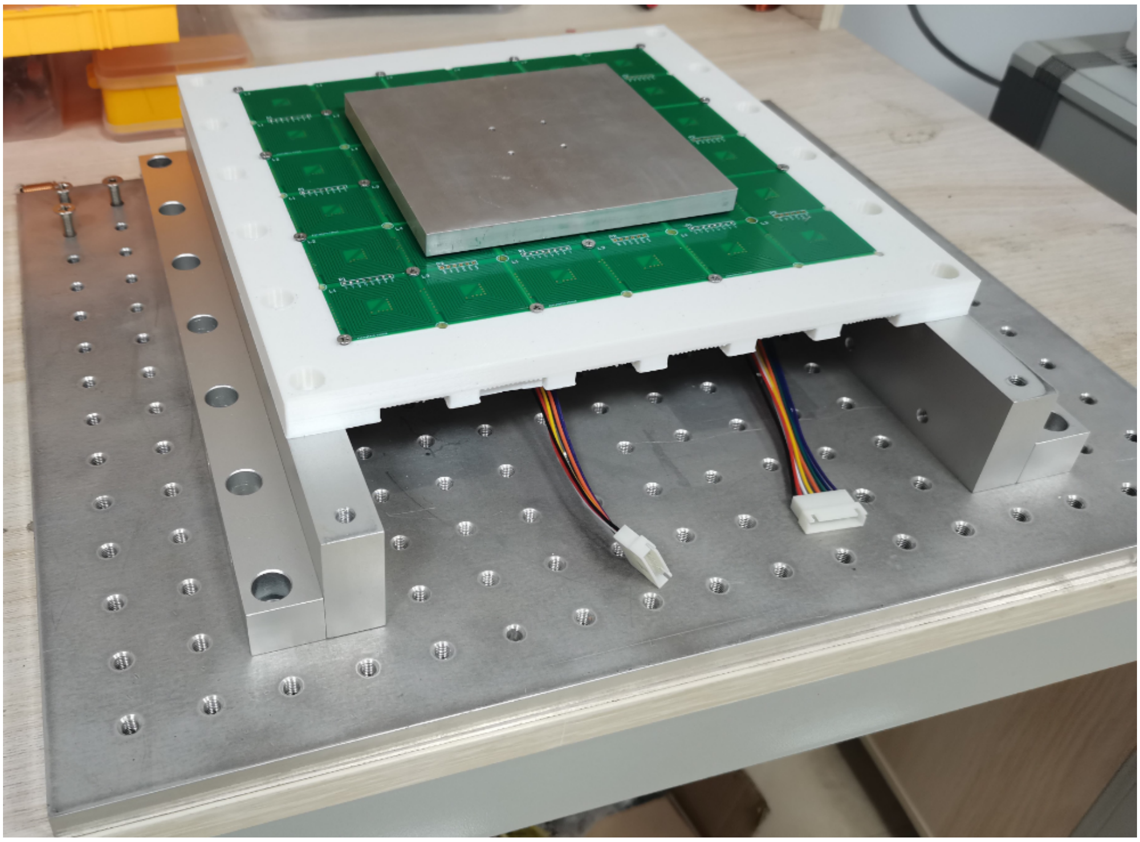


Figure 2.4: Prototype Created in Wuhan University [16]

2.5 Workflow Tools

To ensure reproducibility, the software chosen for this workflow is based on free and open-source software (FOSS) tools that are compatible with each other. To further clarify, we divided our tool selection into two categories.

2.5.1 PCB Design Tools

To effectively design, analyze, and simulate PCB-based planar motors, a comprehensive workflow integrating multiple open-source tools is essential. This workflow spans from PCB design to 3D modeling and simulation, ensuring precision and efficiency at each stage.

1. **KiCad** – An industry-standard open-source Electronic Design Automation (EDA) tool for creating and simulating PCB layouts. It provides essential validation features such as Electrical Rule Check (ERC) and Design Rule Check (DRC) to ensure circuit integrity and manufacturability. KiCad supports the generation of standard fabrication files, ensuring compatibility with manufacturers, while also exporting 3D PCB meshes for integration into mechanical and simulation workflows. Its Python-based architecture enables customization and expandability, making it well-suited for automating design processes and integrating

with external tools for advanced analysis. These features make KiCad an ideal choice for developing PCB-based planar motors, where both precision and adaptability are crucial.

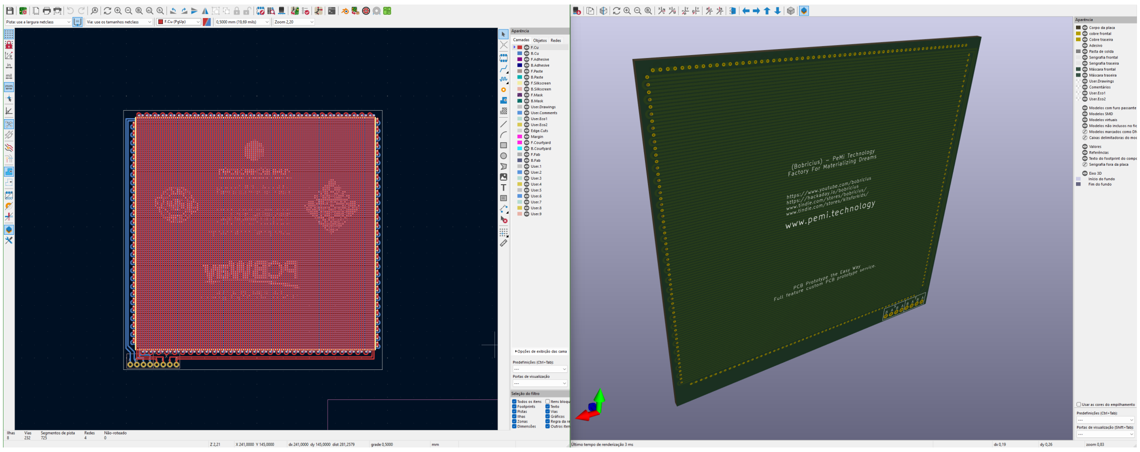


Figure 2.5: KiCAD Interface for PCB Design

2. **FreeCAD** — An open-source parametric CAD software, which is ideal for designing and simulating the mechanical components of the planar motor system. Its parametric modeling features enable rapid prototyping and iterative adjustments, ensuring that mechanical structures are optimized for integration with PCB designs. FreeCAD serves as a standard tool for interacting with 3D objects, ensuring compatibility with various design and simulation workflows.

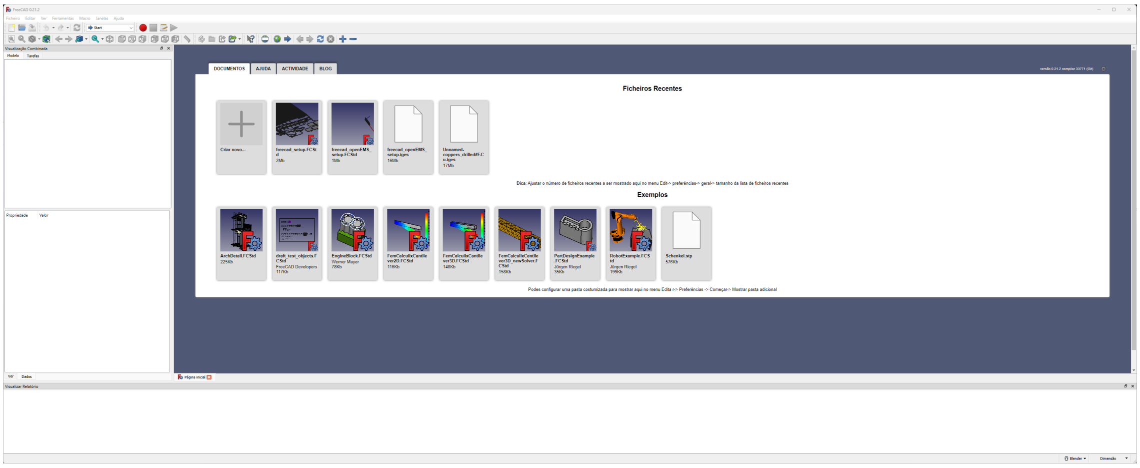


Figure 2.6: FreeCAD Interface

3. **Gmsh** — A powerful open-source tool for mesh generation, primarily used to convert surface meshes into volumetric meshes necessary for Finite Element Analysis (FEA). While Gmsh provides a graphical interface to visualize and evaluate meshing results, its true

strength lies in its functionality as a Python-compatible library. This allows for programmatic control over meshing parameters, enabling automated, script-driven workflows to generate optimized meshes tailored to specific simulation requirements. Its flexibility makes it an essential tool for bridging the gap between geometric modeling and high-precision numerical simulations.

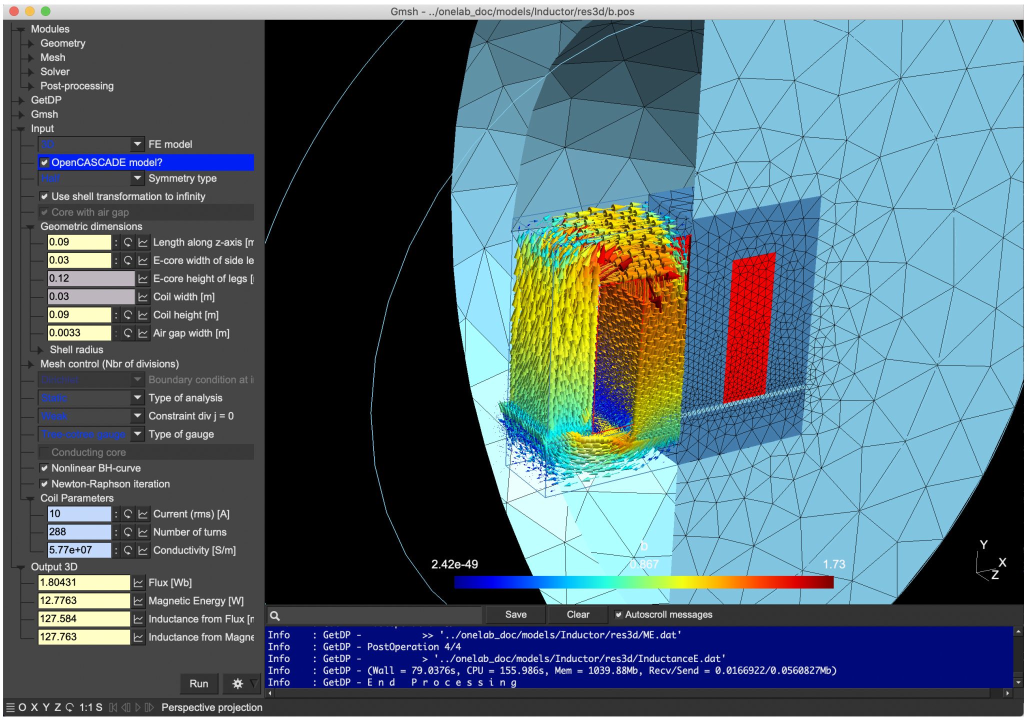


Figure 2.7: GMSH Screenshot [2]

2.5.2 PCB Simulation Tools

To ensure the value and viability of newly developed designs, an evaluation method is needed. This critical step can be approached through various methodologies, offering flexibility in assessing performance and effectiveness. For this purpose, the following tools have been chosen:

1. **ElmerFEM** — An open-source multiphysics simulation software designed for solving complex problems using the Finite Element Method (FEM). It supports a wide range of simulations, including fluid dynamics, structural mechanics, heat transfer, and electromagnetics. However, for this research, we focus specifically on its magnetodynamics capabilities to analyze the electromagnetic behavior of the planar motor. While it provides a graphical interface for setting up and analyzing simulations, its real power lies in its ability to be controlled via scripting, enabling automated workflows and fine-tuned simulations. Its versatility in handling coupled simulations further enhances its applicability to this multidisciplinary project.

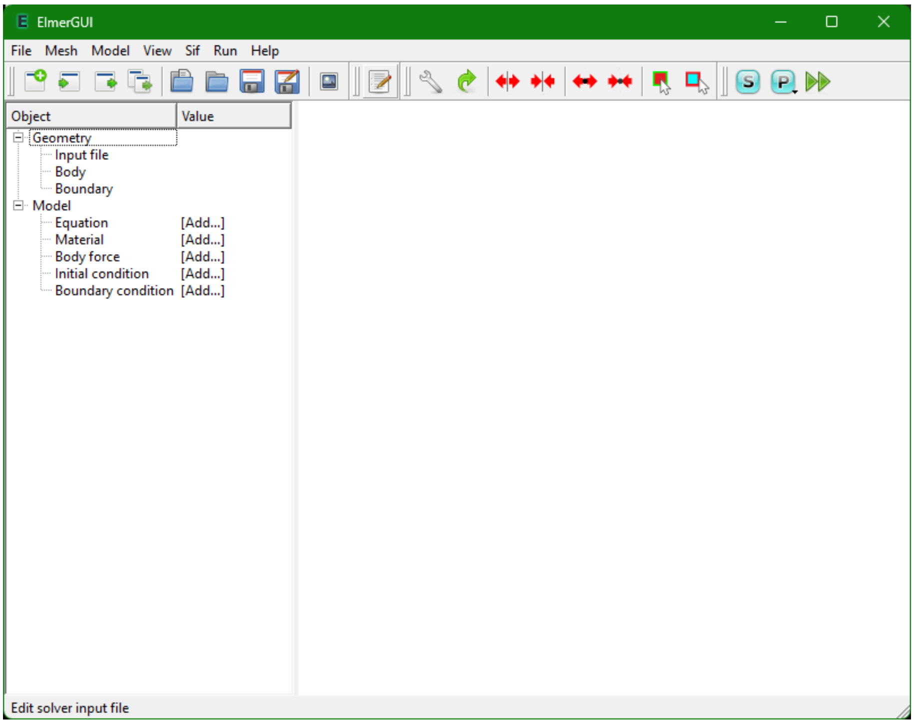


Figure 2.8: Elmer GUI

2. **Paraview** — Following the successful execution of simulations, it is essential to have software capable of interpreting and visualizing the resulting data. ParaView, an open-source data analysis and visualization application, is employed for this purpose. Its advanced capabilities allow for the effective representation of complex simulation results, including electromagnetic fields, thermal distributions, and mechanical stresses. By providing intuitive visualizations and tools for comprehensive analysis, ParaView facilitates a deeper understanding of the system's behavior and supports informed decision-making for design optimization.

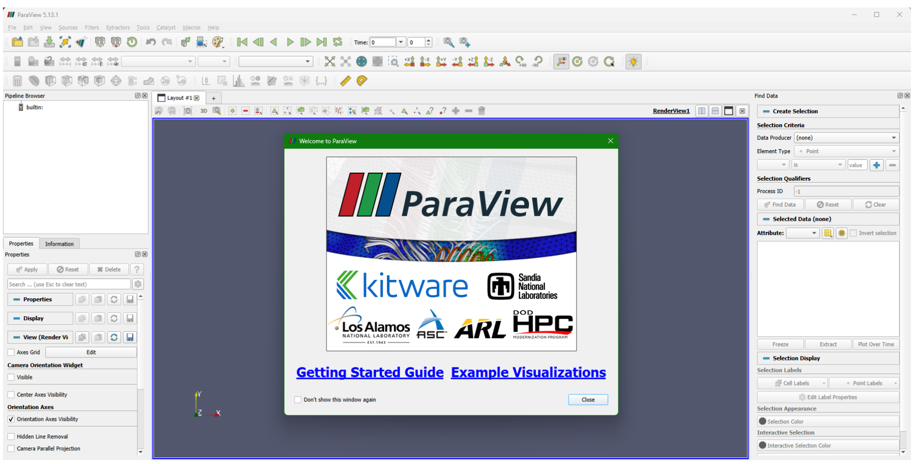


Figure 2.9: ParaView Interface

2.6 Summary

This chapter explored the evolution of electromagnetic motion, from early rotary motors to the development of linear and planar motors. While rotary motors are widely used, the demand for direct linear motion drives the creation of linear motors, which eliminate mechanical transmission losses. Despite their advantages, linear motors remain uncommon in consumer applications but play a significant role in industrial automation and transportation, with growing interest in areas like 3D printing.

Planar motors provide a solution for precise two-dimensional motion control, typically achieved using two perpendicular linear motors. Various configurations exist, each with trade-offs in precision, scalability, and independent actuation. This chapter examined notable implementations, including micro-robotic systems and university research projects that leverage PCB-based actuation to achieve fine control.

Finally, the chapter outlined a workflow for designing and simulating PCB-based planar motors using open-source tools. Software such as KiCad, FreeCAD, ElmerFEM, and ParaView enable precise modeling, simulation, and validation of these motors, ensuring accessibility and reproducibility in research and development.

Chapter 3

MagLev System for Cartesian Robots

3.1 Introduction

This chapter presents the algorithm developed to generate the physical layout of windings for a PCB-based planar motor. The software defines a path for every phase of each motor in the stack, ensuring that the coil arrangement follows the intended design specifications.

By automating the generation of these winding patterns, the tool simplifies the creation of planar motors for maglev-based cartesian robots. The focus is on providing a structured and adaptable method to produce accurate coil layouts, which can be directly integrated into the PCB design process.

3.2 Methodology

The proposed methodology begins by utilizing Python to generate a valid coil based on selected parameters. Since this initial step is parameter-driven, it allows for batch processing, enabling the generation of multiple configurations. These configurations are then converted into PCB copper planes via a custom import script for the KiCad scripting console. From the layered model created in KiCad, a STEP 3D model is generated and imported into FreeCAD.

In FreeCAD, other bodies needed for the simulation, such as the air volume and volumes for the inputs, are added. The complete setup is again passed over to Gmsh to perform tetrahedral meshing. This last step ensures the solving engine can perform computations on the volumes of bodys and not just the surface. Finally, the generated 3D volume meshes and a Standard Interchange Format (SIF), where the simulation setup is defined, are provided to the ElmerFEM solver. Once the simulation is executed without errors, the solver's output can be analyzed in ParaView.

This workflow was chosen to align with processes described in the literature, ensuring consistency with established methodologies [11] [6] [13]. Figure **??** shows the proposed methodology, highlighting in green the scripts developed for this work and in blue the different software tools employed.

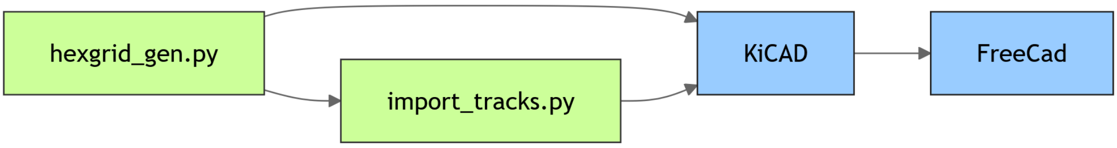


Figure 3.1: Proposed Methodology for design

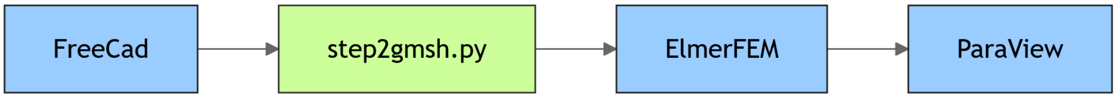


Figure 3.2: Proposed Methodology for simulation

3.3 Coil Generation

In traditional coil winding designs, a standard cartesian grid with orthogonal x and y axes is often employed. However, this approach can impose limitations on the control and efficiency of the coil pattern, particularly in complex electromechanical systems. Drawing inspiration from the advantages of delta robots — where using three axes simultaneously enables faster and more precise movements — this work explores the potential benefits of adopting a hexagonal grid for coil generation.

The hexagonal grid offers a natural alignment with a tri-axis control strategy, as its geometry inherently supports three primary directions, each 120º apart. The hexagonal shape not only optimizes space utilization but also ensures sufficient routing space and connectivity for each phase, facilitating a well-organized and compact coil layout [12].

By leveraging this geometric approach, the resulting design intends to replicate the enhanced control and improved performance from the delta configuration, as well as providing seamless integration into the final electromechanical system. This hexagonal grid representation also facilitates the design process, as the required 120º rotation between each layer can be achieved by simply reordering the grid's axes.

The coil generation script begins by determining the overall size of the hexagonal grid based on the specified number of stages, phases per stage, and coil width. Using these parameters, it creates an empty grid that later serves as the foundation for the coil design. The grid is structured using hexagonal coordinates, where each cell is represented by a tuple (x,y,z,h) with x,y,z defining the cell's position within the hexagonal framework, and h indicating the specific layer. Additionally, the hexagonal grid data structure stores a color value for each cell, which is used to separate each winding and to prevent the crossing of paths.

This method of grid construction naturally accommodates the multi-layered nature of the winding design. To ensure the minimum amount of free space between the coils, the width of the coil is set to a default minimum of 4 cells, which ensures a spacing of one cell between adjacent coils.

As an example, Figures 3.3 to 3.7 pertain to a design that implements 3 linear Permanent Magnet Synchronous Motors (PMSMs), with each motor assigned to a distinct layer of the grid. Each linear PMSM operates with 3 phases, resulting in a total of 9 phases distributed across the 3 layers. This is the selected configuration for the remaining design work and simulation. However, the number of stages remains an open parameter, subject to further optimization and refinement.

Once the empty grid is established, the script initiates the creation of the winding pattern. The first segments to be laid down are the longest paths of the coil which are parallel to one of the axis. This step is critical as it defines the initial repeating structure upon which the rest of the pattern will be built.

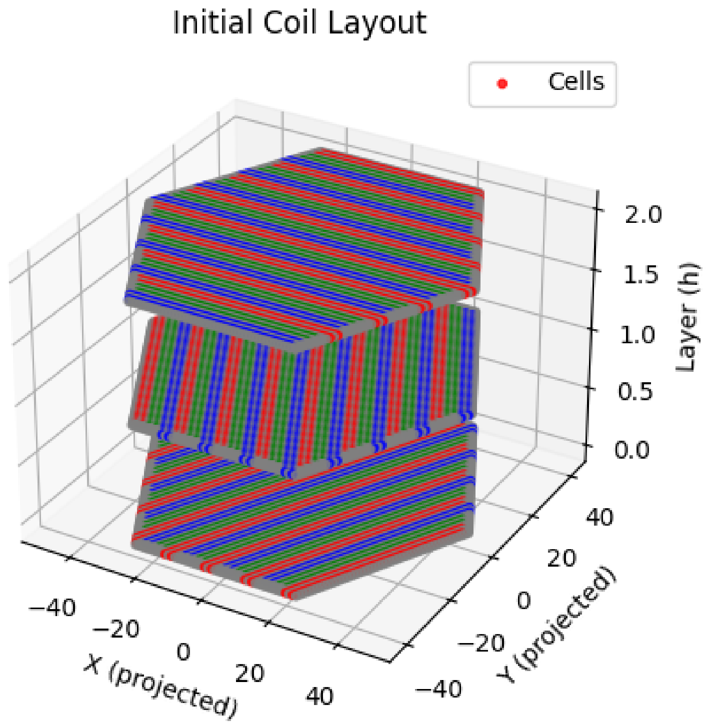


Figure 3.3: Initial Coil Layout for 3 Layers, 3 Phases, 8 Stages

After establishing this primary pattern, the script proceeds to connect phase A and phase C (represented in Figures 3.3 to 3.5 with the colors red and blue) along the 4 outer edges of the grid which are not parallel with the coil within the same layer, with phase A and phase C taking opposing sides. This connection is strategically executed to ensure that the electrical pathways are not directly connected at the edge.

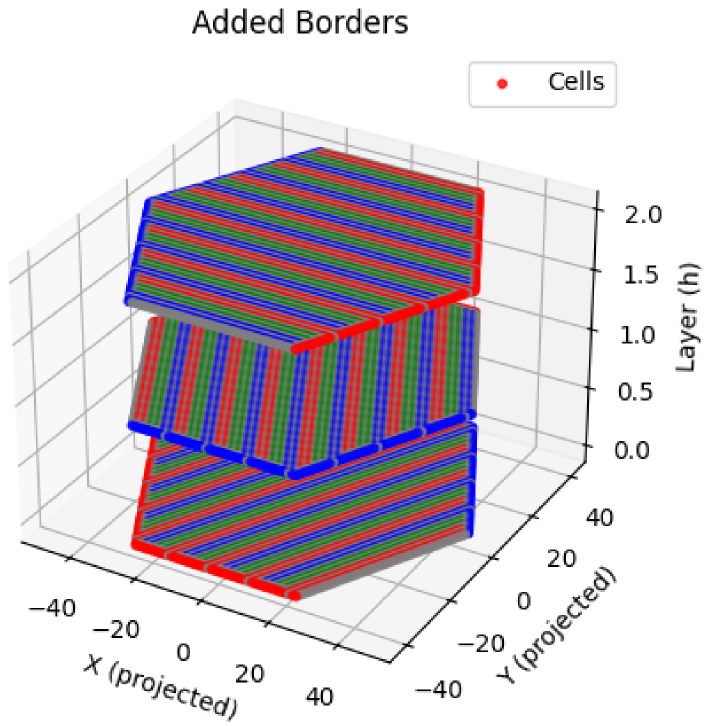


Figure 3.4: Layout After Adding Borders

Following the establishment of the primary winding sections, the script closes the loops by connecting the ends of each segment, effectively completing the circuit for each phase within its designated layer. During this process, it also detects critical points and tags the corresponding cells with a color identifier. This tagging creates a visual and logical grouping that facilitates the subsequent matching process and ensures robust inter-layer connectivity.

These tagged cells indicate locations where a layer change is required to complete the path for a certain phase. In PCB design, such points are known as vias, and from this point forward, this is the term we will use to refer to these cells. To ensure that vias are correctly paired in sequence, two distinct colors are used during via selection, ensuring that only pairs of vias with the same color are connected, maintaining proper electrical continuity.

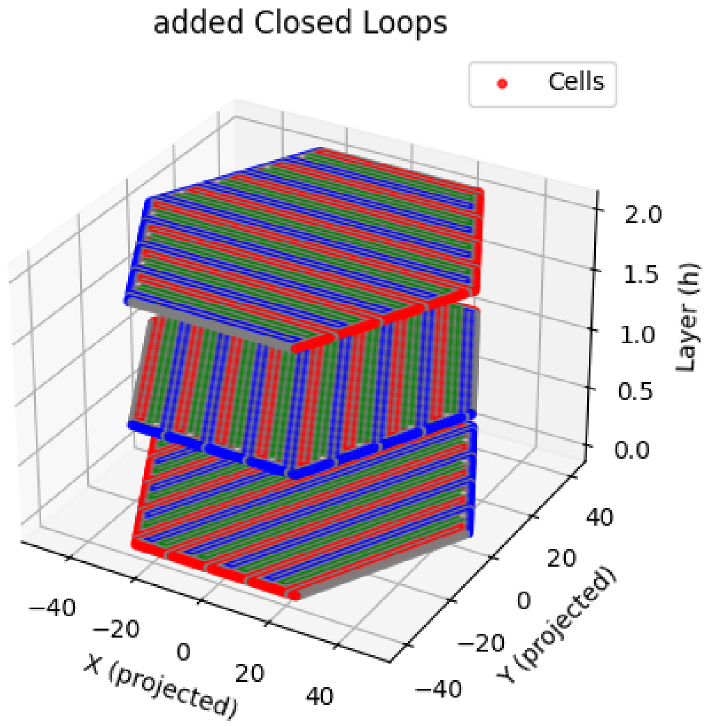


Figure 3.5: Layout after Closing the Loop and Detecting Vias

From here, the script assigns a unique color to each of the 9 phases to enhance visual clarity and distinguish electrical nets. This aids in design verification and troubleshooting by making it easier to identify and separate different phases. Following this, the script adds an additional empty layer to the grid, dedicated to routing the start and end points of each phase to the driver circuit. This layer (described in Figure 3.6 as Back Plane) acts as an routing layer, ensuring a structured connection between each phase's path within the grid and the external driver circuit.

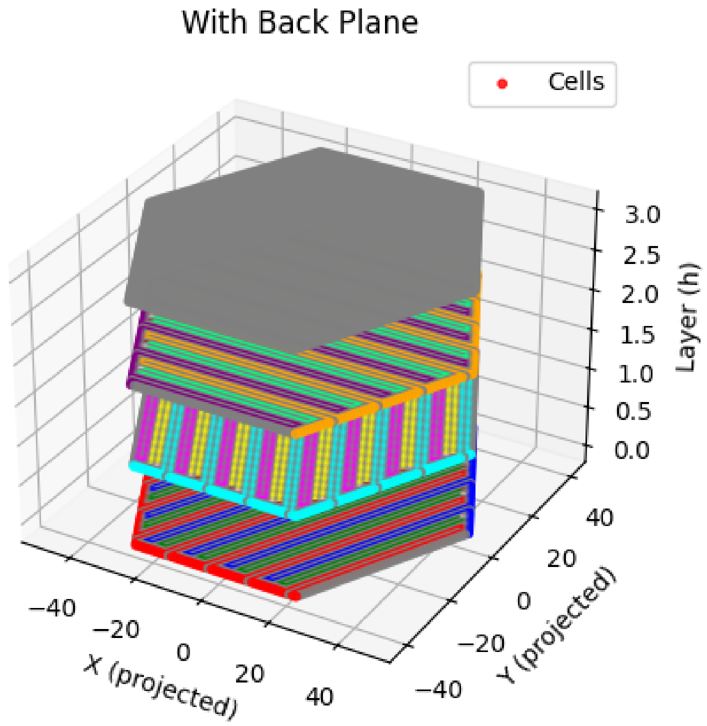


Figure 3.6: Layout After Adding the Back Plane

After completing these preparatory steps, the script employs a modified Best-First Search (BFS) algorithm to find the shortest route between pairs of tagged vias for each colored phase. This process optimizes the interconnections between different layers, minimizing the overall length of the path and the number of changes in direction.

With all paths between vias successfully constructed through both direct connections as well as optimized via routing, the script moves to the final step. This involves connecting the start and end of each path to the back plane, thereby creating a continuous and complete electrical network that is ready for PCB layout integration.

This step encompasses the creation of a path for every phase by searching for each color's longest continuous path possible. These paths are later used to determine the starting and ending cell, which are assigned as starting points for the BFS algorithm, this time with a single layer as the target.

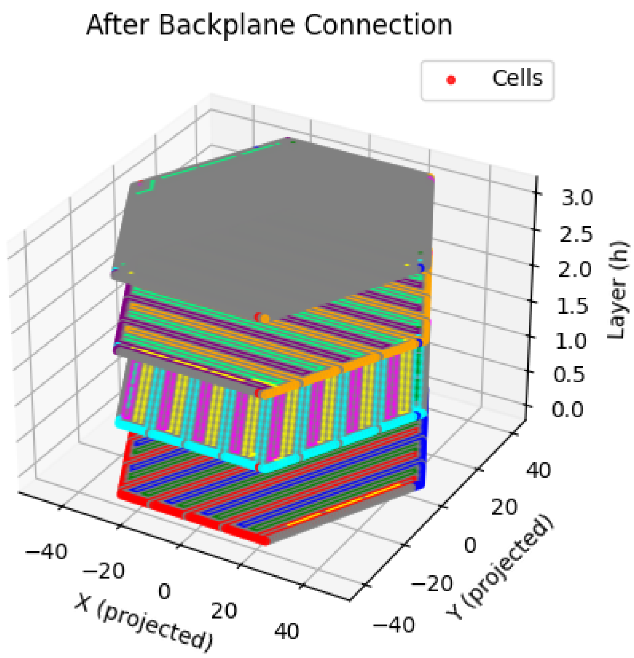


Figure 3.7: Layout After Connecting to the Back Plane

Overall, the script seamlessly integrates multiple steps, from determining grid size and laying down initial winding patterns for each phase, to optimizing via routes and finalizing with the back plane integration, all while making sure that each step takes advantage of the hexagonal geometry.

Additionally, the script produces a design that efficiently utilizes the free space between coils and the rotation between each layer to nest all connections within their respective boundaries. This structured approach ensures that future expansions can seamlessly tile these stacks into a larger hexagonal grid, facilitating scalability and modularity in the design.

Finally, since the abstract representation of the paths remains unchanged regardless of the export format, and due to the fact that generation time increases for larger values, the script includes a caching mechanism. This mechanism stores the precomputed path data, allowing exporters to retrieve and use it without requiring costly recalculations, which significantly reduces the time needed to run different exporters. This method further ensures that the transition from computational design to practical PCB fabrication is smooth and well-coordinated.

3.4 PCB Design

To transform the 3D hexagonal grid paths into a manufacturable design, physical dimensions must be assigned. This process is carried out using KiCad, where manufacturability constraints are enforced through the Design Rule Check (DRC). Three key factors are considered in this process: spacing between objects, via size, and trace width.

The spacing between objects is defined as a manufacturing parameter, commonly set to a default minimum of 0.2 mm, ensuring that all elements remain at least 0.2 mm apart. Since vias are positioned at the center of each hexagon, the minimum hexagon radius must be at least half of the default spacing plus the via radius to maintain proper clearance between vias.

Trace width is constrained by the hexagonal grid, with the most restrictive case occurring along the vertical direction, as the distance between hexagon centers is shorter in one direction than the other. The maximum allowable trace width for a given via size is determined by taking into account this worst case scenario.

Via size is a crucial dimension in this design. To establish an appropriate via diameter, standards from IPC-2221A [3] were consulted, which specify via sizes based on current capacity. The BD6342FV-E2 was chosen as the driver, a three-phase full-wave fan motor driver commonly found in laptop motherboards and other commercial appliances. It was selected due to its fully integrated design, small form factor, and reasonable cost, with the capability to supply 1A of current per phase [1]. Based on IPC-2221A standards, a via size of 0.6 mm is required for 1A current capacity, setting a fixed constraint on via dimensions within the PCB layout.

3.4.1 Schematic

Before importing the generated paths into KiCad, a schematic that includes the labeled nets intended for the PCB layout must be created. To achieve this, a schematic has been designed based on the application reference from the driver's datasheet [1]. This circuit was then replicated three times, once for each layer, ensuring a consistent and well-structured foundation for the PCB design.

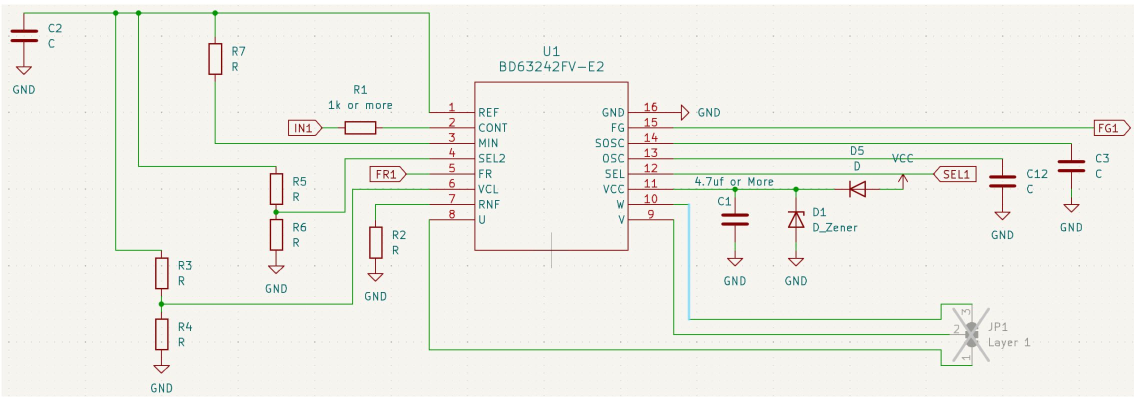


Figure 3.8: Single Driver Circuit

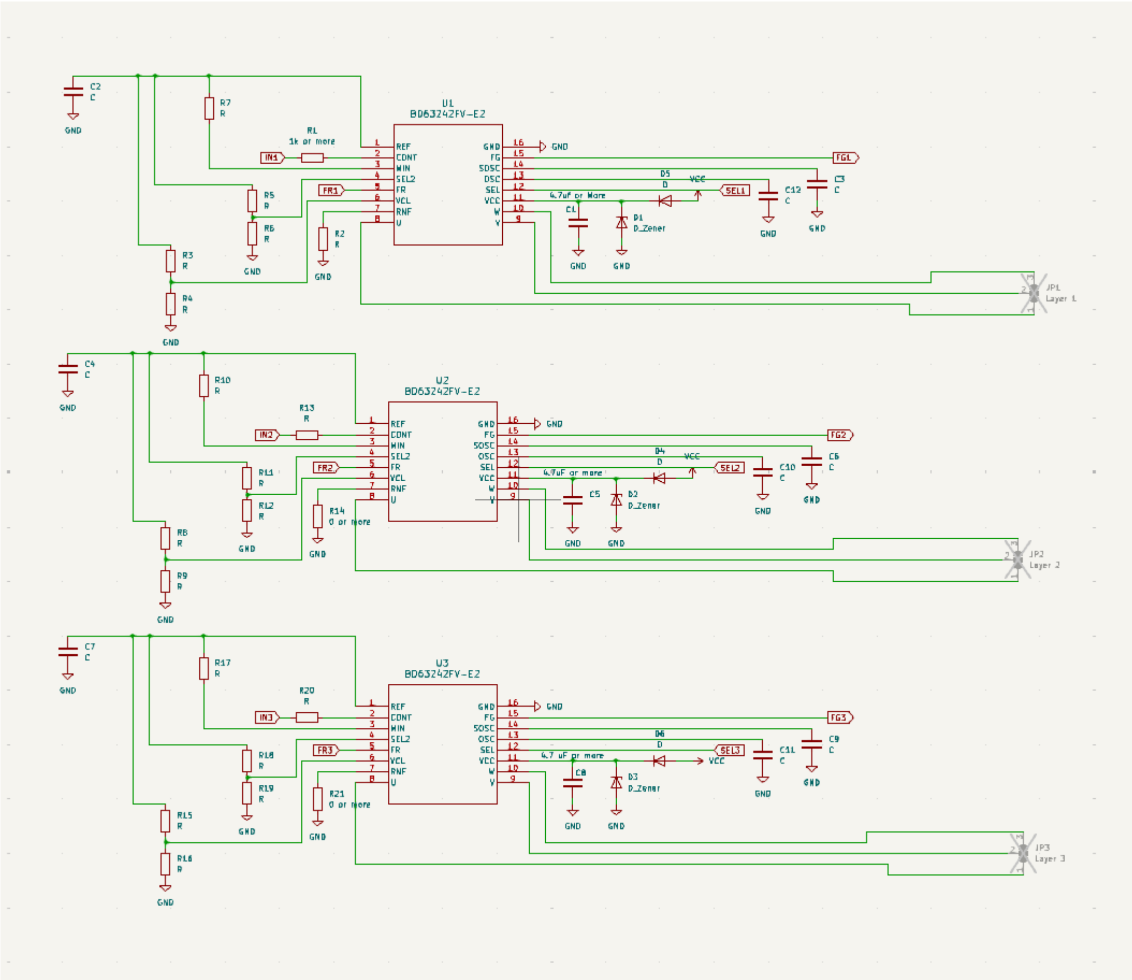


Figure 3.9: Full System Circuit

3.4.2 Layout

After the generation script runs, it outputs a JSON file containing all the line segments in the path, sorted by layer and electrical net. This process works by converting the (x, y, z) indices of the hexagonal grid cells into standard (x, y) coordinates, ensuring compatibility with PCB design tools. This structured format allows for easy integration into KiCad, where the paths can be imported and mapped to the corresponding schematic nets and layers.

In the same iteration where the script processes the path, it also checks if the direction is vertical, meaning along the h-axis (layer transitions). If a vertical movement is detected, the script converts the corresponding (x, y, z, h) coordinates into standard (x, y) coordinates and places a via at that location.

The via's starting layer is defined by the layer of the first cell where movement along the h-axis occurs, while the ending layer is determined by the layer of the last cell in the final consecutive movement along the h-axis. This ensures that vias correctly span only the necessary layers, maintaining accurate electrical connectivity throughout the multilayer PCB. At the same time, since the program generates traces over inner vias, through-hole vias are not viable, leaving blind and/or buried, or stacked micro vias as the only options for proper layer transitions

After importing the traces into KiCad and verifying that no electrical errors were detected, the remainder of the circuit was laid out. Since this final step of designing the driver circuit is manual and time-consuming, only one design was completed. To further simplify the process, the number of stages was set to eight, providing ample space for routing. This step is also crucial as it ensures that the device can be wired and connected using only a single layer for the driver circuit, improving manufacturability and reducing complexity.

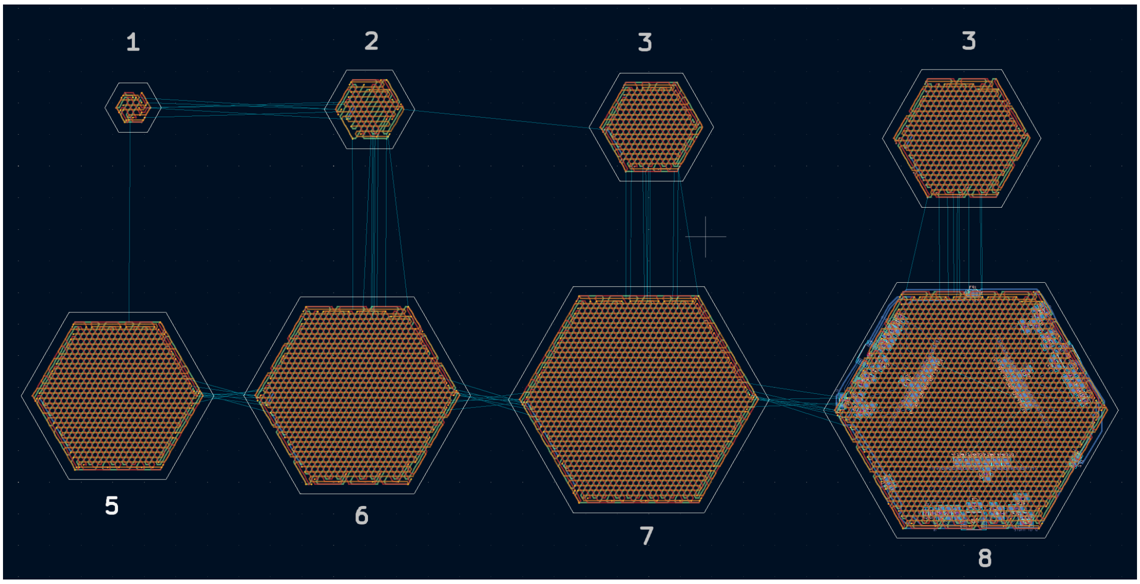


Figure 3.10: Designed Layouts Ranging from 1 to 8 Stages

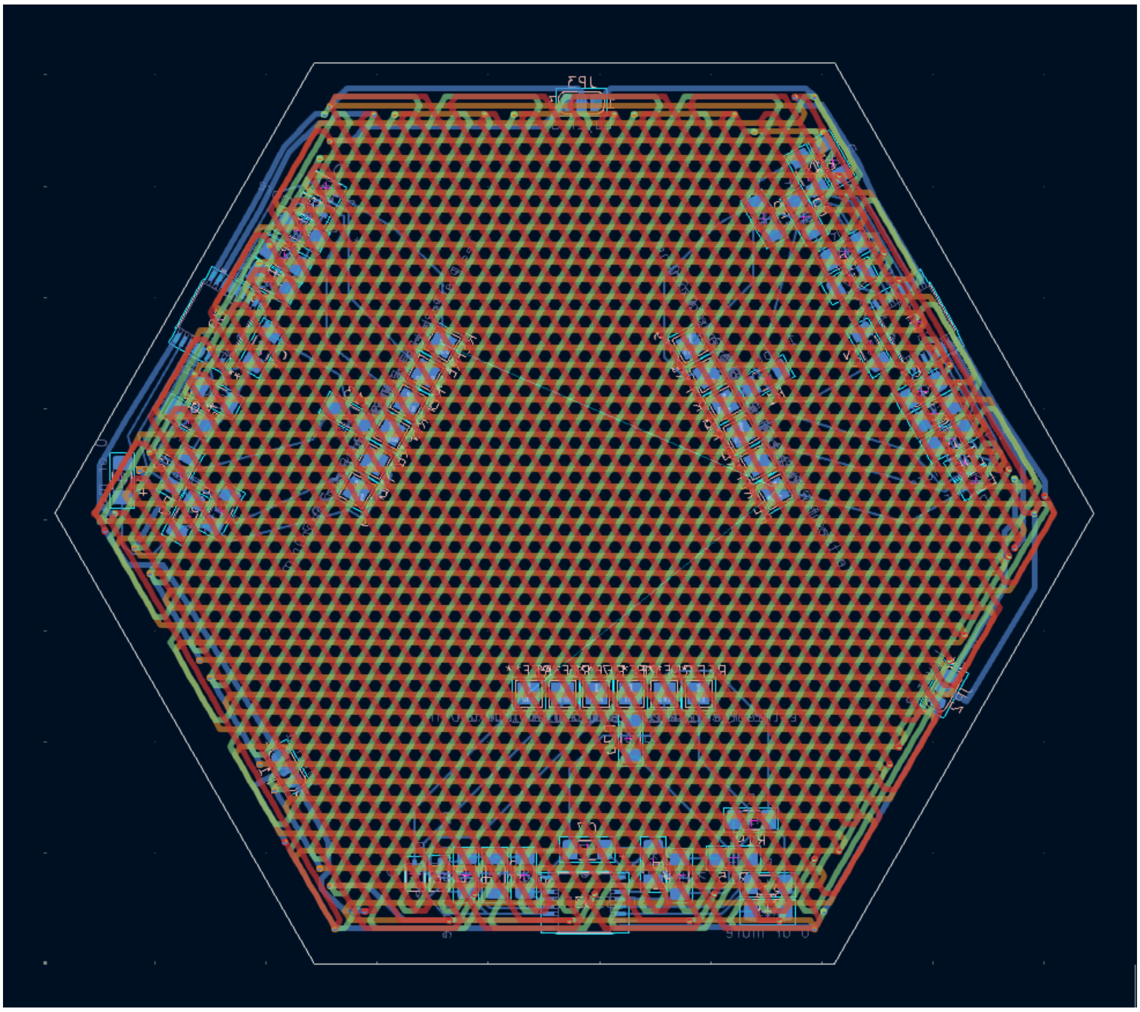


Figure 3.11: Detailed View of the Finished Layout

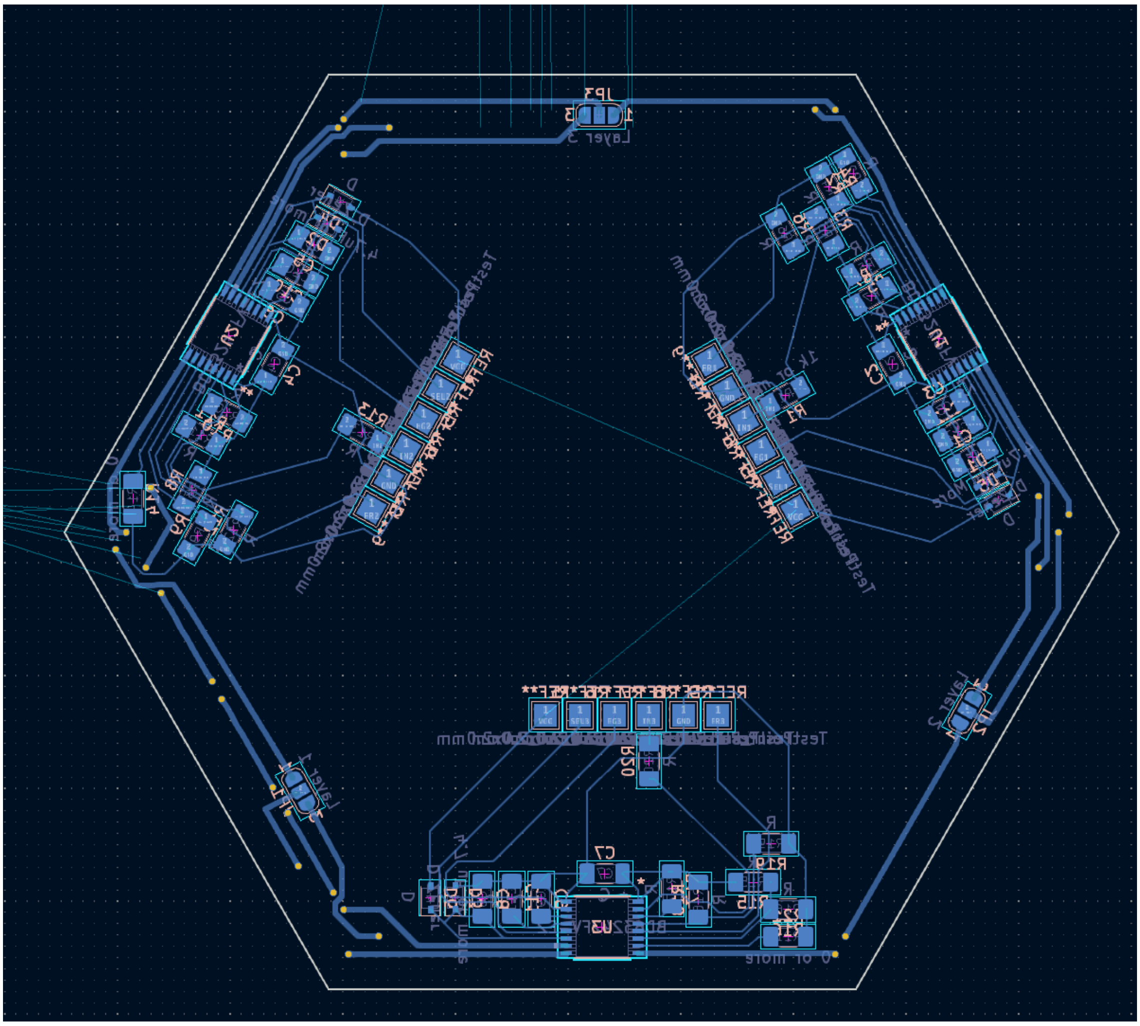


Figure 3.12: Detailed View of the Finished Layout's Back Plane

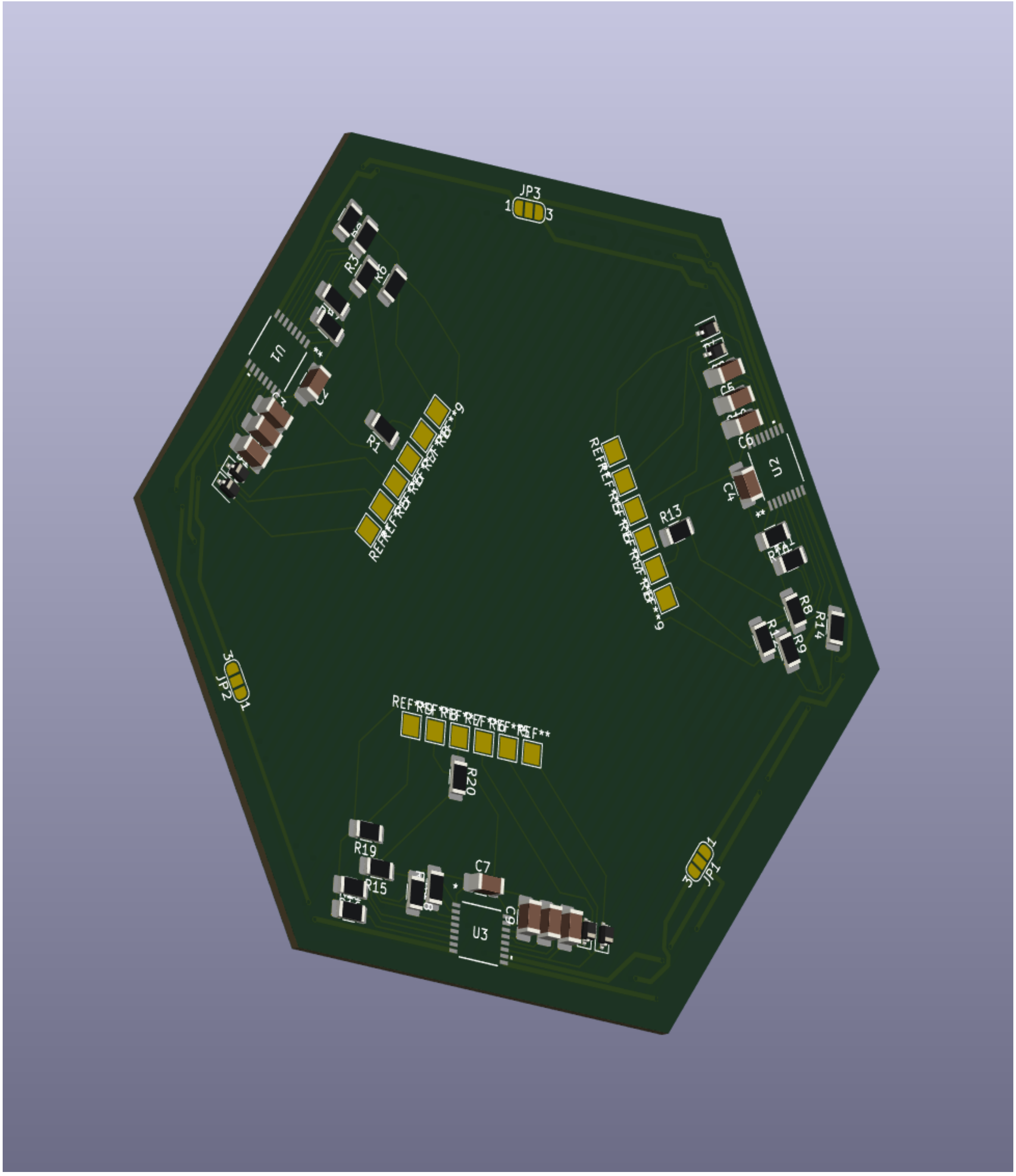


Figure 3.13: KiCAD 3D Render of the Finished Layout

Finally, utilizing KiCad's export features, a STEP file of the circuit is generated, with the FR4 body and traces exported as separate entities. Additionally, Gerber files were created for manufacturing, ensuring compatibility with PCB fabrication processes. The STEP file is, subsequently, used to create a mesh for simulation, providing an accurate representation of the PCB's physical and electrical properties.

3.5 Summary

This chapter detailed the methodology for generating and implementing the physical layout of windings for a PCB-based planar motor, designed for use in maglev cartesian robots. The approach leverages a hexagonal grid structure to optimize space utilization, ensure proper routing, and facilitate a scalable and modular design.

The coil generation process includes the automated creation of winding patterns, the identification of critical points for inter-layer connectivity (vias), as well as the assignment of electrical nets. The hexagonal grid aligns with a tri-axis control strategy, providing an efficient and structured way to generate complex coil layouts while minimizing electrical interference and maximizing available space.

The PCB design process defines physical dimensions and enforces manufacturability constraints using KiCad's Design Rule Check (DRC). Key factors such as trace width, via size, and spacing are considered to ensure compliance with industry standards. The integration of the BD6342FV-E2 driver optimizes the layout to accommodate current requirements while minimizing complexity.

The circuit layout and routing strategy incorporate a caching mechanism to store the abstract representation of the generated paths. This caching feature reduces processing time when exporting to different formats. Additionally, the system enforces the use of blind, buried, or stacked micro vias, as through-hole vias would obstruct trace routing.

Finally, the chapter mentioned the employment of a KiCad-based PCB export, where a STEP file is generated, including the FR4 body and traces as separate entities. This model prepares the design for 3D simulation and analysis, ensuring manufacturability and electrical integrity.

This methodology establishes a structured framework for generating and optimizing PCB-based planar motors, paving the way for future improvements in scalability, performance, and integration with more complex electromechanical systems.

Chapter 4

Simulations Setup And Results

4.1 Introduction

Simulation is essential in design, enabling early-stage optimization and problem identification before physical prototyping. By modeling electromagnetic fields and mechanical interactions, it helps estimate force generation, analyze field distribution, and optimize coil layouts.

Despite its advantages, simulation has limitations. Accuracy depends on model fidelity, mesh resolution, and solver capabilities, with trade-offs between precision and computational efficiency. Approximations and numerical artifacts can introduce errors, making real-world validation necessary to confirm predictions.

While simulations may not provide absolute performance metrics, they offer valuable qualitative insights into system behavior. By carefully structuring models and interpreting results, designers can refine their designs and better understand operational characteristics. This chapter explores the methodologies, limitations, and insights gained through the simulation of the developed design.

4.2 Simulation Methodology

The methodology used in this study relies on Gmsh for meshing, which allows for flexible domain discretization by defining multiple bodies to serve as input or excitation regions for the simulation. The meshing process is crucial in balancing accuracy with computational efficiency, ensuring that the critical regions of the planar motor, such as the coils and magnetic interfaces, are properly resolved without unnecessary computational overhead.

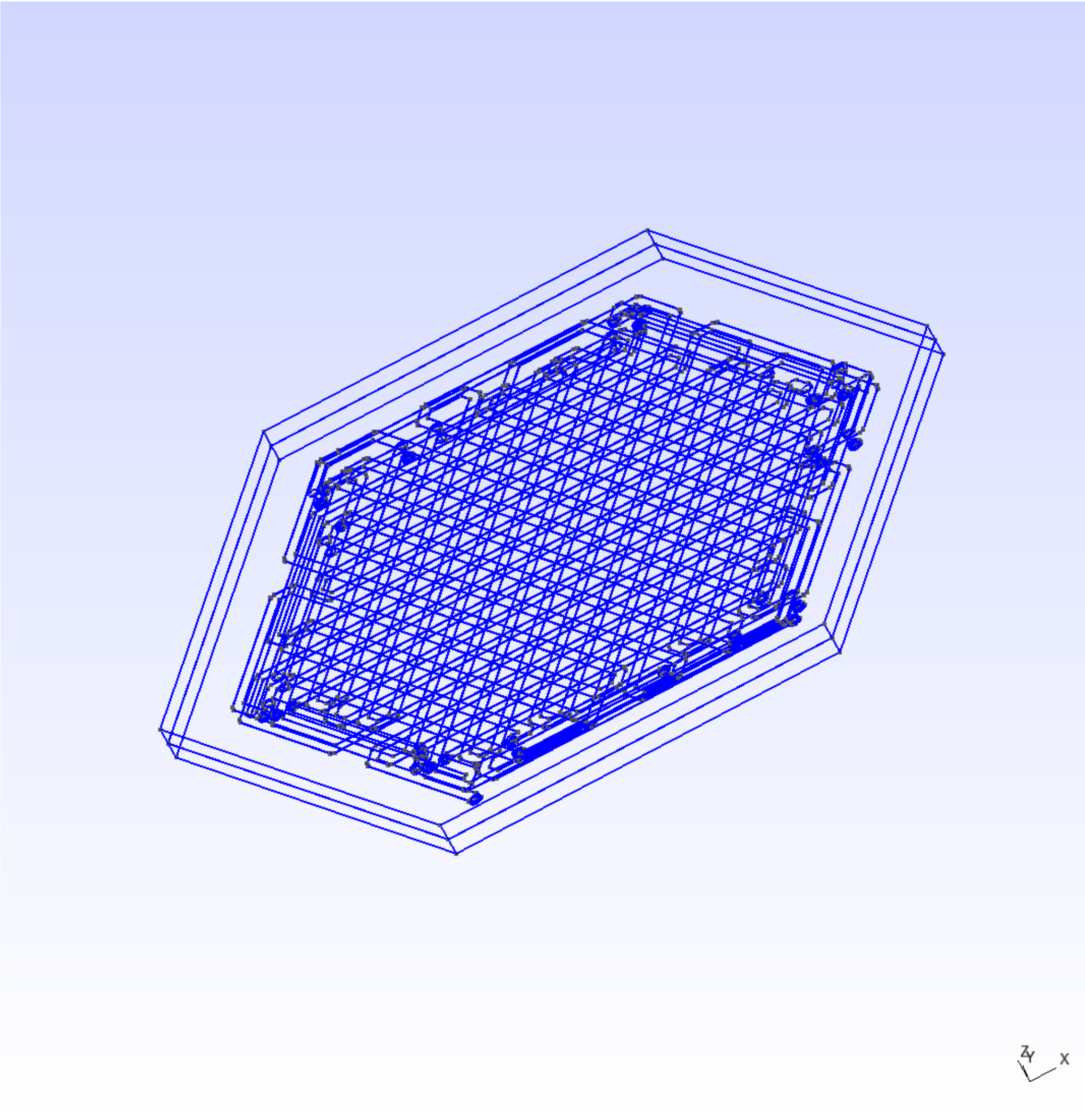


Figure 4.1: Gmsh After Importing the STEP File from KiCad

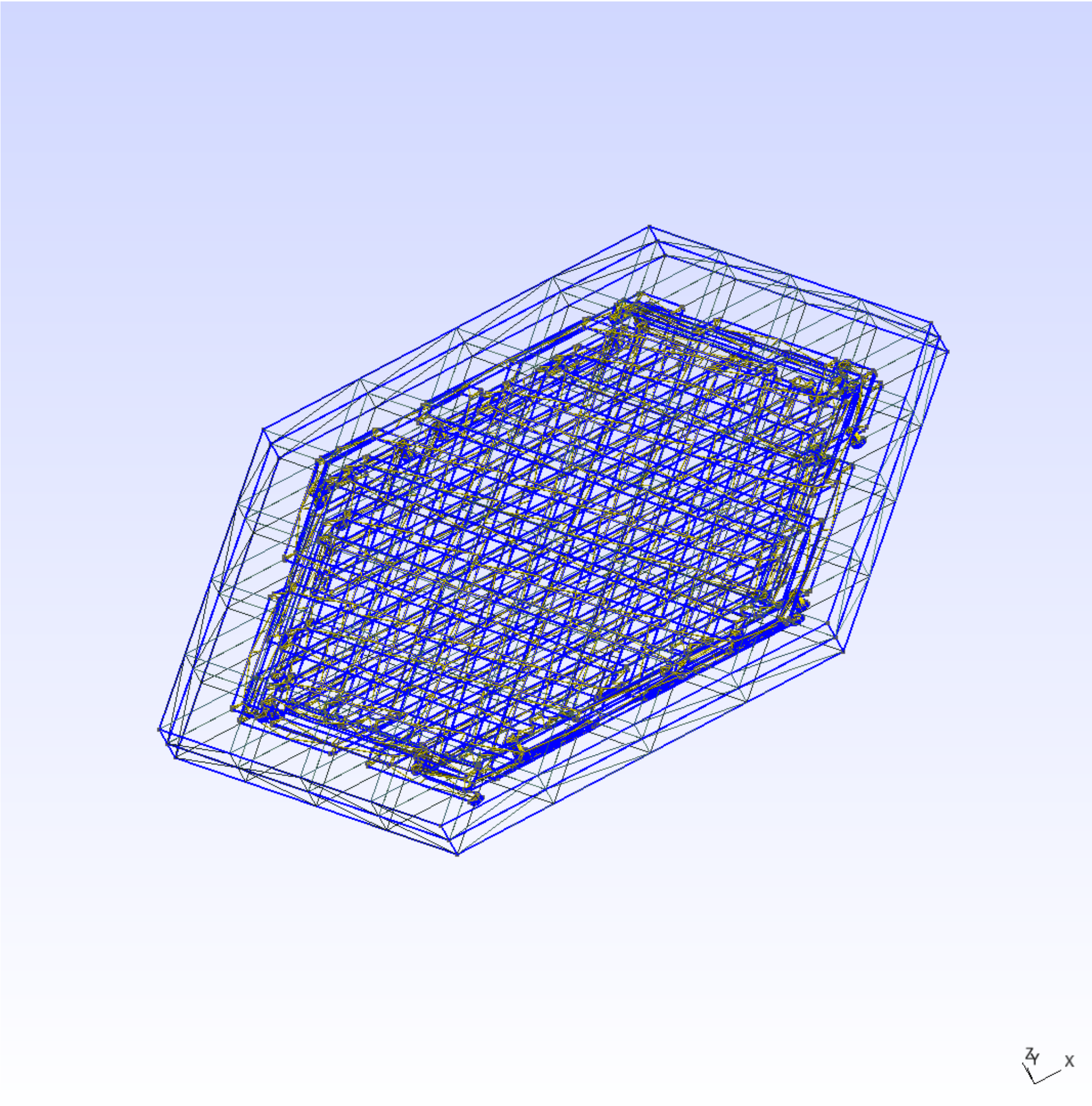


Figure 4.2: Gmsh After 2D Meshing (Surface Level)

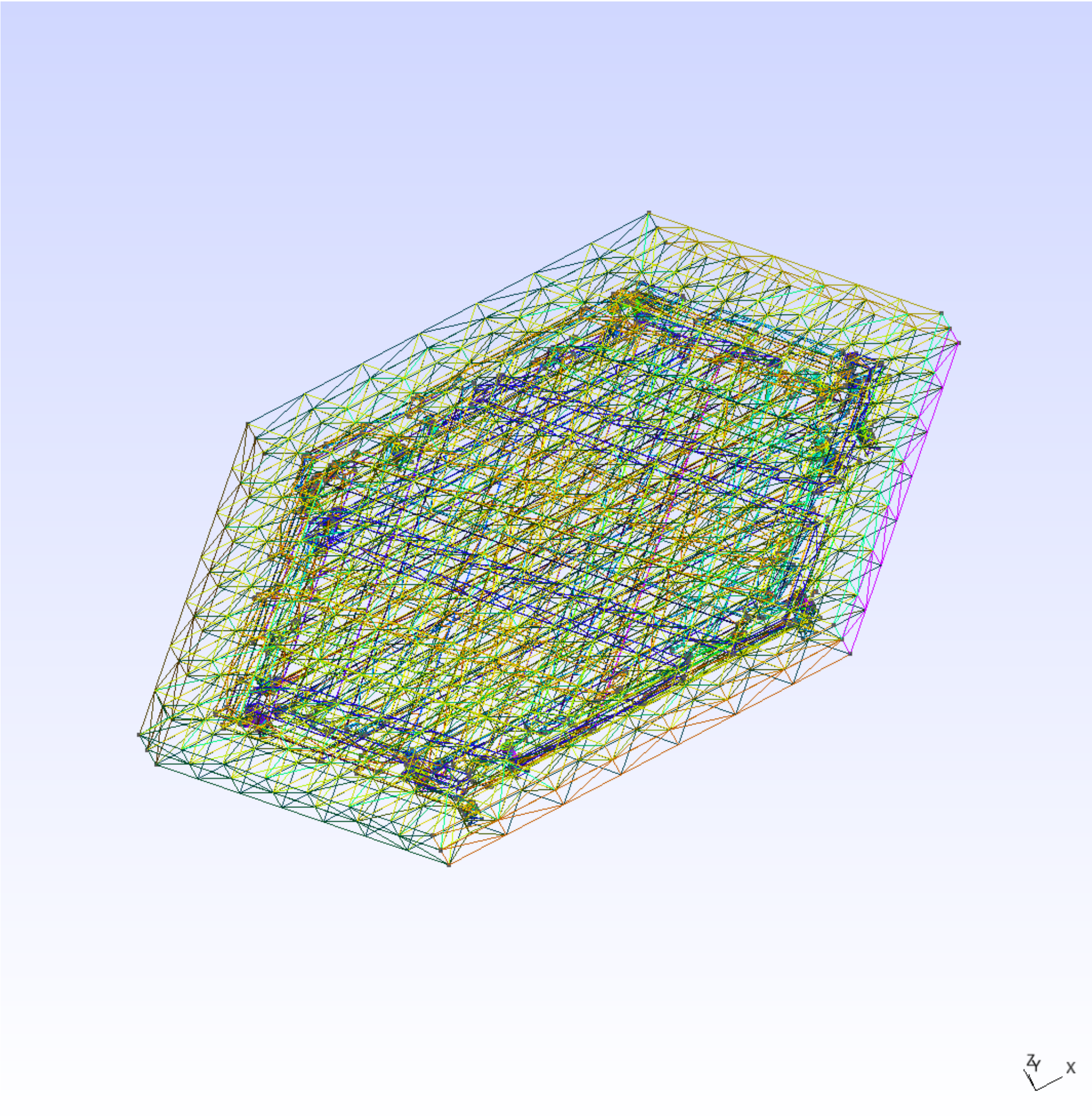


Figure 4.3: Gmsh After 3D Meshing (Volume Level)

To properly define the excitation of the system, the chosen motor driver is first analyzed to determine the most suitable method for injecting the driving signal into the simulation model and two main approaches are considered.

The first method consists of injecting current into a designated body using the body force method in Elmer. This approach directly models the current flow within the conductor volume, allowing the solver to calculate the resulting electromagnetic field distribution and interactions. In this configuration, applying a body force is equivalent to injecting a sinusoidal current into the trace, simulating the alternating current flow imposed by the driver on the physical system.

The second method consists of fixing a boundary condition on the surface of the input region, enforcing a predetermined electrical potential corresponding to the signal generated by the driver. To accurately replicate the driver's excitation, the same Pulse-Width Modulation (PWM) signal used by the driver must be generated. The PWM is configured so that, after filtering, it reconstructs a full-wave sinusoidal signal. By applying this potential constraint, the electric field distribution within the system is controlled, effectively simulating the real-world behavior of an externally driven coil.

While both methods offer advantages, the first method of body force injection has been chosen, due to time constraints and simplicity. The injection of current directly into the trace can be reproduced using the body force method, which provides a straightforward way to model the excitation without requiring additional signal processing in order to match the driver's signal. This approach enables faster implementation and easier interpretation of results while still offering meaningful insights into the electromagnetic behavior of the PCB planar motor.

As this is an early-stage simulation, the focus remains solely on measuring the magnetic field distribution above the PCB rather than modeling the full dynamic behavior of the motor. To facilitate this, a Dirichlet boundary condition is applied to the outer air region to mimic an infinite free space. This prevents the simulation domain from artificially constraining the magnetic field and allows the solver to approximate an open environment where the field naturally extends beyond the PCB surface. By imposing this boundary condition, the simulation aims to produce a more realistic representation of the magnetic field behavior in practical operation.

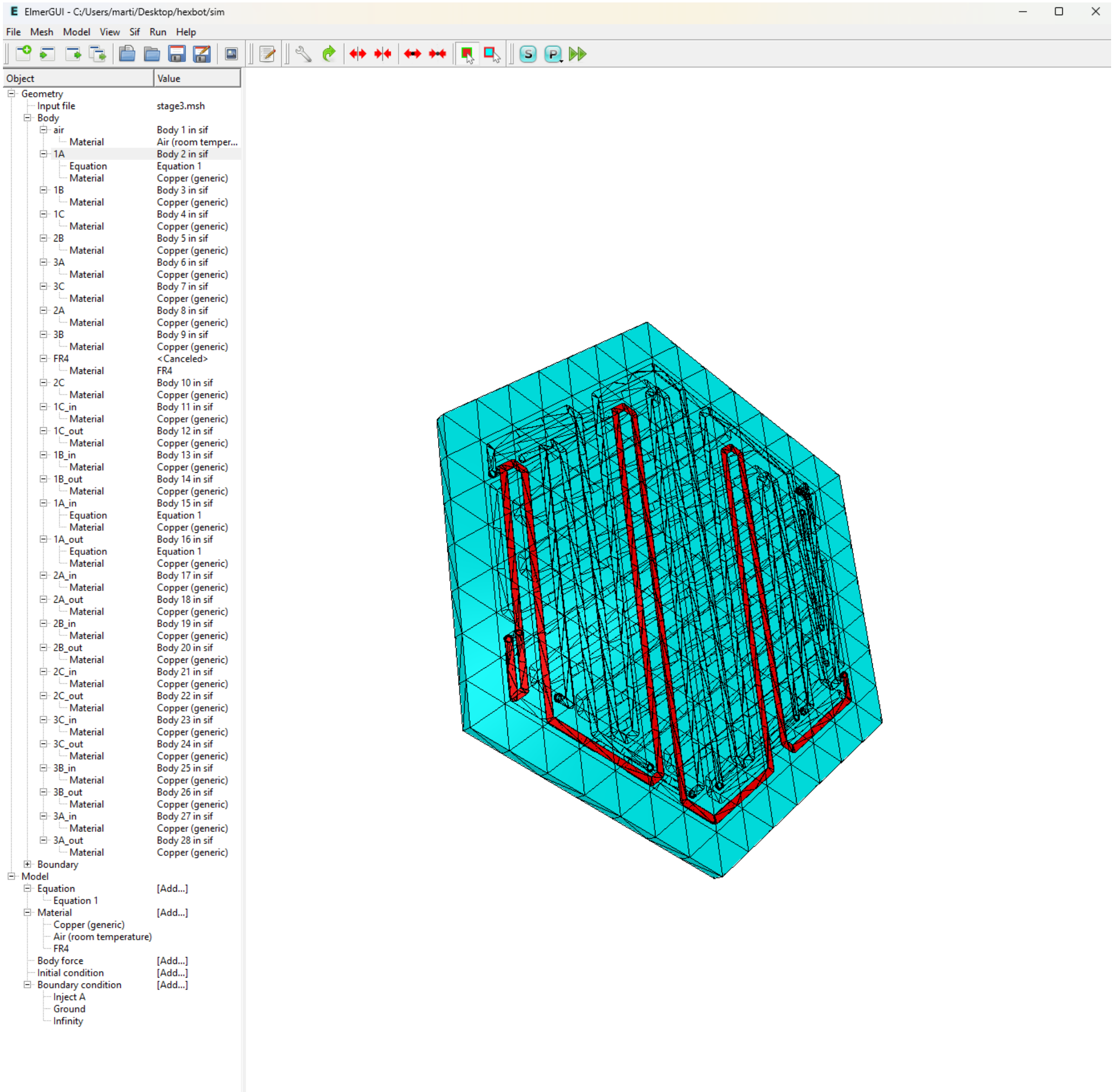


Figure 4.4: Elmer FEM Simulation Setup

By structuring the simulation setup in this manner, this study takes advantage of Gmsh as a meshing tool and Elmer as the simulation solver, with both providing a robust framework for modeling the electromagnetic behavior of the PCB planar motor while keeping computational costs manageable.

4.3 Simulation Results

The simulation is setup to only provide qualitative insights regarding the operation of the PCB planar motor. However, due to the limited number of simulations conducted and the inherent assumptions in the modeling approach, the current simulation setup is not able to provide direct measurements of real-world performance.

Currently, due to time constraints as well as the complexity of the elements that need to be simulated, viable results have not yet been produced. This issue is still being worked on and, with further refinement, the goal is to still simulate the electromagnetic field distributions, to later append the generated results into this document. However, this simulation will only provide approximations that help visualize and understand how the device operates, rather than precise predictions of actual performance.

Several factors can contribute to this discrepancy between simulation results and physical reality. The idealized nature of the model, the simplifications in material properties, and the constraints of the meshing process all introduce uncertainties. Additionally, the real-world system may be affected by manufacturing tolerances, parasitic effects, and dynamic interactions that are not fully captured in the simulation environment.

Ultimately, while simulation results should not be used as absolute performance indicators, they serve as an important tool for developing intuition about the system's operation. Further refinements, including experimental comparisons and more advanced simulation techniques, are necessary to bridge the gap between theoretical modeling and practical implementation.

4.4 Summary

This chapter provided an overview of the meshing process used for simulation and outlines the setup for preliminary testing. It describes the methodology behind generating the mesh and defining the excitation conditions necessary for simulating the PCB planar motor.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Regarding the first goal of creating a framework for designing PCB planar motors, this objective has been successfully achieved. The developed framework provides a structured approach to designing these devices, integrating various open-source tools to facilitate the process. Furthermore, releasing the tools and workflows built for this dissertation with an open source license, allows for greater accessibility, collaboration, and future improvements by the research community, enabling others to replicate, modify, and extend the work for various applications in PCB-based planar motors and electromechanical systems.

Additionally, to verify the manufacturability of the proposed design, Gerber files generated in KiCAD were sent to multiple PCB manufacturers for pricing estimates. Given the constraints imposed by blind and buried vias, the manufacturing cost is higher than a comparable design with through-hole vias. However, at approximately 400€ for a batch of five boards, the cost remains relatively low for a device of this complexity. Moreover, this price is expected to decrease when producing an array of devices, as bulk manufacturing typically reduces per-unit costs.

For the second goal of simulating these devices, the complexity of the resulting mesh led to a significant computational load. While simulations can provide useful insights, the current setup suggests that improvements in meshing techniques, solver configurations, or alternative simulation approaches may be necessary to generate viable results.

5.2 Future Work

One possible immediate improvement involves transitioning to a 6-layer PCB. In the current design, the middle layer of the PCB fiber is thicker for structural stiffness, creating an increased gap between the second and third motors. By using a 6-layer board, these motors can be positioned closer together on one side, bringing them nearer to the outer surface of the PCB, thus increasing the field strength and potentially improving performance.

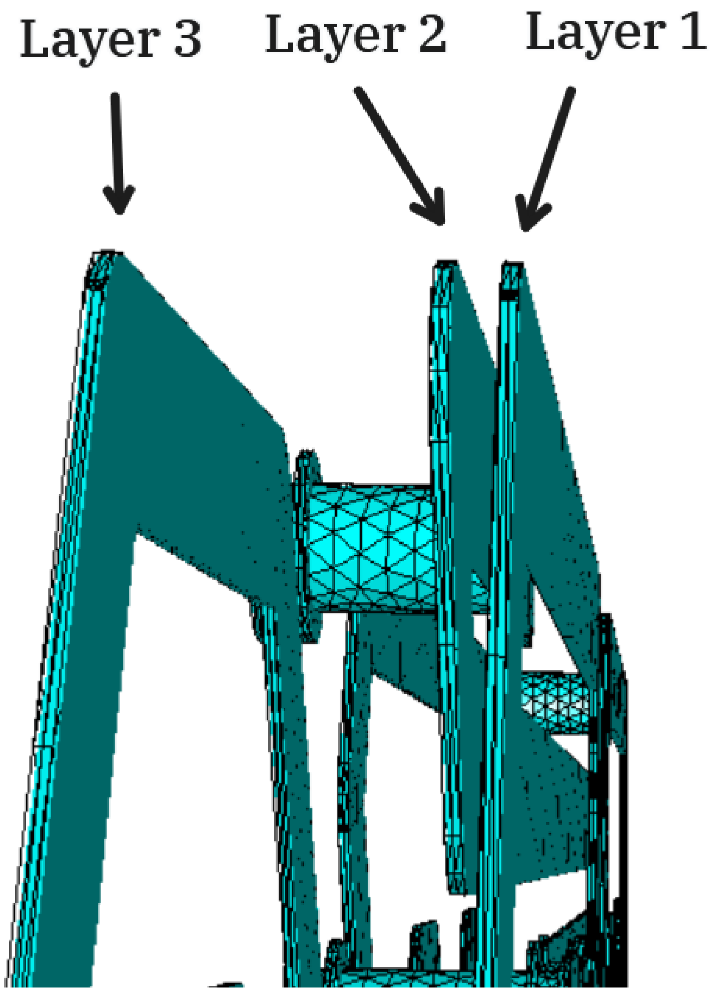


Figure 5.1: Spacing Between Layers According to PCB Construction

Additionally, the extra layers can facilitate better routing for the driver circuits which, in turn, can reduce the overall footprint of the driving circuit. This improvement also opens the possibility of designing an array of motor cells, managed by a microcontroller, enabling more complex and coordinated movements.

However, adopting a 6-layer PCB comes with trade-offs. The additional layers increase manufacturing costs and make fabrication more complex. Moreover, simulating a more intricate PCB stack-up can further complicate numerical analysis, as the increased layer count introduces additional electromagnetic interactions, making meshing and solving more computationally demanding.

Future work should also focus on refining simulation accuracy, optimizing meshing techniques, and exploring alternative simulation approaches to enhance both efficiency and precision.

References

[1] BD63242EFV-E Datasheet.pdf.

[2] Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities.

[3] IPC-2221A.

[4] THE 17 GOALS | Sustainable Development.

[5] Dhairya Sheth. An overview of maglev trains. *Scholarly Review Journal*, 2024. S2ID: 6fd93b1b4db799ef53cb2411477c0014902bab91.

[6] Guoxian Liang, Liang Guo, He Zhang, He Zhang, He Zhang, Michael Galea, Michael Galea, Jingyuan Li, Jing Li, Chris Gerada, and Chris Gerada. Multiobjective Optimization of a Magnetically Levitated Planar Motor With Multilayer Windings. *IEEE Transactions on Industrial Electronics*, 63(6):3522–3532, January 2016. MAG ID: 2342702865.

[7] Hyungwoo Lee, Hyung-Woo Lee, Ki-Chan Kim, Ki-Chan Kim, Ki-Chan Kim, Ju Lee, Ju Lee, and Ju Lee. Review of maglev train technologies. *IEEE Transactions on Magnetics*, 42(7):1917–1925, June 2006. MAG ID: 2153028751 S2ID: 34a231f1e53d77bc99af93b58b9b550133ed15a4.

[8] Ion Boldea, Ion Boldea, Marcello Pucci, Marcello Pucci, Wei Xu, and Wei Xu. Design and Control for Linear Machines, Drives, and MAGLEVs—Part III. *IEEE Transactions on Industrial Electronics*. MAG ID: 4246342088 S2ID: 7c2647e97e8ce79bbd120b46b626bab31c9e5e03.

[9] Jiří Kuthan, Jiri Kuthan, Martin Juřík, Martin Jurik, Martin Vítek, Martin Vitek, František Mach, and Frantisek Mach. Collective Planar Actuation of Miniature Magnetic Robots Towards Individual Robot Operation. *2020 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, 2020. MAG ID: 3118698723 S2ID: cd2f654660bb1fe8f18e7f6f67fe7181b52e1c31.

[10] Jiří Kuthan and Martin Juřík. Robotic System Based on Magnetically Guided Actuation on the Planar Surfaces.

[11] J.W. Jansen, J.W. Jansen, C.M.M. van Lierop, C.M.M. van Lierop, E. A. Lomonova, Elena A. Lomonova, A.J.A. Vandenput, and A.J.A. Vandenput. Modeling of Magnetically Levitated Planar Actuators With Moving Magnets. *IEEE Transactions on Magnetics*, 43(1):15–25, January 2007. MAG ID: 2109795634.

[12] Kiarash Gordiz, Kiarash Gordiz, Akanksha K. Menon, Akanksha K. Menon, Shannon K. Yee, and Shannon K. Yee. Interconnect patterns for printed organic thermoelectric devices with large fill factors. *Journal of Applied Physics*, 122(12):124507, September 2017. MAG ID: 2756665450 S2ID: e6a1ab9af46fcef010476f39335a595713a0ce98.

[13] Luis M. Castellanos Molina, Luis M. Castellanos Molina, Renato Galluzzi, Renato Galluzzi, Angelo Bonfitto, Angelo Bonfitto, Andrea Tonoli, Andrea Tonoli, Nicola Amati, and Nicola Amati. Magnetic Levitation Control Based on Flux Density and Current Measurement. *Applied Sciences*, 8(12):2545, December 2018. MAG ID: 2903775923.

[14] Martin Juřík, Martin Jurik, Jiří Kuthan, Jiřı Vlček, Jiri Kuthan, František Mach, Jiri Vlcek, and Frantisek Mach. Positioning Uncertainty Reduction of Magnetically Guided Actuation on Planar Surfaces. *IEEE International Conference on Robotics and Automation*, pages 1772–1778, May 2019. MAG ID: 2969027980 S2ID: f054f2dabfed23125e2f7e1c723f3a0ee24226fd.

[15] Martin Vítek, Martin Juřík, and František Mach. Scalable, Compact and Easy-to-Control Linear Electromagnetic Actuator. *2023 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, October 2023. MAG ID: 4388039060 S2ID: 76525023fb215d7e90ca40b061938a3d5cfc5fc1.

[16] Meng Zou, Mingxing Song, Shun Li Zhou, Xianze Xu, and Fengqiu Xu. Force and Torque Model of Magnetically Levitated System with 2D Halbach Array and Printed Circuit Board Coils. *Italian National Conference on Sensors*, 23(21):8735–8735, October 2023. MAG ID: 4387957819 S2ID: 45fcc2793d0536cebaa7f9bd9086669f90bc5891.

[17] Ron Pelrine, Ronald E. Pelrine, A. Wong-Foy, Annjoe Wong-Foy, Brian McCoy, Brian McCoy, Dennis Holeman, Dennis Holeman, Rich Mahoney, Rich Mahoney, J. Myers, Greg Myers, James A. Herson, Jim Herson, Thomas Low, and Tom Low. Diamagnetically levitated robots: An approach to massively parallel robotic systems with unusual motion properties. pages 739–744, May 2012. MAG ID: 2101686789.

[18] Ron Pelrine, Ronald E. Pelrine, Allen Hsu, Allen Hsu, Cregg Cowan, Cregg Cowan, A. Wong-Foy, and Annjoe Wong-Foy. Multi-agent systems using diamagnetic micro manipulation — From floating swarms to mobile sensors. pages 1–6, July 2017. MAG ID: 2743941662.

[19] Ronglu Wang, Lu Zhang, and Kai Yang. Overview of the Development of Planar Motor Technology. *Energies*, 2024. S2ID: 0d16c4f1ee20ed8a8117c6c6dd871ba2a94cadf5.

Appendix A

Pseudocode

Algorithm 1 Borders

Input: grid, radius
Output: The grid with borders colored

```
Function borders(grid, radius):
    value ← radius
    Define options_layer_1, options_layer_2, and options_layer_3 as lists of tuples
    for each option in options_layer_1, options_layer_2, options_layer_3:
        Filter cells where cell[option[0]] = value * option[1] and cell[3] = layer number
        Sort filtered cells by cell[option[2]]
        for each cell in filtered_cells:
            if cell color = option color and not first cell, skip 3 cells
            else: set cell color to option color
    return grid
End Function =0
```

Algorithm 2 Close Loops

Input: grid, radius, stages

Output: The grid with loops closed and vias added

```
Function close_loops(grid, radius, stages):
        Initialize empty lists: vias, vias1, vias2, vias3
        value ← radius - 2
        options_layer_1 ← [(2, 1, 0, 'blue'), (1, -1, 0, 'blue'), (1, -1, 0, 'green'), (2, 1, 0, 'green'), (1, 1, 0, 'red'), (2, -1, 1, 'red')]
        for option ∈ options_layer_1:
            filtered_cells ← [cell ∈ grid.cells if cell[option[0]] = value * option[1] and cell[3] = 0]
            Sort filtered_cells by cell[option[2]]
            i ← 0
            while i < len(filtered_cells):
                if cell color and previous/next cell color = option color:
                    set cell color to option color
                i ← i + 1
        Create filtered cell lists for layer 0, filter3 and filter4, combine into filteredsum
        Extract 'green' cells from filteredsum, sort by x coordinate
        for each green cell:
            set cell color based on index, append to vias1
            if cell is first or last, set color to 'green'
        Add vias1 (excluding first and last) to vias list
        if stages is odd and phases = 3: color specific cell in layer 0 to 'green'
        options_layer_2 ← [(1, 1, 2, 'blue'), (0, -1, 2, 'blue'), (0, -1, 2, 'green'), (1, 1, 2, 'green'), (0, 1, 2, 'red'), (1, -1, 0, 'red')]
        for option ∈ options_layer_2:
            filtered_cells ← [cell ∈ grid.cells if cell[option[0]] = value * option[1] and cell[3] = 1]
            Sort filtered_cells by cell[option[2]]
            i ← 0
            while i < len(filtered_cells):
                if cell color is blank and the colors of the cells before and after the current cell are the same as the option color:
                    set the color of the current cell to the option color
                i ← i + 1
        Create filtered cell lists for layer 1, filter5 and filter6, combine into filteredsum2
        Extract 'green' cells from filteredsum2, sort by x coordinate
        for each green cell:
            set cell color based on index, append to vias2
        Add vias2 to vias list
        options_layer_3 ← [(0, 1, 1, 'red'), (2, -1, 1, 'red'), (2, 1, 1, 'blue'), (0, -1, 2, 'blue')]
        for option in options_layer_3:
            filtered_cells = [cellforcellingrid.cellsifcell[option[0]] == value * option[1]andcell[3] == 2]        Sortfiltered_cellsbycell[option[2]]
        i = 0
            while i < len(filtered_cells) :            ifgrid.get_cell_color(filtered_cells[i]) == blank_spaceandgrid.get_cell_color(filtered_cells[i-1]) == option[3]andgrid.get_cell_color(filtered_cells[i+1]) == option[3] :
                grid.set_cell_color(filtered_cells[i], option[3])            i = i + 1
        Create filtered cell lists for layer 2, filter11 and filter12, combine into filteredsum3
        Extract 'green' cells from filteredsum3, sort by x coordinate
        for i, hd in enumerate(sorted(green_filtered3, key = lambdacell : cell[0])) :        color = vias_color_set[0]if((i+1)//2)
        grid.set_cell_color(hd, color)        vias3.append((hd, color))
```

Algorithm 3 Process Vias

Input: grid, vias, stages, layers
Output: The grid with processed vias

```
Function process_vias(grid, vias, stages, layers):
    Initialize empty list: via_pairs
    for each layer h from 0 to layers - 2:
        for each via1 in vias[h]:
            for each via2 in vias[h + 1]:
                if via1 and via2 have same x, y, z coordinates:
                    append (via1, via2) to via_pairs
    for each pair in via_pairs:
        set color of both vias in pair to 'white'
    return grid
End Function =0
```

Algorithm 4 Via Pathing

Input: grid_back, list_search, layer_color_set
Output: Paths between vias with updated grid colors

```
Function via_pathing(grid_back, list_search, layer_color_set):
    for each item in list_search:
        start ← item[0][0]
        goal ← item[1][0]
        if start and goal are on the same layer:
            new_color ← color from layer_color_set for that layer
            allowed_colors ← ['gray']
            if vias have the same color:
                allowed_colors.append(via color)
            path ← grid_back.find_path(start, goal, allowed_colors)
            for each cell in path:
                grid_back.set_cell_color(cell, new_color)
            paths.append(path)
    return paths
End Function =0
```

Algorithm 5 Find Paths by Color

Input: grid_back, layer_color_set
Output: Dictionary of paths for each color

```
Function find_paths_by_color(grid_back, layer_color_set):
    Initialize empty dictionary: color_paths_before_backplane
    for each layer in layer_color_set:
        for each color in layer:
            path ← grid_back.find_path_color(allowed_color=color)
            if path is not None:
                color_paths_before_backplane[color] ← path
    return color_paths_before_backplane
End Function =0
```