An Approach to Switching Converter Design using Machine Learning- and Device Physics-based Component Models

Skye Victoria Reese

B.S., California Institute of Technology, 2020

M.S., University of Colorado Boulder, 2022

A thesis submitted as part of the requirements for the degree of Doctor of Philosophy in Electrical, Computer, and Energy Engineering

at the University of Colorado Boulder

Boulder, CO, USA

November 2024

Committee members:

Prof. Dragan Maksimović, chair

Prof. Luca Corradini

Dr. Christian Debrunner

Prof. Robert W. Erickson

Prof. Abram Handler

Abstract

Reese, Skye (Ph.D., Electrical, Computer, and Energy Engineering)

An Approach to Switching Converter Design using Machine Learning- and Device Physics-based Component Models

Thesis directed by Professor Dragan Maksimović

Power converter design is a complicated, time-consuming process that includes many steps, including the development and applications of steady-state equations, parameter solutions, component selection, printed circuit board design and fabrication, system assembly and packaging, and experimental validation. A common objective in the design of switched-mode power converters is to meet specifications while minimizing losses, size, cost, or a combination of these performance metrics. The incorporation of data-driven design automation is a relatively new way of thinking about the converter design process. This work presents a machine learning (ML)-based, data-driven approach to the component selection step of power converter design. Large quantities of component data scraped from distributor sites and component datasheets are the foundation of the approach. The data is used to train machine learning models and then uses these models within an optimization algorithm. The desired outcome of this work is to develop a systematic way of selecting components to optimize an objective function while meeting design constraints, ultimately assisting power converter designers by significantly reducing the design time and costs by providing automated component recommendations to generate high-performing converters. The approach is developed into an easy-to-use tool that allows for rapid iteration of varying design specifications.

This work covers five major topics involved in the approach. First, the acquisition of large quantities of machine-readable component data from real, commercially available components. Second, the development of datasheet-based, design-oriented parameters for use in the power loss model of a converter topology. The goal is to develop an accurate power loss model that can be computed using only datasheet data and known quantities about the design. Third, the development of machine-learning models trained on component data. These models are used within the approach to quickly come up with proxy component parameter sets for every component in the design at each iteration of the optimization algorithm. Fourth, the development of an approach which is based on an optimization algorithm in order to solve the presented multivariate combinatorial optimization problem of minimizing an objective function subject to constraints and bounds. These optimized parameters are used to reduce the component databases such that they can be used to return sets of specific manufacturer part numbers.

The entire approach is implemented via a generalized, easy-to-use Python-based tool. Once the designer inputs their power loss model and design specifications, the tool quickly determines an optimal component combination and returns a set of manufacturer part numbers available on commercial distributor sites. Finally, the approach is validated using hardware prototypes, to compare the theoretical and experimental performance given the returned parameters. Multiple converter designs are selected for experimental validation. One design is of 48-to-12 V, 5 A synchronous buck converters, built with the objective of minimizing total power loss subject to constraints on the board area and a variety of component cost constraints, using either Si or GaN FETs for the switches. Additional design examples include a 12-to-48 V, 2 A output boost converter and a 430 W photovoltaic microinverter. The experimental results show that the presented approach can be used to successfully select components for high-performance converters that achieve design constraints.

Acknowledgements

I am forever grateful to my advisor, Dragan Maksimović, for allowing me the opportunity to pursue my PhD, for his guidance, advice, ideas, feedback, and mentorship. He made this a wonderful experience.

I would like to thank everyone in the CoPEC lab for their collaboration and support over the years. They brought great joy and insight. Specifically, I would like to thank colleagues Bailey Sauter for collaborating on the design and testing of the tool, and Abhinav Kumar, Shoudong Hu, and Vignesh Iyer for assisting in building and testing hardware prototypes.

I would also like to thank all of the committee members: Profs. Robert Erickson, Abram Handler, Luca Corradini, and Dr. Christian Debrunner, for their time and helpful comments, questions, and suggestions.

Next, I am grateful to the team at Lockheed Martin, including Tom Byrd and Antuwon Butler, for their support and for providing opportunities for this work.

I would also like to thank my Caltech advisor Yu-Chong Tai, who fervently inspired me to become a philosopher of knowledge when I was an undergraduate.

Finally, my endless love goes to my husband Jared Loucks, and my parents and sister Sean Reese, Elizabeth Reese, and Megan Martin, as well as all of my family and friends for supporting me throughout my life and encouraging me to pursue my dreams.

To Grammie, an endless source of light, love, and inspiration.

Contents

List of Tables

List of Figures

List of Algorithms

Listings

Chapter 1

Introduction

This chapter introduces the thesis. Section 1.1 provides a background and motivation for this work. Current methods are presented to solve the converter design problem, and these methods are compared to the presented approach. A summary is included describing various other data-driven methods in power electronics. Section 1.2 discusses the main contributions of the thesis, while Section 1.3 lists and briefly describes the main topics covered in each chapter. Finally, Section 1.4 concludes this chapter.

1.1 Background and challenges

A common objective in designing switched-mode power converters is to meet the specifications while minimizing loss, size, cost, or a combination of these converter performance metrics. Consider, as an example, the design of a simple synchronous buck converter shown in Fig. 1.1. Given the specifications, the design objectives, and the converter steady-state solution, operating characteristics and loss models, the designer may manually select the power stage components among many thousands of available options, considering information such as the switching frequency and the operating mode, *e.g.*, hard-switched continuous conduction mode, discontinuous conduction mode, or soft-switched zero-voltage-switching quasi-square-wave mode. One may note

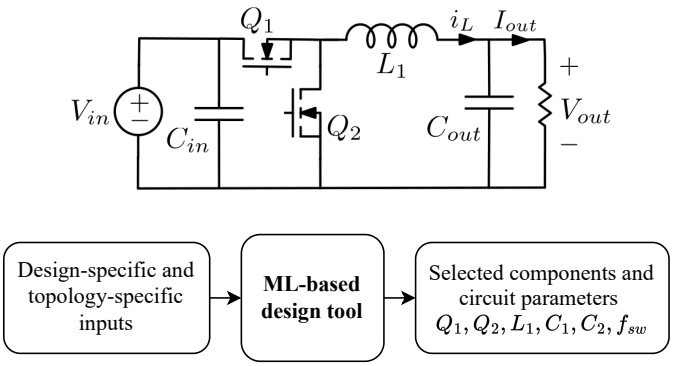


Figure 1.1: A top-level flowchart of how the ML-based design tool is used to select components in a synchronous buck converter example.

that the converter design space can be very large even in the simple example of Fig. 1.1. The design challenges are even more pronounced in practical power electronics systems that involve complex trade-offs in converter architectures, converter topologies, and application-specific constraints.

The standard design approach relies on narrowing the design space based on the designer's knowledge and experience, combined with the use of simulation tools and manual filtering features on commercial component distributor websites. However, the process can be time-consuming given the size of the design space. Furthermore, while many options can meet the specifications, there is no guarantee that any particular choice of architecture, converter topology, components, switching frequency, or operating mode best meets the design objectives and constraints.

Standard design approaches can be improved by incorporating data-driven methods. Such design automation methods have been gaining attention [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. A broad overview of artificial intelligence techniques for power electronics has been presented in [0], while references [0, 0] discuss data-driven techniques for inductor design and core-loss modeling, respectively. Algorithms for component parameter design, where the optimization objectives include volume, efficiency, and cost are presented in [0, 0, 0, 0], whereas [0] uses simulation tools and batch-normalization neural networks to determine operating conditions and component parameters for power

converters. An architecture-level optimization using descent-based design automation methods has been presented in [0], followed by [0] where machine learning models are developed to predict dc-dc converter performance and speed-up the architecture-level design.

This work aims to solve the presented combinatorial optimization problem, which aims to find the set of components that optimizes an objective function subject to constraints. The exact method to solve such a problem is an exhaustive search over all component combinations. However, this quickly becomes intractable given the large design space and complex topology. Various heuristic methods exist that can search over the space and get within a margin of the exhaustive search result. Examples of heuristic search algorithms include greedy search [0], simulated annealing [0], and ML model-based methods [0].

The objective of the approach presented in this thesis is to assist the designer by formulating the choice of the converter components and circuit parameters as an optimization problem that can be solved using a computer programming optimization algorithm, and by using ML-based component models to efficiently iterate over design parameters, as shown in Fig. 2.1. Overall, the design time and cost can be significantly reduced by using the computational power of machines to provide specific manufacturer part numbers for every component in the design that optimizes for the designer's objective, thus leading to a high-performing design with much less work on the part of the designer.

The ML-based design approach relies on large quantities of component data to train the models, and the same data set is used in the exhaustive search. Information on many of the commercially available components can be found on distributor websites such as Digikey and Mouser. Although this information is not currently available in machine-readable format, the websites can be directly scraped, and the links to the component datasheets (most often in PDF format) can be opened and scraped. This data can then be used in a variety of automated design approaches. One automated approach is an exhaustive search across all component combinations in the database.

Given a power loss function and a converter topology, all of the possible combinations of components can be iterated over until a combination is found that meets the constraints and minimizes the design objective. This matrix reduction-based approach is described and implemented in [0] for a buck converter example. The implementation of this method eventually found the combination of two transistors and one inductor that minimized a very simple loss model while meeting cost and area constraints, but this approach took significantly longer in run-time and ultimately became intractable when more components or more complex loss modeling equations were considered.

Although the exhaustive search approach is limited in efficacy because of the long computational run-time, it offers the advantage of computing the losses of combinations of real components. Therefore, if the dataset was smaller, this approach could once again be used. This motivates the use of pre-trained ML models and an optimization algorithm to limit the datasets to only include components that are expected to result in optimal performance, and then uses the exhaustive search approach to account for the bias and variance associated with ML model predictions and thus determine the optimal component combinations out of the reduced datasets.

Given this motivation, the complete approach visualized via the block diagram in Fig. 2.1 is designed and implemented as a Python-based tool. The goal was to create a tool where the designer could simply put in their design specifications and topology, run the program, and within a matter of minutes have specific manufacturer part numbers for every component in their design and a set of operating conditions that can meet constraints and lead to a high-performing converter with respect to the designer's specified objective.

1.2 About this thesis

This section discusses the major contributions of this thesis.

First, the work gathered a large database of commercially available component data

across a wide range of manufacturers. The intention behind this database is to make it open source so that it can be used for many other data-based contributions to the power electronics field. This database was used to visualize commonly considered figures of merit, which can be used to determine the quality of components. In addition, effective methods are developed for how to acquire data from the distributor sites and PDF datasheets and turn this data into formats that can be used to train ML models.

Second, this work developed machine learning models using the datasheet data. Although the inputs and outputs of the ML models were determined based on the applications of the tool, the same model structure can be used to predict component information given any set of inputs. The data was filtered for use in power electronics applications, for example by implementing the concepts of Pareto dominance, outlier removal, and data filtering. The performance results of the ML models are analyzed and limitations are discussed.

Third, this work develops and incorporates various datasheet- and device physics-based loss modeling methods. These approaches combine datasheet information with operating conditions determined by the design specifications, in order to determine the power loss contributions from various major component parameter losses. This is an important step in the design method because a well-developed, accurate power loss model is necessary to determine which components will lead to the best performing designs when efficiency is included in the objective function and/or constraints.

Fourth, all of the aforementioned thesis contributions are combined into the rapid, high-performing converter design approach that is the critical result of this work. This is a novel approach that can quickly and automatically determine components leading to high-performing converters for any converter topology. This approach combines the ML models and datasheet-based loss modeling methods into an optimization algorithm that determines optimized component parameters, and uses these component parameters to filter the large component database down to the best components. These component sets are used in an exhaustive search to determine the combination that

leads to the highest-performing design, and then finally does an automatic switching frequency optimization given the selected components. This approach is implemented via a Python-based tool, and is shown to effectively assist the designer in the converter design process, based on the experimental prototype results for a variety of converter topologies. The tool is designed such that it is easily generalized to any converter topology and design. Thus, any power electronics designer may use this tool to efficiently design high-performing converters for any application.

1.3 Chapter list

This dissertation is organized into eight chapters, starting with this introductory Chapter 1, where the background, motivation, challenges, and contributions are briefly summarized.

Chapter 2 ML-based converter design approach. This chapter provides an overview of the entire ML-based converter design approach, describing the block diagram of the tool as shown in Fig. 2.1. This chapter discusses the problem formulation and how it fits into the selected optimization algorithm and component selection process. Specifically, the optimization algorithm takes information about the converter topology and operating conditions, uses this information as inputs to machine learning models to predict component parameters, and iterates over design variables until the algorithm reaches convergence. Then the optimized component parameters are used to filter the dataset and perform a matrix-based exhaustive search.

Chapter 3 Dataset acquisition. This chapter discusses the acquisition of the data used to train the ML models and perform the exhaustive search. High quality data leads to high quality ML models, however, most datasheets for commercially available components are not in machine-readable format. Therefore, gathering a large dataset was important. Device figures of merit based on the component data are developed and analyzed.

Chapter 4 Design-oriented power loss modeling. This chapter describes the power loss modeling methods used to predict the expected losses in the converter. Estimation techniques combining datasheet data with background knowledge of device physics are validated using various datasheet and hardware examples.

Chapter 5 Machine learning models. This chapter discusses the development of machine learning regression models based on the datasheet data for use in the design approach. The chapter includes an analysis of the results of the ML models and insights gathered from the ML model performance results. Limitations with the ML models and potential mitigations are also discussed.

Chapter 6 Component selection. This chapter discusses the process of using the results of the optimization algorithm step to create subsets of the datasets that can be used for an exhaustive search. The exhaustive search provides the designer with specific manufacturer part numbers for every component in the design, and ensures the components meet all design constraints and bounds and are selected to optimize for an objective function.

Chapter 7 Design examples. This chapter discusses a 48-to-12V, 5A output dc-dc synchronous, hard-switched buck converter example, a 12-to-48V, 2A output dc-dc synchronous, hard-switched boost example, and a 430W microinverter example. Various FET technologies and other design constraints are iterated over. The component combinations returned by the tool were used to design the prototype printed circuit boards (PCBs). The prototypes were tested and compared to the theoretical results. The results show that the tool successfully allows for rapid design of high-performing converters, and suggests how the power loss model could be improved.

Chapter 8 Conclusions. This chapter provides a summary of the thesis and presents potential future research directions.

1.4 Conclusions

This chapter introduces the thesis, describes the research objectives, and presents the main contributions of this work. Each chapter is briefly summarized, where the chapters include an overview of the ML-based converter design approach, the process of dataset acqusition, design-oriented parameter development, ML model development, the component selection process, multiple design examples using the tool, and a conclusion of the thesis.

Chapter 2

ML-based converter design approach

This chapter describes the entire approach as shown in the block diagram in Fig. 2.1, which is turned into a Python-based design tool. This approach can be applied to any converter topology and design specifications. The approach aims to combine knowledge of device physics and losses, feature relationships and predictions, and an exhaustive search over component database combinations into one efficient, machine learning-based automated design approach. Each chapter in this thesis covers in detail an element of the block diagram (where specific chapter references are in bold in the block diagram). This chapter first compares the presented approach to other methods of solving the converter design optimization process in Section 2.1. Descriptions are provided of the standard manual approach, alternative combinatorial optimization methods, and the presented automated ML-based approach to achieve converter design and optimization. Section 2.2 describes what user inputs are required to run the tool, pointing to how the tool can easily generalize to different topologies and design objectives. Section 2.3 describes how the problem is structured as an optimization problem to be solved by an optimization algorithm, and Section 2.4 discusses which optimization algorithm was selected. Finally, Section 2.5 concludes this chapter.

2.1 Comparison between design approaches

This section explores the motivations for developing an ML-based converter design approach by comparing it to other standard approaches. The presented design approach can be used for a wide range of converter designs, as shown by the buck, boost, and microinverter examples presented later in the thesis. It is acknowledged that for the buck and boost examples, the component selection process is fairly simple for a designer to complete by hand by solving the steady-state equations and filtering the distributor websites in order to select the transistors, the inductor, and the capacitors. However, this design approach is less systematic and less structured in the form of an optimization problem, where meeting the specifications is often satisfactory. However, as more considerations are added into the design problem such as the complexities seen in the microinverter example, automated tools can offer considerable advantages to the designer when selecting components that will meet design constraints and lead to high performance.

A successful implementation of the approach would allow designers to input their design specifications and topology selection, and in a matter of minutes have specific manufacturer part numbers for each component in their design that will meet all design requirements and achieve a high-performing converter as shown in Fig. 1.1. The objective function of the optimization algorithm and the constraints can be any combination of the performance metrics predicted by the tool, such as power loss, BOM cost, and BOM PCB area. In addition, the power loss model equations are improved using design-oriented parameters combining datasheet quantities and device physics, to generate accurate predictions of the expected converter loss.

A description of the differences between various design approaches can help to illustrate where an automated approach can be particularly advantageous to a designer, such as by saving time and cost.

2.1.1 The standard manual approach

To design a converter using the manual approach, all of the equations representing the specified loss model must be known. These equations may include some datasheet quantities, but the equations developed for the automated tool augment these datasheet quantities based on operating conditions. Given any known rating requirements, the designer could then filter to find all components that meet these requirements. They could then select a component, open its datasheet, and calculate the losses associated with that component. However, some of these losses depend on the value of another component, e.g. transistor reverse-recovery charge losses, which in turn depend on the inductor current and the inductance value. This adds complexity to the selection process. Given these selected components, the designer can then check that the total costs meet a cost constraint, and that the areas meet the PCB area constraint. If they do not, then the designer will have to choose which component(s) they will allot less cost or area to. There is no guarantee that their initial selection will have values that lead to the lowest possible loss–potentially, the converter could perform much better given a different selection of components. Trade-offs to meet constraints are also hard to evaluate, e.g., whether cutting costs on the inductor versus the transistors will lead to better or worse performance based on the operating conditions. Although the automated design tool does not claim to lead to the absolute best-performing selection of components (which would only be possible if every component combination could be considered, using a completely accurate power loss model that accounts for all operating condition variations), it helps with incorporating the operating conditions into the power loss calculation, assuring the components will meet constraints, and accounting for how variations in parameter distribution can affect the overall performance of the converter.

2.1.2 Alternative combinatorial optimization methods

Given the data collected as part of this research, various multivariate combinatorial optimization methods could be considered to find a set of optimized components. One

method is the exhaustive search, which looks at all combinations of components. Although this is considered an exact method, it quickly becomes intractable. The exhaustive search was implemented for the converter design component selection problem presented here, and [0] shows how the approach takes many orders of magnitude longer and eventually becomes intractable for standard operating systems. Various heuristic methods exist, which can come close to the optimized solution without reaching it exactly, while also significantly reducing the execution time. One heuristic method is to use ML models. This method is what was selected in this thesis, where the ML models capture trends and use these trends within an optimization algorithm. In this implementation, the ML-based step is followed by an exhaustive search step.

Many other heuristic combinatorial optimization methods exist, such as a greedy search [0] or simulated annealing [0]. These methods are beneficial because they search across the discrete component combinations from the start, and so may determine a set of components that is closer than the ML-based method to what could be achieved with the exhaustive search. However, these approaches can take longer than ML-based methods for complex problems, because the ML method relies on pre-trained curves that need only be trained once. For the ML method presented here, any number of components can be added to the database with little impact on total run-time, because the number of components considered is constant in the exhaustive search step. An additional benefit to not looking at the discrete combinations initially is that results are not as sensitive to component availability. For example, if a certain inductor becomes unavailable, some methods may require an entire re-run of the algorithm to find a new solution, whereas in the ML-based approach, the results after the optimization step would essentially be the same because the trends are captured. These results can then be used in the exhaustive search without considering the unavailable component. Therefore, the approach is robust to component fluctuations.

2.1.3 Automated ML-based tool approach

Next, we will present the process of using the automated tool. First, the designer opens up the Python template file and adds a statement with their topology name (see Listings 7.1, 7.2,7.3, 7.4 for an example template). This template has all of the functions that must be filled in for that converter topology, and all of the variable names that must be used to specify the loss model and any steady-state equations to find component values. There is also a function where the designer inputs a dictionary of their design specifications and constraints. The same topology-specific inputs (power loss model, steady-state equations), as shown in Fig. 2.1, can be re-used for any design. Next, the designer simply runs the tool by running the script. No further interaction is required. The tool returns specific manufacturer part numbers, expected losses, cost, and area, as well as an optimized switching frequency given the selected components. The designer can then go onto distributor sites and directly paste in these part numbers. The designer can be confident the selection will meet all design constraints and has considered a well-developed power loss model based off the datasheet values of component attributes, augmented to account for design specifications.

The scraped data (Ch. 3) is combined with design-oriented parameters (Ch. 4) and used to train ML models (Ch. 5), which use known parameters about a converter design to predict unknown component parameters associated with loss, cost, and package size. All of these parameters are used within the optimization algorithm (Ch. 2) to compute a detailed power loss equation representing the losses associated with the converter topology, as well as the contributions to total BOM PCB cost and area for each component. At each iteration, the selected optimization algorithm computes the objective function and checks the constraints. When the algorithm converges, the predicted optimal parameters are used to reduce the component database to a size easily handled by an exhaustive search (Ch. 6), which returns the optimal component combination and operating conditions. These components and operating conditions can then be used to construct a prototype, as discussed for several example converters

(Ch. 7).

When implementing the approach as a software tool, the software was specifically structured to generalize to be used for any design specifications and topology. The user does not need to open the code for the tool itself, all they need to do is to input their power loss model into a separate template, as well as their set of design specifications and constraints. After running the program, < 10 minutes later the tool prints out combinations of components that will lead to a high-performing converter given the specified design requirements.

This chapter describes the design approach shown in Fig. 2.1, which represents a fully automated design approach that is simple to use. The approach combines the pre-trained ML models into an optimization algorithm and determines the component parameter values which will lead to an optimized objective function as seen by the algorithm. This objective function can include metrics such as power efficiency, total PCB BOM cost, total PCB BOM area, or a combination of these. Because the tool is based on ML models, which create approximations of components by predicting the component values, an exhaustive search step is added based on the optimized values to determine an optimized combination of real, commercially available components. The results from the ML step are used to filter the database to a smaller subset, such that the exhaustive search can be run on standard computational engines. The manufacturer part numbers of the final component combination are returned to the user. The user can then use these components or find similar components with which to build the converter hardware.

It is a challenging question to consider how the presented approach compares to the traditional design approach in terms of how much better a converter can be arrived at using the tool. To truly determine the 'optimal converter', all possible iterations of components available would have to be built and tested via hardware, and tested across all operating conditions. This is infeasible. The next-best option would be to ask an experienced power electronics designer to select components to optimize the

objective function of a converter. The results would likely vary between designers, and the results would also depend on the amount of time the designer has available. This can become a limiting factor when the design is extremely complex, or when a designer is interested in comparing the results between many different converter topologies and operating conditions. Overall, the presented approach offers a systematic way of selecting components that optimize an objective function and meet all design constraints, and allow for rapid results and iteration with very little work required on the part of the designer.

2.2 Design tool implementation

The design tool diagrammed in Fig. 2.1 and implemented in Python is built so that it can be applied to any converter topology and any set of design specifications. The inputs $\bar{T}$ are broken into two groups: design- and topology-specific. Listings7.1, 7.2, 7.3, 7.4 show code examples of these inputs for a buck converter example.

The first topology-specific input is the set of component indices to be included in the optimization process. The designer can choose to include any number of transistors (Q), inductors (L), and capacitors (C). The second topology-specific input is the set of optimization variables, where the designer determines which variables the tool should iterate over. The third topology-specific input is the set of steady-state solution equations, including standard relationships between control variables, such as the duty cycle and switching frequency, and operating characteristics, such as the conversion ratio, voltage and current stresses, and switching ripples. The fourth topology-specific input is a power loss model. This loss model consists of two parts: the computation of relevant datasheet-based and physics-based loss parameters for each component, and the equations that compute the total power loss based on the precomputed physics-based loss parameters and the operating conditions. The pre-trained ML models, the optimization algorithm, and the final step in which specific component candidates are selected remain the same regardless of the specific converter topology or design-specific

inputs.

The design-related inputs include the selected converter topology label, the operating mode selection, and the design specifications: input and output requirements, all component rating and value requirements, the selected converter metric to be optimized, and the design constraints, *e.g.*, on power efficiency, BOM cost, or BOM footprint area.

2.3 Optimization problem formulation

The converter design is posed as a constrained optimization problem,

$$\begin{aligned} &\min_{\bar{O}} f(\bar{X}, \bar{O}) \\ &U_{tot} \leq U_{max,total} \\ &A_{tot} \leq A_{max,total} \end{aligned} \tag{2.1}$$

The objective is to minimize a user-defined converter performance metric $f(\bar{X}, \bar{O})$, such as power loss, under user-defined constraints, such as U and A limits. The objective function $f(\bar{X}, \bar{O})$ and the constraints can be defined in many different ways, depending on the user's specified design objectives.

Input $\bar{T}$ is split up into inputs related to the overall converter $\bar{T}_{con}$ and inputs related to each device $\bar{T}_Q, \bar{T}_L, \bar{T}_C$ for transistors, inductors, and capacitors, respectively. The mapping from $\bar{O}, \bar{T}$ to $\bar{X}$ required to compute $f(\bar{X}, \bar{O})$ is facilitated by the one-time trained ML-based component models,

$$\bar{O}, \bar{T} \underset{\text{ML models}}{\longrightarrow} \bar{X} \tag{2.2}$$

ML models m_Q, m_L, m_C predict $\bar{X}_Q, \bar{X}_L, \bar{X}_C$, which are all of the parameters required for computing the power loss-, cost-, and area-related quantities of each component. A pseudo-code of this step is shown in Algorithm 1. The output of the optimization

algorithm $\bar{X}_{opt}, \bar{O}_{opt}$ is used to filter each component dataset, and these subsets of components are used in an exhaustive search across component combinations as discussed in Chapter 6.

The results of three different steps in the presented design approach may be used. First, the results $\bar{O}$, $\bar{X}$ of the optimization step can be used for quick estimation of a topology performance. These results can be used to compare topologies, operating conditions, and constraint values without considering specific components. Second, the results from the component selection step can be used to generate possible component combinations that meet all design constraints. From here, the designer may decide which combination they want to use, depending on factors such as up-to-date availability, prices, and manufacturer preferences. Finally, specific component selections may be used to evaluate expected performance using those components, including a detailed breakdown of power loss, cost, and area contributions. Examples are provided in Chapter 7.

2.4 Optimization algorithm selection

Various optimization algorithms are compatible with the flowchart of Fig. 2.1 and the pseudo-code in Algorithm 1. The algorithm selected is COBYLA (Constrained Optimization BY Linear Approximation) [0], which is a constrained, bounded, multivariate optimization method, an implementation of which is available via the Scipy optimize class [0]. COBYLA creates successive linear approximations of both the objective function and constraints. At each iteration, the approximation is minimized using a simplex within a trust-region, and this candidate solution is evaluated using the original objective function and constraints. The trust region is reduced to fine-tune the search, ultimately converging when the trust region drops below a threshold size. The algorithm hyperparameters are chosen to assist in successful convergence and effectively search over the entire design space. These parameters include the tolerance for termination, the tolerance for constraint violations, and the maximum number of iterations.

Algorithm 1: Algorithm for optimization step of design approach

`// Converter specs, design variables, and objective function`
input : $\bar{T}, \bar{O}, f(\bar{X}, \bar{O})$
`// List of component indices`
input : $\bar{Q}, \bar{L}, \bar{C}$
`// COBYLA algorithm hyperparameters`
input : rhobeg = 1.0, tol = 10e-11, maxiter = 1000
`// ML models`
input : m_Q, m_L, m_C
output: $\bar{O}_{opt} = O_{Q,opt}, O_{L,opt}, O_{C,opt}$, $\bar{X}_{opt} = X_{Q,opt}, X_{L,opt}, X_{C,opt}$: optimal parameters for each component and optimal operating conditions, to be used to filter the databases prior to the exhaustive search step

`// Initialize` $\bar{O}$
$\bar{O} \leftarrow \bar{O}_{init}$
while (iter < maxiter) **do**
  while (trust_reg < tol) **do**
    `// Generate` $\bar{X}$ `(values used to calculate objective function and constraints)`
    for q $\in \bar{Q}$ **do**
      $\bar{X}_q = m_Q(\bar{T}_Q, \bar{O}_Q)$
    end
    for l $\in \bar{L}$ **do**
      $\bar{X}_l = m_L(\bar{T}_L, \bar{O}_L)$
    end
    for c $\in \bar{C}$ **do**
      $\bar{X}_c = m_C(\bar{T}_C, \bar{O}_C)$
    end
    `// Update` $\bar{O}$ `and trust region size based on COBYLA minimization of objective function`
    $\bar{O}, trust\_reg \leftarrow$`min(`$f(\bar{X},\bar{O})$`)`
  end
end
return $\bar{O}_{opt}, \bar{X}_{opt}$ and convergence message

2.5 Conclusions

The chapter presents benefits and limitations of the proposed data-driven design approach compared to other design approaches. In addition, this chapter discusses the implementation of the optimization step of the automated design approach. The approach uses the dataset presented in Chapter 3, the power loss model presented in Chapter 4, and the trained ML models from Chapter 5 to determine optimal component parameters by using an optimization algorithm to determine proxy component parameters based on predictions. These component parameters are then used to create subsets of the datasets, which are used within an exhaustive search to return to the user the optimal component combination and operating conditions, as discussed in Chapter 6.

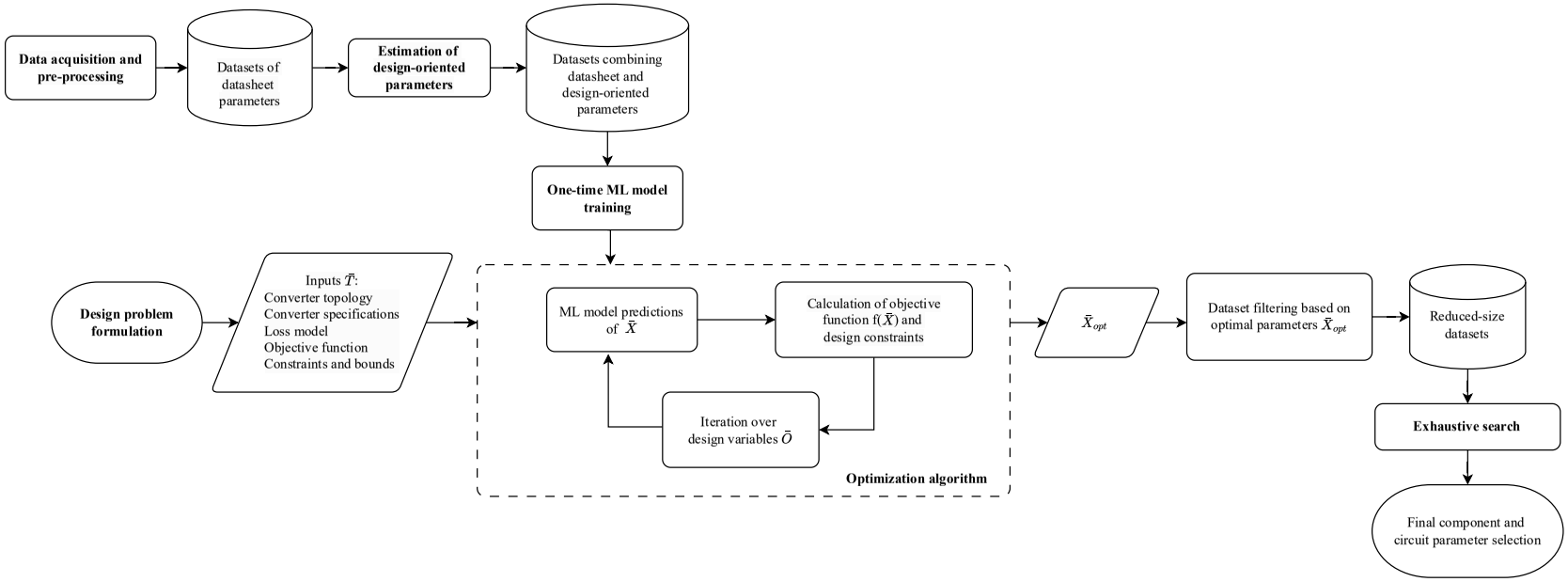


Figure 2.1: Flowchart of the switching power converter design tool using machine learning-based component models. Each bolded item is discussed in detail in a corresponding section: the design problem formulation and implementation in Chapter 2, data acquisition and pre-processing in Chapter 3, design-oriented parameters in Chapter 4, ML modeling in Chapter 5, component selection in Chapter 6, and complete design examples of various topologies using the tool in Chapter 7.

Chapter 3

Data collection

This chapter examines one of the most critical steps to developing an effective data-based design approach: the process of data acquisition and generating a high-quality dataset. Machine learning models, a significant element of the design approach, follow a "garbage in, garbage out" philosophy. It is necessary to gather a large quantity of high-quality data upon which to train the ML model in order to develop ML models that accurately predict component outcomes. In addition to ML model training, the collected component database is used to perform an exhaustive search, where the component parameters from the datasheets are used with the developed design-oriented parameters (discussed in Chapter 4) to compute the total power losses given a combination of commercially available components. One significant challenge with data acquisition is that component data is not readily available in machine-readable format. Therefore, a process was developed in this work to collect data from component datasheets.

The data acquisition process described here is general and can be used to obtain component data even as the websites update with pricing and component availability. Any component information available on the websites and datasheets could use a similar process to create a machine-readable database. This process can be used to gather

additional component data to be used in various future data-driven component-related research efforts. For example, the component data may be used to develop an automated custom inductor design approach, or to visualize and select components based on their figures of merit, as discussed in [0]. This chapter describes the process for obtaining and cleaning the datasets of components used in the current version of the tool: power FETs, fixed MPP inductors, and Class II multi-layer ceramic (MLC) capacitors datasets cleaned using outlier removal and Pareto dominance. Additional component databases that can be acquired include those of transformers and gate drivers. The more component databases available, the more components can be added to the optimization algorithm representation of loss, cost, and size. This will benefit the goal of representing the entire converter as part of the design objective, rather than only the power stage.

This chapter is organized as follows: Section 3.1 describes the process of acquiring substantial amounts of data and turning it into machine-readable form to be used by automated tools. Sections 3.2 and 3.3 describe outlier removal and taking the Pareto front, respectively. These are two methods of cleaning the data for use in the presented application. Section 3.4 shows figures of merit and corresponding visualizations using the collected transistor, inductor, and capacitor data. Finally, Section 3.5 concludes this chapter.

3.1 Data acquisition

The first step to developing the ML-based optimization tool is generating a high-quality, high-quantity set of training data for the ML models. Using web-scraping tools [0] available in Python, the component information on 22,000 power FETs, 48,000 fixed inductors, and 110,000 Class II MLC capacitors was collected from commercial distributor sites, including Digikey and Mouser. All available attributes were scraped and included in the database. Similar methods could be used to scrape additional data, e.g., component information on additional classes and temperature coefficients of capacitors. This

process used a built-in Python package called "Beautiful Soup". However, some important device parameters and characteristics are only available in component datasheets, which come in different formats and are not as easily accessible electronically. Two examples of parameters only available in PDF datasheets are MOSFET body-diode reverse recovery charge and device capacitance values, both of which are important to modeling switching losses. A method of opening and scraping the PDF datasheets was developed. The process scraped the PDF datasheets by collecting each included table as a separate graph. Therefore, each datasheet was associated with many graphs. Once these graphs are stored in association with a particular component, they can be opened and searched at any time to find a particular value. A Regex-based process was developed to match text to find the desired numerical value, but this process involved some labor to develop. This is because the way that the values and text labels are represented when turned into machine-readable form varies from datasheet to datasheet. For example, to find the value corresponding to the reverse recovery charge Q_{rr} value, the datasheet data could represent Q_{rr} as 'Qrr', 'Q rr', 'QRR', 'Q RR', among many such permutations. Then to find the associated value, various case statements had to be checked because sometimes the numerical value of Q_{rr} was placed right after the text, and sometimes there were non-alphanumeric characters in between, to list a few different cases. Cases were developed to cover all of the different scenarios the process may see, but this was time-consuming to develop. A complete set of transistor data was available for 3,100 components. Very little additional information was available in inductor and capacitor PDF datasheets, so only the main distributor page scraping was applied. Table 3.1 shows the complete list of the quantity of each component parameter that was scraped and the source of acquisition.

One challenge with data acquisition is that the data included in datasheets can be inconsistent across manufacturers. One common instance of this inconsistency was when gathering Q_{rr} data, because many manufacturers simply do not include these values on their datasheets, so only a limited number could be acquired. In the future, design approaches such as the one presented here would be greatly improved by manufacturers

Data source	Distributor site			PDF-numerical	Datasheet	
Component type	Power FETs	Inductors (surface-mount MPP)	Capacitors (Class II MLC)	Power FETs	Power FETs	Capacitors (Class II MLCC)
# of scraped components	22,000	48,000	110,000	3,100	100	25
Available attributes	• V_{dss} • $R_{ds,on}$ • Q_g • Cost • Package case • FET technology • FET type	• Inductance • Cost • Size • Thickness	• Capacitance • V_{rated} • Cost • Size • Thickness • Temperature coefficient	• C_{oss} • $V_{ds,meas}$ • Q_{rr} • I_F • dI_F/dt • t_{rr}	• C_{oss} vs. V_{ds}	• ΔC vs. V_{dc} • R_{ESR} vs. frequency

Figure 3.1: Components and component parameters scraped from various sources.

creating consistent, exhaustive, and machine-readable datasheets for all components.

After gathering the data, various pre-processing techniques were applied to improve the quality of the dataset and retain components likely selected by a designer. These techniques include outlier removal and Pareto dominance.

3.2 Outlier removal

By observing visualizations of the data, it was found that some of the components appear to be outliers with characteristics substantially different than the majority of similar components in the dataset. Upon investigating these individual outlier components manually, it was found that the parameter values reported on the distributor's main webpages (and thus, what was scraped) occasionally differ from the values in the component datasheets. These values can be manually updated, but various ML-based outlier removal techniques are available to automate the outlier identification and removal process. Here, we use a multivariate outlier removal method based on the Mahalanobis distance, where each datapoint is considered with respect to the trend [0]. The distance between every point and the center of a distribution is calculated, and the number of standard deviations is used to determine which components are outliers. The larger the Mahalanobis distance, the more likely a component is an outlier. The formula to calculate the Mahalanobis distance is:

$$D^2 = (x - m)^T C^{-1} (x - m) \tag{3.1}$$

where D is the Mahalanobis distance, x is a row in the dataset, m is the mean of each column, and C is the covariance matrix. Cleaning the data in this way improves the predictive accuracy of ML models because model algorithms are influenced by outliers.

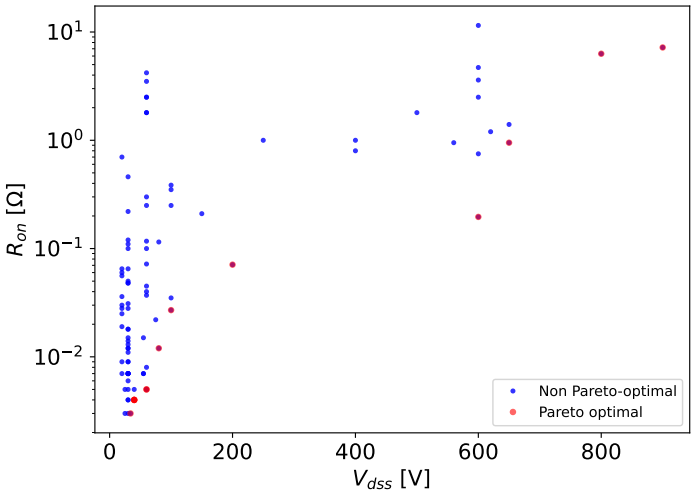
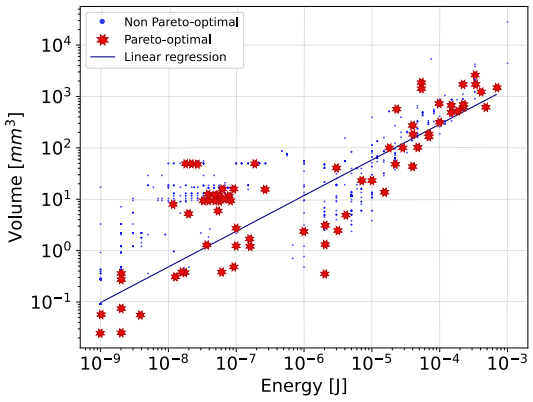
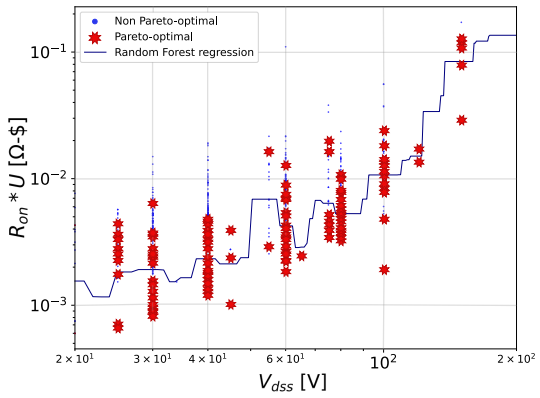


Figure 3.2: Retaining the lowest envelope of a subset of transistor data using the Pareto front in 2D, where the only considered features are V_{dss} and R_{on}.



(a)



(b)

Figure 3.3: Examples of datasets and ML regression models trained on Pareto dominant data points for (a) inductors with a fixed I_{max} current rating, and (b) Si MOSFETs with a fixed R_{on} rating.

Table 3.1: Features included in taking the Pareto front of each component's dataset. $U =$ component cost, $A =$ package area, and $H =$ package height.

	Features
Transistors	V_{dss}, R_{on}, Q_g, U, A
Inductors	I_{max}, R_{dc}, U, A, H
Capacitors	V_{rated}, U, A, H

3.3 Pareto dominance

Pareto dominance is a technique useful for determining the best envelope of components based on parameters, *i.e.*, dimensions of interest. The relevant features included for each device type are shown in Table 3.1. Pareto dominance looks at all of the specified features for a dataset and determines if a component is strictly worse than the rest and therefore unlikely to be used in an optimized design [0]. Taking the Pareto front on n dimensions yields an $(n-1)$-dimensional surface of Pareto optimal data points. This is most clearly seen when taking the Pareto front on two dimensions, *e.g.*, voltage rating V_{dss} and on-resistance R_{on}), as shown in Fig. 3.2. The 2D graph clearly shows the 1D curve of the lowest envelope. Extending this concept to all dimensions relevant to the optimization tool, Fig. 3.3 shows a 2D plot of the datapoints after taking the Pareto front on all dimensions shown in Table 3.1 for inductor and transistor data. Taking the Pareto front reduces the variance in the data.

3.4 Device figures of merit

For power MOSFETs, various figures of merit (FOMs), such as $[0, 0, 0, 0, 0, 0]$, have been considered to (i) compare semiconductor device technologies and guide device technology developments, and (ii) help designers select devices best suited for particular converter configurations and applications. A common objective has been to formulate a relatively simple physics-based FOM expression in terms of the device parameters such that the FOM is indicative of a switched-mode power converter performance metric of interest, such as power loss, cost, or area. Similarly motivated approaches, such as $[0, 0]$, have been pursued for scaling and sizing of passive components – inductors

and capacitors.

Converter-level performance metrics, such as efficiency, size, cost, etc., have very complex dependencies on component characteristics, and these dependencies vary significantly across input/output specifications, converter topologies, operating modes, and application-specific requirements. For example, a device FOM deemed appropriate for a hard-switching pulse-width modulated converter is unlikely to predict device performance in a soft-switching resonant converter. This is why this thesis takes a different, data-driven approach to the development of component FOMs and models suitable for switched-mode power converter design optimization. By obtaining and manipulating the large quantities of data available on commercial distributor sites using web-scraping tools, one may gather insights into the relationships between features of available components beyond what can be predicted from conventional physics-based models.

By automating the process of gathering and pre-processing parameters available in component datasheets, the analysis of component characteristic relationships becomes tractable over a large design space. The objective of this section is to present a novel approach to component characterization using commercially available data. In particular, we analyze transistor, inductor, and capacitor features relevant to power loss, cost, and size. All FOMs are presented after filtering (to include components most relevant to power converters), outlier removal, and Pareto-dominance have been applied to the original datasets. The analysis then compares the linear versus quadratic representations of these FOM relationships on the log-log scale.

3.4.1 Power FETs

Three FET technologies are available in the datasets: Si, GaN, and SiC. Among transistors, 21,000 were Si, 70 were GaN, and 330 were SiC. After filtering to include only N-channel devices, then removing outliers and taking the Pareto front on the dimensions shown in Table 3.1, 830 Si, 36 GaN, and 65 SiC components remained to construct data-based FOMs. Converter conduction and switching loss relationships to

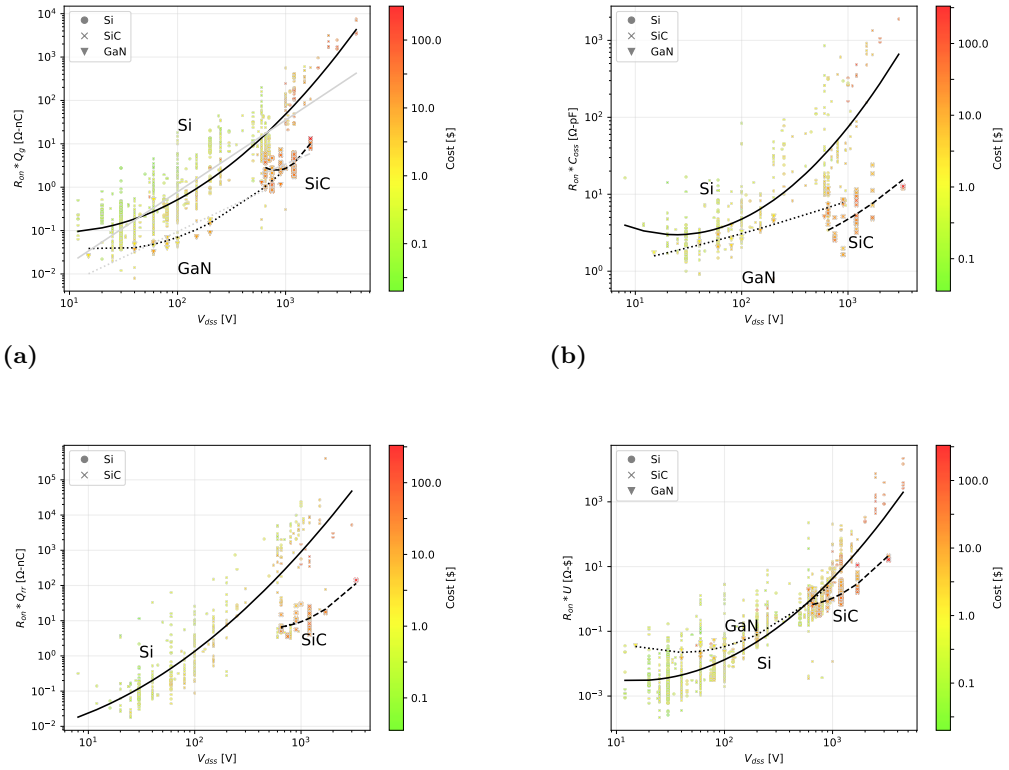


Figure 3.4: Transistor FOMs vs. voltage rating V_{dss}: (a) $\text{FOM}_g = R_{on}Q_g$, (b) $\text{FOM}_{oss} = R_{on}C_{oss}$, (c) $\text{FOM}_{rr} = R_{on}Q_{rr}$, and (d) $\text{FOM}_u = R_{on}U$, where U is a device cost. The FOM_g data is overlaid with linear and quadratic best-fit lines, while quadratic best-fit lines are shown for the other FOMs.

component parameters are complex and depend on converter topology and operating modes, *e.g.*, hard- versus soft-switching. This is why, in general, no single transistor FOM is sufficient to enable component selection in a converter design process. Four simple data-based FOMs are considered in this subsection, all as functions of the device voltage rating V_{dss}: $\text{FOM}_g = R_{on}Q_g$, $\text{FOM}_{oss} = R_{on}C_{oss}$, $\text{FOM}_{rr} = R_{on}Q_{rr}$, and $\text{FOM}_u = R_{on}U$, where U is a device cost.

All four data-based transistor FOM visualizations are presented in Fig. 3.4, where data points are shown color-coded by device cost. One may note the expected distributions between Si, SiC, and GaN devices. The trends can be captured by simple linear or quadratic polynomial fits on the log-log scale.

A linear fit on the log-log scale corresponds to the familiar FOM relationship,

$$\text{FOM} = 10^a x^b. \tag{3.2}$$

where $x = V_{dss}$ is expressed in Volts, b is the slope, and a is the y-intercept of the linear log-log polynomial fit. For example, for FOM_g, $b = 1.6$, corresponding to $R_{on}Q_g \sim V_{dss}^{1.6}$, which can be compared to physics-based expectation of $b = 1.5$ for the FOM expressed as the product of R_{on} and input capacitance [0].

One may further note that a quadratic relationship on the log-log scale may present a better fit to the data. For example, as seen in the FOM_g chart, the quadratic fit flattens at lower voltages, which is particularly visible for Si and even more so for GaN devices. This is due to confounding effects on the on-resistance, such as on-die semiconductor-to-metal vias and metal layers, die-to-pin resistances, and other factors that depart from the intrinsic device physics-based dependencies of R_{on} or Q_g on the rated voltage. The quadratic fit,

$$\text{FOM} = 10^a x^{b+c\log x}. \tag{3.3}$$

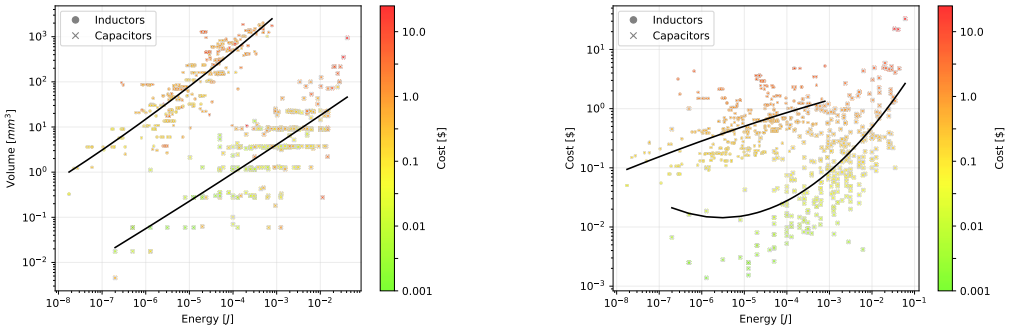
captures such subtler details in the FOM trends, where $x = V_{dss}$, and c, b, a are the second- to zeroeth-order coefficients, respectively.

A table of the polynomial fit coefficients ((b, a) for the linear fit and (c, b, a) for the quadratic fit) and the Relative Standard Error (RSE) of the linear and quadratic best-fits on the log-log scale for each data-based FOM are shown in Table 3.2. The RSE is calculated as $RSE = \sqrt{RSS/(n-2)}$, where n is the total number of data points and RSS is the residual sum of squares. On the log-log scale, the RSE represents an average fraction of an order of magnitude between the data points and the best-fit line.

We note that $\text{FOM}_u = R_{on}U$, where U is the device cost, cannot be found by physics-based models. GaN FETs, which outperform Si MOSFETs in terms of FOM_g and FOM_{oss}, trend higher in terms of the cost FOM_u, and the gap becomes larger at

Table 3.2: Coefficients and relative standard errors (RSE) for linear and quadratic relationships on the log-log scale between V_{dss} and various transistor FOMs.

		Si		SiC		GaN	
		Linear	Quadratic	Linear	Quadratic	Linear	Quadratic
$R_{on}Q_g$ [Ω·nC]	Coef.	(1.6,-3.4)	(0.59,-1.0,-0.61)	(1.2,-3.2)	(1.8,-9.7,14)	(1.3,-3.5)	(0.74,-2.0,0.042)
vs. V_{dss} [V]	RSE	0.48	0.45	0.22	0.21	0.34	0.29
$R_{on}C_{oss}$ [ps]	Coef.	(0.89,-0.94)	(0.54,-1.5,1.5)	(0.90,-2.0)	(0.33,-1.2,1.2)	(0.40,-0.31)	(0.04,0.23,-0.12)
vs. V_{dss} [V]	RSE	0.44	0.41	0.22	0.22	0.093	0.093
$R_{on}Q_{rr}$ [Ω·nC]	Coef.	(2.5,-4.8)	(0.53,0.15,-2.3)	(1.4,-3.3)	(1.9,-10,-15)	N/A	N/A
vs. V_{dss} [V]	RSE	0.78	0.77	0.25	0.24	N/A	N/A
$R_{on}U$ [Ω·$]	Coef.	(2.0,-5.7)	(0.95,-2.2,-1.2)	(1.8,-5.4)	(2.1,-11,15)	(1.4,-4.0)	(1.1,-3.5,1.1)
vs. V_{dss} [V]	RSE	0.62	0.56	0.24	0.22	0.33	0.19



(a) **(b)**

Figure 3.5: Inductor and capacitor FOM-related results: (a) volume vs. energy and (b) cost vs. energy. The data is overlaid with quadratic best-fit lines.

lower voltages. On the other hand, SiC MOSFETs, which are currently available at $V_{dss} \geq 600\,\text{V}$, outperform Si MOSFETs in all considered FOMs, and the gap becomes larger at higher voltages, which is consistent with expectations and findings in [0].

3.4.2 Inductors

Information was scraped on 48,000 commercially available fixed inductors. These inductors were then filtered to include those with inductances $\geq 10\,\text{nH}$ and dc resistances $\leq 1\,\Omega$. After filtering, outlier removal, and the Pareto front taken on the dimensions discussed in Section 3.3, 1,100 components remained. The considered inductor FOMs are as follows:

$$\text{Energy density} = \frac{\text{Energy}}{\text{Volume}} = \frac{0.5LI_{sat}^2}{AH} \tag{3.4}$$

$$\text{Cost per unit energy} = \frac{\text{Cost}}{\text{Energy}} = \frac{U}{0.5LI_{sat}^2} \tag{3.5}$$

$$\text{Loss per unit energy} = \frac{\text{Power loss}}{\text{Energy}} = \frac{R_{dc}I_{rated}^2}{0.5LI_{sat}^2} \tag{3.6}$$

The FOM-related graphs and the trends shown in Figs. 3.5 and 3.6 and Table 3.3 are consistent with physics-based expectations. Larger, higher-energy inductors exhibit

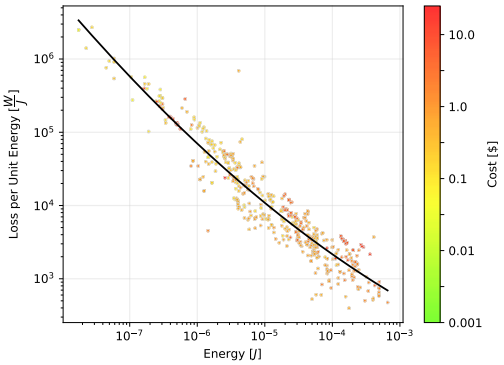


Figure 3.6: Inductor rated loss per unit energy storage capability vs. energy data, overlaid with a quadratic best-fit line.

Table 3.3: Coefficients and RSE for linear and quadratic best-fit lines and curves on inductor data.

		Linear	Quadratic
Vol. [mm^3] vs. Energy [J]	Coef.	(0.74,5.6)	(0.022,0.97,6.2)
	RSE	0.41	0.40
Cost [$] vs. Energy [J]	Coef.	(0.24,0.91)	(-0.0088,0.15,0.68)
	RSE	0.36	0.36
Loss [W/J] vs. Energy [J]	Coef.	(-0.77,0.23)	(0.054,-0.22,1.6)
	RSE	0.24	0.23

higher energy density and lower relative loss [0]. In Fig. 3.5(a), the slope $b < 1$ confirms that energy density increases for larger components. Furthermore, one may observe that the cost increases with increased energy, but also at a rate $b < 1$.

3.4.3 Capacitors

Information on 110,000 Class II ceramic capacitors (temperature coefficients X5R, X7R, X8R, X7S, and Y5V) was scraped and analyzed. Capacitors were filtered to only include capacitance $\geq 10\,\text{nF}$, voltage rating $\leq 1\,\text{kV}$, and MLC (surface-mount) package type. After filtering, outlier removal, and taking the Pareto front as discussed in Section 3.3, 2,800 components remained. The considered FOMs are:

$$\text{Energy density} = \frac{\text{Energy}}{\text{Volume}} = \frac{0.5CV_{rated}^2}{AH} \tag{3.7}$$

$$\text{Cost per unit energy} = \frac{\text{Cost}}{\text{Energy}} = \frac{U}{0.5CV_{rated}^2} \tag{3.8}$$

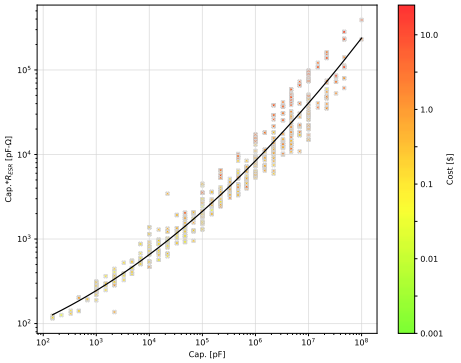


Figure 3.7: Capacitor $R_{ESR}C$ vs. capacitance C data, overlaid with a quadratic best-fit line.

The FOM-related plots of the data shown in Fig. 3.5 reveal that capacitors can offer substantially higher energy density and lower cost per unit energy compared to inductors, which is consistent with expectations [0]. One may also note that dispersal among specific components is relatively large, meaning the linear and quadratic curve fits, as summarized in Table 3.4, represent trends with notable variances.

A separate dataset was used to generate a relationship between the capacitor time constant CR_{ESR} and capacitance C, because equivalent series resistance R_{ESR} values are not readily available from most component datasheets. An equivalent circuit library database is available from TDK capacitors containing information on 663 Class II ceramic capacitors, and additional datapoints were collected from the impedance plots of other manufacturer datasheets, including Kyocera AVX. The time-constant plot is shown in Fig. 3.7. The relationship between CR_{ESR} and C (where (b,a) are the first- and zeroeth-order coefficients, respectively):

$$CR_{ESR} = 10^a C^b \tag{3.9}$$

can be used in the converter design process to estimate voltage ripple or power loss associated with the capacitors.

Table 3.4: Coefficients and RSE for linear and quadratic best-fit lines and curves on capacitor data.

		Linear	Quadratic
Vol. [mm^3]	Coef.	(0.63,2.5)	(0.0064,0.69,2.6)
vs. Energy [J]	RSE	0.52	0.52
Cost [$]	Coef.	(0.49,0.53)	(0.12,1.4,1.9)
vs. Energy [J]	RSE	0.65	0.62
$R_{ESR}C$ [ps]	Coef.	(0.57,0.52)	(0.04,0.13,1.6)
vs. C [pF]	RSE	0.19	0.17

3.5 Conclusions

This chapter discusses the process of acquiring the component database by scraping information off of distributor websites and the PDF datasheet links. Gathering component parameters of real, commercially available parts allowed for the development of a large, high-quality dataset that was used to train ML models and perform an exhaustive search. This chapter presents various preprocessing techniques to clean the scraped data and retain components most likely selected by power electronics designers. Examples of data-driven component figures of merit (FOMs) are presented to show how the collected datasets can be used to capture trends in component performance metrics. Overall, the goal of having a component database is to allow machines to augment some of the manual work otherwise performed by hardware designers and reduce cost and time spent during the design process.

Chapter 4

Design-oriented power loss modeling

This chapter examines the development of a power loss model that can be computed knowing only datasheet parameters, converter operating conditions, and steady-state solutions. The equations incorporate design-oriented parameters based on background knowledge of device physics. The purpose of developing a power loss model is so that expected power losses can be automatically computed at each design iteration within the optimization algorithm and given each component combination in the exhaustive search step. An accurate power loss prediction is critical to determining which parameter values will lead to the most efficient converter and therefore the best design. It is worth noting that these are all estimation methods, and require some approximations and assumptions. Even given a hardware prototype, it is challenging to determine the losses associated with each individual loss contribution. Therefore, this work aims to develop estimation techniques for the most significant loss contributions.

The specific design-oriented parameters are discussed in Section 4.1. This section discusses physics- and data-based approaches to the estimation of several design-oriented parameters: transistor on-resistance R_{on} as a function of junction temperature T_j,

diode reverse recovery parameters τ_c, τ_{rr}, charge-equivalent capacitance $C_{ds,q}$, dc bias voltage-adjusted capacitance of ceramic capacitors C_{Vdc}, and inductor ac loss parameters $f_b, b, K_{fe}, \alpha, \beta, N_{turns}, A_c, V_{core}$. Each approach is validated on a selection of out-of-sample components. Section 4.2 discusses how each modeling method is incorporated into the design tool, and Section 4.3 concludes this chapter.

Note how these design-oriented parameters can be used throughout the power loss model of any converter topology. As an example, the power loss model for a hard-switched buck converter using these design-oriented parameters is discussed in Subsection 7.1.2.

4.1 Estimation of Design-oriented Parameters

This section discusses methods for developing the aforementioned design-oriented parameters in ways that can be applied in an automated manner to large datasets of scraped datasheet parameters.

4.1.1 R_{on} vs. T_j dependence

The maximum on-resistance of the transistor is reported on the main page of distributor sites, but usually only at an operating temperature of $T_j = 25°C$. To better represent practical operating conditions, where the transistor typically reaches an elevated temperature of, for example, $T_j = 100°C$, this section aims to determine a multiplier k_T for R_{on} based on the device voltage rating V_{dss}. The reported value of R_{on} is then multiplied by

$$k_T = \frac{R_{on}(T_j = 100°C)}{R_{on}(T_j = 25°C)} \tag{4.1}$$

The transistors are split into five groupings; low-voltage ($\leq 200\,\text{V}$) and high-voltage ($> 200\,\text{V}$) silicon (Si), low-voltage ($\leq 100\,\text{V}$) and high-voltage ($> 100\,\text{V}$) gallium nitride (GaN), and silicon carbide (SiC). These groupings were determined by examination of the k_T vs. V_{dss} plots of each device type, an example is shown in Fig. 4.1. The collection of this plot data shows clear slope changes at 200 V and 100 V for Si and GaN,

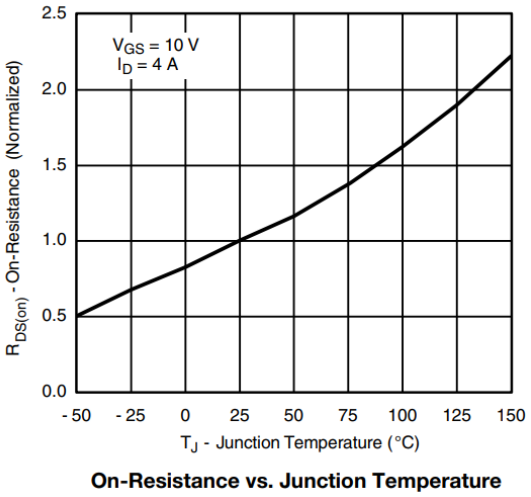


Figure 4.1: An example plot of R_{on} vs. T_j, copied from the datasheet of MOSFET SI7450DP [0]. In some datasheets such as the one presented here, R_{on} is normalized with respect to $25°C$, whereas in other datasheets the raw R_{on} data values are used.

respectively. The R_{on} vs. T_j plots included in the datasheets of ~ 15 components for each grouping were manually digitized. The components were randomly selected but intentionally spanned a range of voltage ratings and device manufacturers. From the manually digitized graphs, the multiplier k_T at $100°C$ was recorded. As an example, the plot of k_T vs. V_{dss} is shown in Fig. 4.2 for low-voltage Si devices, where it can be seen that the multiplier k_T can be approximated by a linear relationship with $\log_{10} V_{dss}$, where V_{dss} is expressed in Volts.

Table 4.1 shows a summary of the linear relationships and datasheet-validated performance results, based on the mean absolute percentage error (MAPE) between 5-10 k_T values from capacitors hidden during training, and their respective datasheet-calculated k_T values.

4.1.2 Diode reverse-recovery

In a hard-switched converter, losses due to diode or body-diode reverse recovery can be very significant. This switching loss P_{Qrr} is modeled using the reverse recovery charge Q_{rr} and the reverse recovery time t_{rr}. Datasheets usually report only one set of (Q_{rr}, t_{rr}) values at a specific operating point and under specific test conditions characterized

Table 4.1: R_{on} temperature multiplier $k_T(V_{dss})$ for each of the five transistor groupings, determined by the linear best-fit line of the k_T vs. V_{dss} [V] data based on manually-digitized $R_{on}(T_j)$ plots for a random selection of components in each grouping.

Grouping	$k_T(V_{dss})$	MAE
Low-voltage Si	$0.233 \log_{10} V_{dss} + 1.15$	0.125
High-voltage Si	$0.353 \log_{10} V_{dss} + 0.83$	0.094
Low-voltage GaN	$0.066 \log_{10} V_{dss} + 1.34$	0.0493
High-voltage GaN	$0.148 \log_{10} V_{dss} + 1.15$	0.0576
SiC	$0.572 \log_{10} V_{dss} - 0.523$	0.194

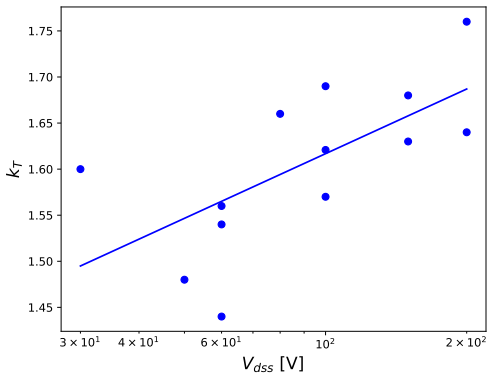


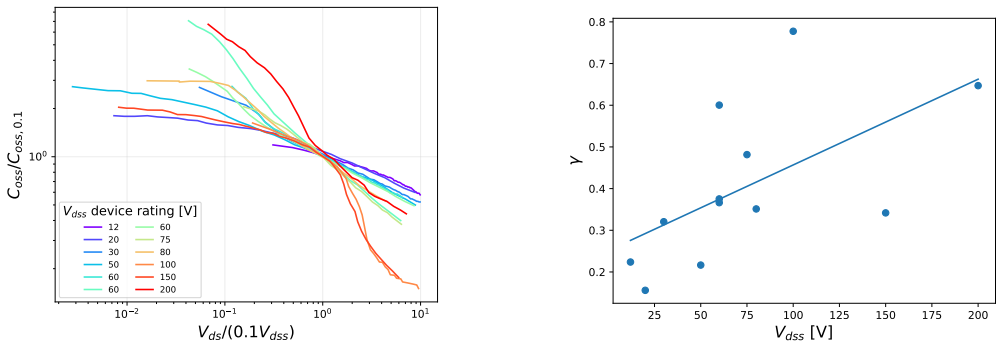
Figure 4.2: R_{on} temperature multiplier k_T for a randomly selected assortment of low voltage, Si MOSFETs from a variety of manufacturers. k_T is plotted vs. V_{dss} to determine the linear correlation reported in Table 4.1.

by the diode current I_F prior to turn-off, and the slope $\frac{di_F}{dt}$ of current decay. As summarized in Subsection 7.1.2, the device physics-based approach discussed in [0, 0] can be used to determine the reverse recovery time constants τ_c and τ_{rr}, which then allow one to estimate Q_{rr}, t_{rr} and P_{Qrr} under the operating conditions considered in the design.

4.1.3 Charge-equivalent capacitance

In a hard-switched half-bridge stage, the energy loss due to the device output capacitance C_{oss} can be found using the charge-equivalent capacitance $C_{ds,Q}$ given by

$$C_{ds,Q} = \frac{1}{V_{in}} \int_0^{V_{in}} C_{oss}(V_{ds})\, dV_{ds}\,, \tag{4.2}$$



(a) **(b)**

Figure 4.3: (a) Normalized C_{oss} vs. V_{ds} curves for a randomly selected assortment of low voltage, Si MOSFETs from a variety of manufacturers. The V_{ds} values are normalized by $0.1V_{dss}$, and the C_{oss} values are normalized by $C_{oss,0.1} = C_{oss}(0.1V_{dss})$. (b) The γ exponent in Eq. (4.3) of each component, plotted vs. V_{dss} to determine the linear correlation Eq. (4.4).

where V_{in} is the voltage to which the device is exposed during the circuit off-state $[0, 0, 0]$. As shown in Eq.(4.2), $C_{ds,Q}$ requires C_{oss} as a function of drain-to-source voltage V_{ds}. MOSFET datasheets typically include $C_{oss}(V_{ds})$ in graphical form, as well as C_{oss} measured at a particular voltage $V_{ds,meas}$. Unfortunately, $V_{ds,meas}$ varies drastically between datasheets, implying inconsistent C_{oss} comparisons. For any given device, $C_{ds,Q}$ can be found by digitizing the $C_{oss}(V_{ds})$ graph, but it is difficult to automate this process on large MOSFET datasets. Due to this limitation, the approach taken here aims to determine the $C_{oss}(V_{ds})$ curve for all components by combining the machine-scraped single reported values (C_{oss}, $V_{ds,meas}$) with trends determined by manual examination of select $C_{oss}(V_{dss})$ curves.

The MOSFETs are once again divided into five categories: low-voltage Si, high-voltage Si, SiC, low-voltage GaN, and high-voltage GaN. For each group, the C_{oss} vs. V_{ds} plots for a select number of components with varying voltage ratings and device manufacturers are collected and digitized into a machine-readable format. The plots confirmed that the same V_{dss} ranges are appropriate for Si and GaN devices as those for k_T. These datapoints are then normalized by $0.1V_{dss}$ and $C_{oss,0.1} = C_{oss}(0.1V_{dss})$ as found from

Table 4.2: $\gamma(V_{dss})$ for each of the five groupings of transistors, along with the respective MAPE comparing the predicted γ with the γ manually obtained from the datasheet for a selection of out-of-sample components. Note that the slopes γ for SiC components can be well approximated by a constant mean value.

Grouping	$\gamma(V_{dss})$	MAE
Low-voltage Si	$0.0021\,V_{dss} + 0.251$	0.042
High-voltage Si	$0.00089\,V_{dss} + 0.427$	0.20
Low-voltage GaN	$0.00062\,V_{dss} + 0.355$	0.037
High-voltage GaN	$0.00039\,V_{dss} + 0.353$	0.083
SiC	0.451	0.083

the digitized C_{oss} vs. V_{ds} curve. As an example, Fig. 4.3(a) shows the normalized C_{oss} plots collected for low-voltage Si components, revealing an approximately linear trend on the log-log scales,

$$C_{oss} \approx C_{oss,0.1} \left(\frac{0.1 V_{dss}}{V_{ds}} \right)^{\gamma} . \tag{4.3}$$

By plotting the slope γ as a function of V_{dss} as seen in Fig. 4.3(b), the approximate linear fit for $\gamma(V_{dss})$ can be determined,

$$\gamma = \gamma_1 V_{dss} + \gamma_0 . \tag{4.4}$$

Given Eq.(4.4), V_{dss} of the component, and the machine-scraped (C_{oss}, $V_{ds,meas}$) values, Eq.(4.3) can be used to determine $C_{oss,0.1}$ and then calculate the charge equivalent capacitance using Eq.(4.2), without the need to digitize all C_{oss} versus V_{ds} plots.

Similarly to the validation approach described in Subsection 4.1.1, components not used to generate the $\gamma(V_{dss})$ equations are selected from each of the five transistor groups. First, the V_{dss} of the component and the corresponding $\gamma(V_{dss})$, as shown in Table 4.2, are used to calculate γ. In comparison, the true γ is determined by finding the slope of the manually digitized C_{oss} vs. V_{ds} plot reported in the datasheet. The MAE values between the predicted and reported values of γ are shown in Table 4.2 for each of the component groups, showing that Eqs.(4.3) and (4.4) offer a good prediction of $C_{oss}(V_{ds})$ and thereby of $C_{ds,Q}$ in Eq.(4.2).

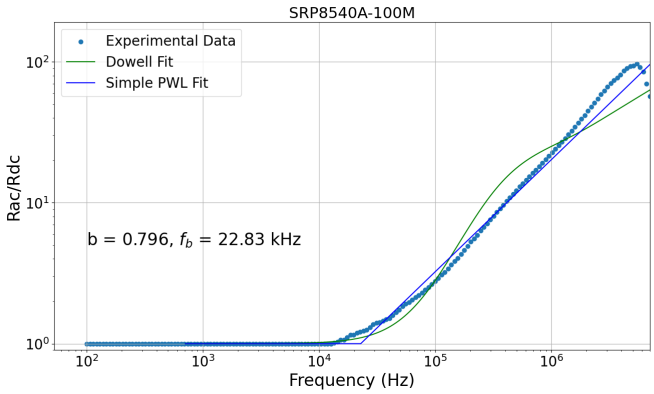


Figure 4.4: R_{ac}/R_{dc} estimation results for SRP8540A-100M inductor, compared to Bode 100 measurements.

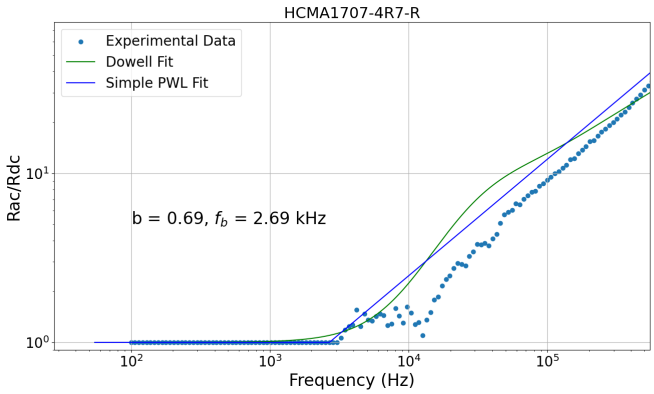


Figure 4.5: R_{ac}/R_{dc} estimation results for HCMA1707-4R7-R inductor, compared to Bode 100 measurements.

4.1.4 Inductor ac loss parameters

Datasheet parameters for fixed inductors typically include the nominal inductance L, inductance at saturation current L_{sat} @ I_{sat}, dc resistance R_{dc}, rms current rating at certain temperature rise I_{rms} @ ΔT, geometrical parameters of the package, and a core-material designation. Datasheets may also include the Q-factor at a particular frequency Q @ f_Q and the self-resonant frequency f_{res}. Unfortunately, the datasheet parameters are insufficient to accurately predict ac winding or core losses when inductors are used in switched-mode converters.

The goal of the approach presented in this subsection is to estimate the parameters

required for the calculation of ac winding losses and core losses in off-the-shelf inductors. The approach currently focuses on commercially available surface-mounted metal-powder-core fixed inductors.

Starting from the geometric parameters of a component, the ac loss parameter estimation algorithm iterates over the number of turns N_{turns}, adjusting the wire cross-sectional area to meet the datasheet value of R_{dc}. Next, the algorithm estimates the number of winding layers to fit the winding into the available window area and the relative permeability of the material necessary to obtain the datasheet value of inductance L. In each iteration, Dowell's method [0, Section 10.4] is used to calculate the Q-factor of the inductor. Assuming approximately equal contributions of winding loss and core loss to the inductor Q-factor, iteration over N_{turns} stops when the estimated Q is closest to the datasheet value. Unfortunately, for many inductors, the Q-factor is not reported in the datasheets. This challenge is addressed by a simple linear correlation using the core volume V_c of the component,

$$Q \,@\, 100\,\text{kHz} \approx 0.0052\; V_c[\text{mm}^3] + 12. \tag{4.5}$$

The correlation in Eq. (4.5) is based on data collected from a subset of inductors where Q is reported. Larger inductors generally having a larger Q-factor is consistent with the analysis in [0] and the data-based results reported in [0].

Once the number of turns N_{turns}, the wire size, and the number of layers are determined, the estimated R_{ac}/R_{dc} follows using Dowell's method [0, Section 10.4]. To further simplify loss modeling, a piecewise fit on the log-log scales,

$$R_{ac}(f)/R_{dc} \approx \max\Big(1, (f/f_b)^b\Big), \tag{4.6}$$

is adopted so that ac winding losses can be found for an arbitrary inductor current waveshape using only two design-oriented parameters: the break frequency f_b and the exponent b. Finally, the Steinmetz core-loss parameters K, α, and β are adopted

from [0] for the metal powder material with μ_r closest to the estimated value for use in core loss estimation.

To validate the approach, the ac loss parameter estimation method has been automated and applied to a selection of 14 surface-mount powder-core fixed inductors from various manufacturers. The winding-loss results are summarized in Table 4.3, which uses the loss model equations from Table 7.2 in Subsection 7.1.2: $P_{Lw,dc}$ using R_{dc}, P_{Lw} using the PWL fit Eq. (4.6) for R_{ac}, and the losses $P_{Lw,m}$ obtained with R_{ac}/R_{dc} measured experimentally using a Bode 100 network analyzer.

The winding losses are calculated for a representative inductor current waveform that has a dc component $I_{dc} = I_{sat}/2$ and a triangular waveform with 100% ripple magnitude $\Delta i = I_{sat}/2$ at a switching frequency f_s, where

$$f_s = \max(5f_b, 20\,\text{kHz})\,, \tag{4.7}$$

and f_b is the frequency at which measured R_{ac}/R_{dc} exceeds 1.1. The results in Table 4.3 show that the basic model $P_{Lw,dc}$ always underestimates the winding loss. Despite some notable errors compared to $P_{Lw,m}$, which can be attributed to various assumptions the estimation algorithm must make, the improved estimate P_{Lw} mitigates the underestimation and provides a more accurate representation of the inductor winding loss.

Figs. 4.4 and 4.5 more closely examine the estimation results verified by R_{ac} measurements using the Bode 100 impedance analyzer for two example inductors. Fig. 4.4 visually shows a case where the predicted fit closely matches the measurements. Fig. 4.5, on the other hand, shows a case where the fit is not as close to the measurements. Table 4.3 reports that in both cases the inductor ac loss model overestimates the losses compared to the losses predicted by the Bode 100 measurements of R_{ac}. In the case of Fig. 4.4, due to the better fit, the percent error in the loss estimation is smaller than for the inductor considered in Fig. 4.5.

Table 4.3: Results of R_{ac} estimation for selected fixed inductors, where the inductor part numbers by index are as follows: (1: SRP1038A-2R2M, 2: SRP1265A-5R6M, 3: SRP8540A-100M, 4: 74437368100, 5: 744766904, 6: SRP4020FA-1R0M, 7: IHLP2525CZER6R8M11, 8: LMLP13B3M1R2DTAS, 9: HCMA1707-4R7-R, 10: DFEH7030D-2R2M=P3, 11: HCM0503-1R0-R, 12: SRP2313AA-100M, 13: 744373965010, 14: XAL1010-102MED).

Inductor index	L [µH]	I_{rms} [A]	$P_{Lw,dc}$ [W]	$P_{Lw,m}$ [W]	P_{Lw} [W]
1	2.2	12.0	1.58	2.13	1.78
2	5.6	12.5	1.50	1.89	2.32
3	10.0	5.6	1.77	2.47	3.00
4	10.0	7.0	2.54	3.04	2.71
5	4.7	1.2	0.10	0.12	.10
6	1.0	11.0	0.10	0.16	.16
7	6.8	5.5	0.24	0.34	.39
8	1.2	30.0	1.61	1.64	2.78
9	4.7	27.0	0.79	1.15	1.68
10	2.2	7.1	0.29	0.33	0.30
11	1.0	10.1	0.25	0.45	0.45
12	10.0	30.0	0.51	1.05	1.71
13	1.0	23.0	2.27	2.30	3.44
14	1.0	43.5	1.01	3.98	1.31

4.1.5 Capacitor nominal capacitance based on V_{dc}

For Class II MLC capacitors, the capacitance strongly depends on the applied dc bias voltage V_{dc} because of the effects of the change in permittivity of the material. Although ΔC vs. V_{dc} graphs are available in a variety of formats, these are extremely difficult to gather in large quantities in an automated manner. The process is even more difficult because of the inconsistency with where this information is provided, such as manufacturer datasheets, online tools, or separate characterization sheets. For example, Samsung reports the ΔC vs. V_{dc} graphs on their component datasheets, whereas Kemet reports the graphs on their KSIM tool, and TDK uses separate characterization sheets. Fig. 4.6 shows an example of the ΔC vs. V_{dc} plot from a TDK capacitor characterization sheet.

This work aims to gather data and capture trends for capacitance versus dc bias so that the correct capacitance is selected to ensure the voltage ripple requirements are met at V_{dc}. For a buck converter example, one may want to calculate the required capacitance

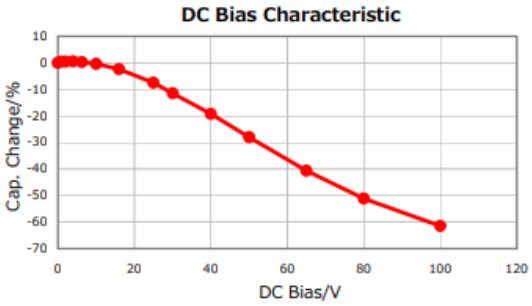


Figure 4.6: An example of a TDK capacitor characterization sheet showing ΔC vs. V_{dc}.

for C_1 (input capacitance) and C_2 (output capacitance) at a given V_{dc}, according to the following standard set of equations:

$$\Delta v_{C1} = \frac{I_{out}}{2C_1 f_s} D(1-D) \tag{4.8}$$

where

$$\frac{\Delta v_{C1}}{V_g} \leq 5\%, \tag{4.9}$$

and

$$\Delta v_{C2} = \frac{\Delta i_L T_s}{8C_2} \tag{4.10}$$

where

$$\frac{\Delta v_{C2}}{V_{out}} \leq 1\% \tag{4.11}$$

at the given point of dc operation: $V_{dc,C1} = V_g$ and $V_{dc,C2} = V_{out}$. This will allow the proper capacitance to be selected to ensure the voltage ripple requirements are met at V_{dc}.

In this work, data-based techniques are used to automate the capacitance versus dc bias characterization across the dataset of Class II MLC capacitors. Using online graph-to-data digitization tools, the graph data of ΔC versus V_{dc} were collected for 50 Class II MLC capacitors from a variety of manufacturers, case sizes, voltage ratings, and capacitance.

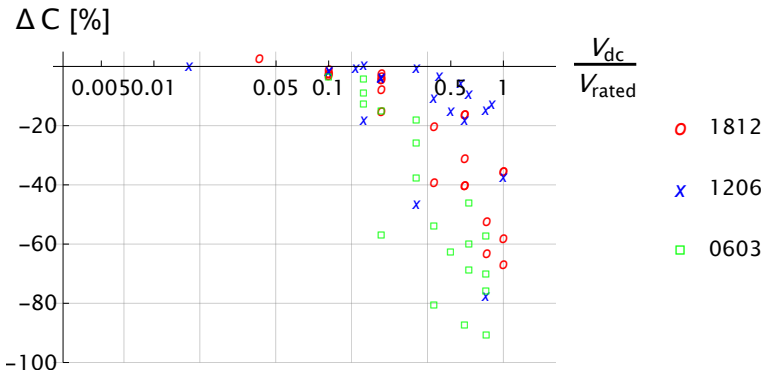


Figure 4.7: ΔC vs. $\frac{V_{dc}}{V_{rated}}$ for a variety of Class II MLC capacitors.

An approximately linear relationship can be observed on the log-linear plot of ΔC vs. $\frac{V_{dc}}{V_{rated}}$ in Fig. 4.7. These datapoints were used to train a linear regression ML model to find the expected capacitance C_{0Vdc} at zero dc bias. The known quantities in the design are V_{dc}, which is determined by the voltage seen across the capacitor, and C_{Vdc}, which can be calculated using the standard set of steady-state equations. C_{Vdc} and V_{dc} are used along with scraped quantities V_{rated}, A as inputs to predict output C_{0Vdc}. The MAPE was found to be 14.5%. The predicted output value of C_{0Vdc} is considered the minimum nominal capacitance. C_{0Vdc} is the quantity scraped via automation and can then be used as an input for predictions of other capacitor parameters such as U.

4.2 Incorporation of modeling methods into the tool

This section describes how the data- and physics-based modeling methods and corresponding trained ML models are incorporated into the tool, as shown in Fig. 2.1, with the goal of improving the quality and accuracy of the parameters of the recommended converter components returned by the optimization tool.

The multiplier k_T for scaling R_{on} with respect to T_j is incorporated by using the specified V_{dss} of each transistor in the design and the linear relationships in Table 4.1 to compute each k_T multiplier. To incorporate reverse recovery charge losses, quantities τ_c and τ_{rr} are predicted using ML models trained on other known transistor quantities.

To find the I_F of operation, for the buck converter example the design specification I_{out} and optimization variable Δi_L are known at each instance of the optimization, and can then be used to calculated I_F. To incorporate $C_{oss,0.1}$, the $C_{oss,0.1}$ value is computed for every component in the dataset and used as the output of ML model training. Eq. (4.2) is then used to calculate $C_{ds,Q}$ given V_{in} and the required V_{dss} according to the converter design specifications. To incorporate the inductor ac loss parameters, all the following quantities are predicted using trained ML models: f_b, b, K_{fe}, α, β, N_{turns}, and A_c. Winding loss P_{Lw} considers the first 11 harmonics of the inductor current waveform as described in Subsection 7.1.2, where R_{ac} is estimated using Eq. (4.6) with ML-predicted values of f_b and b. The core loss P_{Lc} is incorporated using the iGSE approach [0] with the predicted values of the core loss parameters. Capacitor dc bias voltage-adjusted capacitance is incorporated by computing C_{Vdc} given the design specifications, and then using C_{Vdc}, the optimization variable Δi_L, and additional known design parameters to make a prediction of C_{0Vdc}.

At this point, all quantities have been computed and predicted to compute the equations in Table 7.2.

4.3 Conclusions

This chapter discusses the development of design-oriented parameters used to more accurately model the main power loss contributions of components. Power loss estimation methods are presented to model losses due to temperature dependence, reverse recovery charge, charge-equivalent capacitance, nominal capacitance based on V_{dc}, and inductor ac losses. These loss mechanisms are used within the tool to predict the power losses given the predicted component parameters at each optimization algorithm iteration. These design-oriented parameters are also used when computing the power loss using the scraped datasheet parameters given a specific component combination within the exhaustive search. The individual component losses are combined to compute the overall converter power loss.

Chapter 5

Machine learning models

This chapter examines the development of machine learning (ML) models trained on component data for use in the context of the presented design approach. ML models are trained algorithms that use inputs to automatically predict outputs. ML models capture trends of large quantities of data and need only to be trained once. Existing Python packages such as Scikit-learn allow for a fairly straightforward implementation of ML models. The model hyperparameters are selected to improve predictive accuracy.

ML models are developed for each type of component with scraped datasets (transistors, inductors, and capacitors). Fig. 2.1 shows how after one-time training, ML-based component models m_Q, m_L, m_C are used within the optimization loop. These models are useful for the design approach because they can predict values of component attributes given a certain set of known design inputs. The attributes can then be used by the optimization tool to compute the total power loss, cost, and area of the components on the PCB. After training the ML models, testing data that was unseen during the training process is used to evaluate each model's predictive accuracy.

This chapter is organized as follows: Section 5.1 discusses the problem formulation used to structure the ML models in the context of the tool and shows the evaluation of the models based on a performance metric, Section 5.2 discusses which ML models

were selected for this application and why, and Section 5.3 investigates the limitations of using ML models within the optimization approach. Finally, Section 5.4 concludes this chapter.

5.1 ML model structure and performance evaluation

The pre-processed datasets[1] acquired as described in Chapter 3 are used to train the ML models. As an illustration, Fig. 3.3 shows Linear and a Random Forest regressions, respectively, overlaid on the training set of Pareto front datapoints. As a result of using the Pareto front of the dataset for training and testing, the data variance is reduced and the regression will predict values closer to the best envelope. Fig. 3.3(a) shows how the inductor volume versus energy rating can be represented by a linear log-log relationship, whereas Fig. 3.3(b) illustrates how a more complex regression can be used to capture characteristics of available components.

In general, as shown in Eq. (2.2), the known inputs to the developed ML models are a combination of component specifications $\bar{T}$ based on the converter design and variables $\bar{O}$ over which the multivariate optimization algorithm iterates. The ML models need to predict the quantities $\bar{X}$ used to evaluate power loss models (such as the loss model discussed in Subsection 7.1.2), as well as the BOM cost U and footprint area A contributions of the considered converter transistors, inductors, and capacitors. The input/output structure of the developed ML models is summarized in Table 5.1.

Listing 5.1: Example implementation of a Kernel Ridge regression ML model trained and evaluated on component datasets.

```
cv = KFold(n_splits=5, random_state=1, shuffle=True)
model = KernelRidge(degree=degree, alpha=alpha, gamma=gamma,
                    kernel='rbf')
chain = RegressorChain(base_estimator=model, order=order)
scores = cross_val_score(chain, X, Y, cv=cv,
        scoring='mean_absolute_percentage_error')
```

[1] Datasets were last scraped on September 13, 2023.

Component	Inputs to ML model		Outputs $\bar{X}$	MAPE [%]
	$\bar{O}$	$\bar{T}$		
Q_i (distributor site)	$\{R_{on,i}\}$	$\{V_{dss,i}$, N-channel, FET technology$\}$	$\{Q_{g,i}, U_i, A_i\}$	{43, 27, 347}
Q_i (datasheet)	$\{R_{on,i}\}$	$\{V_{dss,i}, U_i, A_i,$ N-channel, FET technology$\}$	$\{C_{oss,0.1,i}, V_{ds,meas,i}, \tau_{c,i}, \tau_{rr,i}\}$	{34, 17, 40, 58}
L_i	$\{f_{sw}, \Delta i_L\}$	$\{I_{rated}, I_{saturation}, L\}$	$\{R_{dc}, U, A, f_b, b, K_{fe}, \alpha, \beta, N_{turns}, A_c, V, V_{core}\}$	{27, 47, 24, 71, 10, 36, 3, 0.6, 26, 23, 33, 26}
C_i	$\{A_{Ci}\}$	$\{V_{rated,i}, C_i\}$	$\{U_i\}$	{84}

Table 5.1: ML model structure and training results on the transistor, inductor, and capacitor datasets given inputs $\bar{O}, \bar{T}$ and outputs $\bar{X}$. Note that the transistor models are split into outputs $\bar{X}$ gathered from distributor site parameters and from PDF datasheet parameters. The models are trained using the Kernel Ridge regression algorithm. The mean absolute percentage error (MAPE) compares the model predictions to the corresponding parameter values for components in the testing dataset.

The ML models were trained on the training data and evaluated on unseen testing data. The hyperparameters for the selected regression model were tuned using "RandomizedSearchCV", an automatic hyperparameter tuner in the Scikit-Learn library [0]. This tuner iterates over potential hyperparameter values in order to minimize the root-mean-squared-error score on the validation data and attempt to find the optimal bias-variance trade-off point. The code for implementing the training and scoring is shown in Listing 5.1. Table 5.1 shows the MAPE performance results on the testing data for transistors, inductors, and capacitors. Each model took less than a few minutes to train and test using cross-validation. The component cost models are based on low-volume, single-component distributor quotes. The testing results show that some values are well-predicted, while others show significant differences between the predicted and true values. These differences are expected given knowledge about the dataset as discussed in Section 5.3, and are accounted for by incorporating the component selection process addressed in Chapter 6.

5.2 Selected models for the optimization tool

Various factors are considered when selecting an ML regression algorithm for the tool: performance, run-time, application, and complexity. There was minimal variation in performance accuracy between various regression models. All models were implemented via the Scikit-learn Python package, and therefore were similarly easy to implement. Linear regression was computationally the simplest and fastest model, and Random Forest had slightly higher accuracy but longer run-time compared to other models. Among many available regression algorithms, Kernel Ridge regression was selected because the results had comparatively high predictive accuracy and low run-times.

Kernel Ridge regression is a multivariate method that uses L2-norm regularization to account for multicollinearity by reducing the maximum size of the coefficients of the linear equation. Unlike Lasso regression, Ridge regression does not force coefficients to zero. In the training process, the algorithm uses the residual sum of squares (RSS) to

determine how close a prediction is from the true value. The RSS equation including the regularization term is shown below:

$$RSS = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + \lambda \sum_{j=1}^{P} B_j^2 \tag{5.1}$$

where Y_i is the true value, $\hat{Y}_i$ is the predicted value, λ is the L2 normalization factor, and B_j are the model coefficients. This algorithm is useful for datasets that are more likely to experience overfitting issues.

The kernel trick allows the model to transform data to higher dimensions, which can benefit the model when finding best-fit spaces. The kernel trick and Ridge regression form Kernel Ridge regression. The hyperparameters for Kernel Ridge regression are the kernel type, γ, degree, and α, which describe the internal kernel mapping, degree of the polynomial, and regularization strength. Unlike other regression algorithms such as K-Nearest-Neighbors regression, Kernel Ridge regression also offers efficient prediction times, which is critical for the optimization tool, which must run for many iterations.

The one-time trained ML models are then used to capture the trends in component parameter relationships so that the tool can make estimates on the predicted performance at any given optimization iteration, as discussed in Chapter 2.

5.3 Model limitations

The results shown in Fig. 3.3 and Table 5.1 are illustrative of ML-based device model limitations. One limitation arises from the inherent nature of the data itself and the limited set of known input quantities $\bar{T}, \bar{O}$, which results in a high standard deviation in the output even given the exact same input values. This concept is reflected in the performance scores when evaluated on the training data itself, which were similar to those of the testing data shown in Table 5.1. As an illustration of why this phenomenon is to be expected, consider an example of N-channel Si MOSFETs. Fig. 3.3(b) shows an example where V_{dss} is the only known input, and it can be seen that the standard

deviation in output $R_{on} * U$ is high for some V_{dss} values. To expand this concept, given an initial value of the optimization variable R_{on} and the known required voltage rating V_{dss}, one can expect that there will be some significant variation in the predicted quantity, such as gate charge Q_g. All three quantities can then be used to predict the cost U, but even if Q_g is perfectly known rather than predicted, the cost will vary significantly among components in the transistor dataset.

Slight adjustments can also lead to significant variation in $\bar{X}$ as modeled by the regression best-fit line in Fig. 3.3(b), where the best-fit line jumps around given small changes in V_{dss}. Similarly, consider again the N-channel Si MOSFET example where V_{dss} is known. The captured average Q_g values may vary non-linearly across increasing R_{on} values, but capturing each deviation may not translate well to new datapoints. This illustrates the challenge of high variation, where the model overfits the data. The Kernel Ridge regression algorithm is formulated to account for this variation.

As shown in Table 5.1, the prediction error for the transistor footprint area is very large, which is to be expected given the weak correlation between the package and the electrical parameters of the component. The standard deviation in package area for a given set of inputs is expected to be very high, even when using a more robust set of inputs.

It is important to note that the inherent variability and high standard deviation are both expected and accounted for in the context of the design approach presented in this thesis. The *trends* among component parameters are captured by ML models well enough such that the optimization algorithm can identify a representative set of optimized component parameters. These results are then used to substantially filter the component datasets to much smaller subsets such that combinations of components can be examined in a tractable manner.

One limitation to note is the dynamic nature of component prices and availability. This can be mitigated, in part, by periodically updating the datasets and retraining

the models. However if many components, *e.g.*, increase in price without updating the datasets, the ML models will not represent the current cost trends well.

Another significant limitation relates to the quality and quantity of component data available for training. Component datasheets have traditionally been tailored towards design engineers. While datasheets contain extensive data, the data is often insufficient and is not readily machine-readable. This presents substantial challenges, as discussed in Chapters 3 and 4. It is anticipated that advances by manufacturers, distributors, and CAD-tool vendors, in combination with open-source initiatives, such as [0, 0, 0], will lead to further improvements in the quality and quantity of the component data available for the developments of data-driven design approaches such as the approach presented in this work.

5.4 Conclusions

This chapter examined the development of ML models for use in the presented design approach. This chapter analyzed the testing results after training on the scraped component data and discussed how the models are used within the developed tool. The limitations of ML models are also presented, revealing the benefits of improved, expanded machine-readable datasets for use in automated tools. Some of the errors associated with the predictive accuracy of ML models are mitigated by the presented component selection method, as discussed in Chapter 6.

Chapter 6

Component selection

The major objective of this research is to develop a design approach that provides designers with specific manufacturer part numbers for each component in their design while considering an optimized objective function subject to design constraints. The results of the optimization algorithm step are used to find a discrete set of commercially available components for the design. This section discusses the process of filtering the original datasets to include those with the best components for the specific design, and then the matrix-based exhaustive search across the filtered datasets. Chapter 6.1 discusses how the datasets are filtered based on which components have parameters that are close to or better than the parameters deemed optimal by the optimization algorithm results. Chapter 6.2 describes the matrix-based exhaustive search across the component subsets, and Chapter 6.3 concludes this chapter.

6.1 Normalized scoring

Once the optimization algorithm has converged, the optimized set of component parameters is used to create subsets of the component datasets and then perform an exhaustive search to identify a set of real components that minimizes the objective function, as

shown in Fig. 2.1. As previously noted, the matrix-based exhaustive search on component subsets helps to mitigate the errors associated with the ML model predictions and improves the results of heuristic approaches by bringing the result closer to that of a complete exhaustive search. This section describes the process by which the optimized parameter sets are used to filter the datasets to a subset of components best suited for the design, and how these subsets are used in a matrix-based exhaustive search for component selection.

To determine which components in the dataset will most likely lead to a high-performing design given the specific design inputs, the set of optimized parameters $\bar{O}_{opt}$, $\bar{X}_{opt}$ is used as the starting threshold for the parameter values across all n_a component attributes using the following normalized scoring metric:

$$s_j = \sum_{i=1}^{n_a} \frac{(x_i - x_{i,opt})}{x_{i,max}} \tag{6.1}$$

where x is a component parameter value, x_{opt} is the optimized component parameter value, and x_{max} is the maximum parameter value in the dataset. Note that absolute value $x - x_{opt}$ is *not* used in the numerator, because the desired component has attributes at least as good as the optimized attributes. Including the absolute value may give a component with parameters closer to the optimized set, but overall has a higher, unfavorable parameter set. The number n of components to retain in the dataset presents a trade-off between the execution time and how close the end result may be to the minimum of the considered objective function. This trade-off is examined in the design examples in Chapter 7.

The relevant attributes used to compute the loss, cost, and footprint area of each component type are used in the normalized scoring process. Transistors are filtered to meet or exceed the required voltage rating V_{dss}. Then they are scored based on the following parameters: R_{on}, Q_g, τ_c, τ_{rr}, $C_{oss,0.1}$, U, and A.

To reduce the capacitor datasets, the components with capacitances equal to or greater

than what is required to meet the given ripple ΔV are retained. Then, capacitors are scored based on the cost U and footprint area A. One limitation in capacitor selection is that capacitor root-mean-square current I_{rms} requirements are currently not considered. Therefore, the designer must manually check that the I_{rms} requirement is met by the selected components. The I_{rms} rating information usually cannot be accessed via distributor sites, and automating the process of identifying I_{rms} via datasheets or manufacturer sites is very challenging and could be considered in future work.

To reduce inductor datasets, the tool first calculates winding and core losses, because these losses do not depend on the values of any other components. Computing these losses up-front significantly reduces process run-time because it ensures fewer calls to the fast Fourier transform function to determine the current harmonics. Also, this ensures the inductor selection considers the losses in the context of the operating conditions, which is useful, particularly for cases where the inductor is the largest contributor to the loss.

6.2 Exhaustive search over reduced datasets

Once the datasets are reduced to include only components with the best scores, the tool has a set of m component datasets each of length n. These datasets are then used in a version of the exhaustive search algorithm presented in [0], which relies on matrix operations to make the process computationally efficient. In short, the tool creates a meshgrid of all possible n^m component combinations. The design-oriented parameters are then calculated for each component and used to compute the loss, cost, and area metrics. Component combinations that violate the constraints are dropped, and the list of viable combinations is sorted in the order of best minimization of the objective function. The designer may then select the specific combination of components based on the predicted performance and based on up-to-date availability and pricing, vendor preferences, or any other criteria not accounted for by the tool. Once specific components are selected, the tool can be used to perform an iteration over the switching

frequency to identify the optimal switching frequency given the selected components and design specifications.

6.3 Conclusions

This chapter discusses the implementation of the component selection process, which is the last step of the automated design approach presented here. This is an important step in the heuristic approach based on ML models, because it checks for the discrete component sets to optimize the objective function and thus mitigates errors associated with the predictions of trends. The approach uses the dataset presented in Chapter 3 to determine which components have parameters equal to or better than $\bar{X}, \bar{O}$ from the optimization algorithm step. Then the power loss model using the design-oriented parameters presented in Chapter 4 is used in an exhaustive search to determine which combination of components out of the filtered datasets leads to an optimized objective function.

Chapter 7

Application examples

This chapter applies the automated design approach presented in Chapter 2 to multiple design examples: buck, boost, and microinverter topologies. The selected design specifications are input into the tool, along with all topology-specific inputs including well-developed power loss equations. For all examples, the objective function is chosen to be the power loss minimization. Various constraints are placed on the BOM cost and PCB area of the design depending on the example. In addition, the set of component indices the tool is asked to design varies between examples. As previously mentioned, the objective function and constraints can be any combination of power loss, cost, and footprint area.

Section 7.1 presents an analysis of 48 V-to-12 V, 5 A synchronous buck converter designs. This section includes descriptions of the inputs into the software tool, results after the optimization algorithm converges as well as after the final component selection process, and experimental results. Section 7.2 presents a 12 V-to-48 V, 2 A boost design example, discussing the inputs to the tool and a comparison between experimental and theoretical results. Section 7.3 presents the complete design and build of a more complicated topology, a 430 W dc-ac microinverter example. The inputs to the tool

are described, including a detailed description of the power loss model used throughout the optimization algorithm step and exhaustive search. The optimization step and experimental prototype results are also compared. Finally, Section 7.4 concludes this chapter.

7.1 dc-dc buck converter

The tool described in Chapters 2-6 is used to design 48 V-to-12 V, 5 A synchronous buck converters using Si or GaN transistors, based on Fig. 1.1 and according to the specifications and constraints shown in Table 7.1. The optimization problem is defined in Eq. (2.1), with the objective function defined as the total power loss,

$$f(\bar{X}, \bar{O}) = P_{loss,total}. \tag{7.1}$$

This section is organized as follows: Subsection 7.1.1 discusses the inputs given to the design tool, including examples of the code template, Subsection 7.1.2 details the power loss model $P_{loss,total}$ for the buck converter, Subsection 7.1.3 runs the design tool for the specified buck converter example, and selects four specific design points to complete the component selection. Section 7.1.4 uses the selected components to build hardware prototypes, and discusses the experimental testing results of all four prototypes.

7.1.1 Tool inputs

Corresponding to Algorithm 1 and the block diagram in Fig. 2.1, the list of component indices are:

$$\bar{Q} = [Q_1, Q_2],\ \bar{L} = [L_1],\ \bar{C} = [C_1, C_2]. \tag{7.2}$$

The converter specifications and constraints are

$$\bar{T}_{con} = [V_{in},\ V_{out},\ I_{out}, U_{max,total},\ A_{max,total}, \Delta i_L/I_L] \tag{7.3}$$

Table 7.1: Synchronous buck converter example specifications and constraints.

Constraint	Value
V_{in}	48 V
$V_{dss,required}$	60 V
V_{out}	12 V
I_{out}	5 A
V_{GG}	10 V (Si) or 5 V (GaN)
$U_{max,total}$	\$5-\$11
$A_{max,total}$	800 mm^2
$\Delta i_L / I_L$	70%

The transistor, inductor, and capacitor inputs $\bar{T}$, optimization variables $\bar{O}$, and outputs $\bar{X}$ of the ML models m_Q, m_L, m_C are shown in Table 5.1, where U is the BOM cost and A is the BOM footprint area. At this stage in the tool's development, all other converter components (gate drivers, controller) are assumed to collectively have approximately the same impact on the converter losses, size, and cost in all considered designs, and are therefore excluded from calculations of $f(\bar{X}, \bar{O})$ and the constraints.

Code listings for the inputs are shown in Listings 7.1,7.2,7.3,7.4.

Listing 7.1: Component index inputs for the the buck example.

```
if param_dict['topology'] == 'buck':
    self.fet1 = OptimizerFet(param_dict, 0)
    self.fet2 = OptimizerFet(param_dict, 1)
    self.ind1 = OptimizerInductor(param_dict, 0)
    self.cap1 = OptimizerCapacitor(param_dict, 0)
    self.cap2 = OptimizerCapacitor(param_dict, 1)
    self.fet_list.extend([self.fet1, self.fet2])
    self.ind_list.extend([self.ind1])
    self.cap_list.extend([self.cap1, self.cap2])
```

Listing 7.2: Optimization variables and a subset of all steady-state equations for the buck example.

```
self.fet1.Rds = x[0]
self.fet2.Rds = x[1]
self.fsw = x[2]
self.delta_i = x[3]
if len(x) > 5:
    self.cap1.Area = x[4]
    self.cap2.Area = x[5]

delta_V = 0.05 * self.converter_df.loc[0, 'Vin']
self.cap1.Capacitance = self.converter_df.loc[0, 'Iout'] * self.dc *
    (1 - self.dc) / (
        2 * (x[2] * self.ind_normalizer) * delta_V)
# delta_V spec is: delta_V/Vout <= 1%, call it = 1%.
delta_V = 0.01 * self.converter_df.loc[0, 'Vout']
self.cap2.Capacitance = (self.delta_i / self.delta_i_normalizer) *
    self.converter_df.loc[0, 'Iout'] * (
            1 / (x[2] * self.ind_normalizer)) / (8 * delta_V)
```

Listing 7.3: Snippets of design-oriented variable estimation and power loss model for the buck example.

```
if self.topology == 'buck':

    self.fet1.I_d2_off = self.converter_df.loc[0, 'Iout'] -
    (self.delta_i / self.delta_i_normalizer) * \
                        self.converter_df.loc[0, 'Iout']

    # predict Cossp1 for the fet
    self.fet1.compute_Cdsq(self)
    self.fet1.Cdsq = \
        scipy.integrate.quad(
            lambda Vds: (1 / self.converter_data_dict['Vin'][0]) *
            self.fet1.Cossp1 * 10 ** -12 * (
                    0.1 * self.fet1.Vdss / Vds) ** self.fet1.gamma,
                    0, self.converter_data_dict['Vin'][0])[0]

    # Add the info to the df about kT,
    # the factor to multiply Ron by based on Vdss
    self.fet1.compute_kT(self)

if self.topology == 'buck':
    self.fet1.Rds_loss = self.I_Q1 ** 2 * self.fet1.kT *
    self.fet_normalizer * self.fet1.Rds
    self.fet1.Qg_loss = self.ind_normalizer * self.fsw *
    self.fet1.Qg * self.fet1.Vgate
```

Listing 7.4: Snippets of design-specific input sets for the buck example.

```
if self.example_num == 1 and self.topology == 'buck':
    self.converter_data_dict = {'Vin': [48], 'Vout': [12], 'Iout': [5]}
    if self.fet_tech == 'MOSFET' or self.fet_tech == 'SiCFET':
        self.fet_data_dict = {'Vdss': [60, 60], 'Vgate': [10, 10]}
    elif self.fet_tech == 'GaNFET':
        self.fet_data_dict = {'Vdss': [60, 60], 'Vgate': [5, 5]}
    self.ind_data_dict = {'Current_rating': [10], 'Ind_fsw': [4.6],
    'Current_sat': [1.2 * 10]}


param_dict_buck = {'opt_var': 'power', 'plotting_var': 'cost',
'set_constraint': 'area', 'set_constraint_val': 800,
'example_num': 1, 'tech_list': ['MOSFET', 'GaNFET'],
'test_fet_tech': 'MOSFET', 'num_points': 2,
              'plotting_range': [5, 11], 'predict_components': False,
              'topology': 'buck'}
```

7.1.2 Loss Model for a Synchronous Buck Converter

This section describes a loss model for the buck converter shown in Fig. 1.1. The converter operates from V_{in} input dc voltage and produces V_{out} dc output voltage at I_{out} dc output current. It is assumed that the converter operates in continuous conduction mode with inductor current ripple $\Delta i < I_{out}$.

As shown in Table 7.2, the conduction loss P_{Qc} of the MOSFETs Q_1 and Q_2 is obtained using the standard expression, with the multiplier k_T to account for the anticipated temperature rise. Estimation of k_T is discussed in Subsection 4.1.1.

Loss Symbol	Loss Description	Modeling equation
P_{Qc}	Conduction	$\sum_{i=1,2} k_T R_{on,i} I^2_{rms,i}$
P_{Qrr}	Reverse-recovery charge	$(I_F t_{rr} + Q_{rr})\, V_{in} f_s$
P_{Qds}	Charge-equivalent capacitance	$C_{ds,Q} V^2_{in} f_s$
P_{Qoff}	Off-transition	$\frac{I^2_{off,Q1} t^2_{off}}{48 C_{oss,0.1}} f_s$
P_{Qg}	Gate-charge	$\sum_{i=1,2} Q_{g,i} V_{GG} f_s$
P_{Lw}	Winding	$R_{dc} I^2_{out} + \sum_{k=1}^{11} R_{ac}(k f_s) I^2_{rms,k}$
P_{Lc}	Core	$V_c \cdot \text{iGSE}(\mathbf{B}, \mathbf{t}, K_{fe}, \alpha, \beta)$
P_{PCB}	PCB	$R_{PCB} I^2_{L,rms} r_T$

Table 7.2: Loss model equations for the buck converter based on datasheet parameters and the design-oriented parameters estimated as described in Chapter 4.

Estimation of the switching loss P_{Qrr} due to the reverse recovery of the body diode of the synchronous rectifier Q_2 is challenging. Power MOSFET datasheets most often include a single value $Q_{rr,d}$ for the reverse recovery charge, obtained for recovery from conducting current $I_{F,d}$ at a certain current decay rate $S_d = (di/dt)_d$, as well as single values for the reverse recovery time $t_{rr,d}$, and the amplitude $I_{rr,d}$ of reverse current during recovery. Starting from the datasheet-reported values $Q_{rr,d}$, $I_{F,d}$, S_d, $t_{rr,d}$, $I_{rr,d}$, the diode reverse recovery model [0] can be used to determine the time constants associated with the pn-junction: effective carrier lifetime τ_c and time constant τ_{rr} of the current decay when the junction is reverse-biased. Once $\tau_{c,d}$ and $\tau_{rr,d}$ are determined, the model can be used to calculate Q_{rr}, t_{rr} and I_{rr} at different operating conditions characterized by I_F and S. Details of this datasheet-based reverse recovery model can be found in [0, 0].

In this work, the datasheet-reported values $Q_{rr,d}$, $t_{rr,d}$, S_d, and $I_{F,d}$ have been scraped from the power MOSFETs datasheets. The corresponding calculated values of $\tau_{c,d}$ and $\tau_{rr,d}$ are used to train ML models for power MOSFETs, as discussed in Chapter 5 and summarized in Table 5.1. In the optimization step, the ML model-predicted values for τ_c and τ_{rr} are used to calculate Q_{rr}, t_{rr} and then the loss contribution P_{Qrr} as shown in Table 7.2, based on the operating conditions in the considered converter design. In the buck converter example, $I_F = I_{out} - \Delta i$, while the operating decay rate S has an assumed value of 100 $\mathrm{A\,\mu s^{-1}}$. One may note that the expression for P_{Qrr} in Table 7.2 assumes a "snappy" diode, which tends to overestimate the reverse recovery related loss on Q_1.

Hard discharging of Q_1 output capacitance and charging of Q_2 output capacitance contributes to the switching loss P_{Qds}, which can be modeled using the charge-equivalent capacitance $C_{ds,Q}$ [0, 0, 0]. The automated estimation of $C_{ds,Q}$ across the MOSFET dataset is discussed in Subsection 4.1.3. Note that one assumption made in this estimation method is that $C_{oss,0.1}$ is assumed to be equivalent between Q_1 and Q_2, and $C_{oss,0.1}$ of Q_1 is used in calculations.

An overlap in device voltage and current during the Q_1 switch-off transition leads to a P_{Qoff} switching loss, which assumes a linear decay of current in the Q_1 channel from $I_{off,Q1} = I_{out} + \Delta i$ to zero over the interval t_{off}. This results in an increase in voltage across the output capacitance of the device, approximated as $C_{oss,0.1}$. Here, t_{off} can be considered a constant for a given device voltage rating and type, and $C_{oss,0.1}$ is the device output capacitance C_{oss} at V_{ds} equal to 10% of the rated voltage. Automated estimation of $C_{oss,0.1}$ across the MOSFET dataset is discussed in Subsection 4.1.3.

The inductor loss model reported in Table 7.2 includes the inductor winding loss P_{Lw} due to the dc resistance R_{dc} reported in datasheets, as well as the loss due to the winding ac resistance as a function of frequency $R_{ac}(f)$ in Eq.(4.6) ($m = 11$ current harmonics are considered in the loss model calculations). Furthermore, inductor core

loss P_{Lc} is estimated using the improved generalized Steinmetz equation (iGSE) approach [0]. For the inductor in the buck example, the inputs are the flux density waveform $\mathbf{B} = [-\Delta B, +\Delta B, -\Delta B]$, $\mathbf{t} = [0, DT_s, T_s]$, $\Delta B = L\Delta i_L/(N_{turns}A_c)$, and core volume V_c. Automated estimation of the design-oriented parameters needed for the inductor loss model is discussed in Subsection 4.1.4.

Finally, the total converter loss is obtained as

$$\begin{aligned} &P_{loss,total} = \\ &P_{Qc} + P_{Qrr} + P_{Qds} + P_{Qoff} \\ &+ P_{Qg} + P_{Lw} + P_{Lc} + P_{PCB} \end{aligned} \tag{7.4}$$

7.1.3 Optimization step results and component selection

To examine the impact of the BOM cost constraint $U_{max,total}$ on the achievable power loss, the design tool is run repeatedly over a range of $U_{max,total}$ from \$5 to \$11, and for two transistor types, silicon (Si) and gallium-nitride (GaN). The BOM footprint area constraint is kept constant, as shown in Table 7.1. The algorithm converges in less than one minute for each run. Fig. 7.1 shows the power loss predicted based on the optimized component parameters as a function of the increasing cost constraint $U_{max,total}$. As expected, the optimized power loss decreases as the BOM cost constraint is increased, until the losses reach a lower limit. Also, a minimum cost constraint is necessary for the tool to converge to a feasible design. A comparison is made between Si- and GaN-based optimized designs. One may note how the GaN-based designs outperform Si-based designs as the cost constraint is increased.

Next, two specific cost constraints are used to examine the Si- and GaN-based designs: \$5 (Si_A and GaN_A design) and \$11 (Si_B and GaN_B designs). Table 7.3 shows the converged values of the optimization variables. One may observe that the trends in the optimization variables match expectations, because more expensive designs allow for lower switching frequencies, reduced transistor on-resistances, and smaller inductor

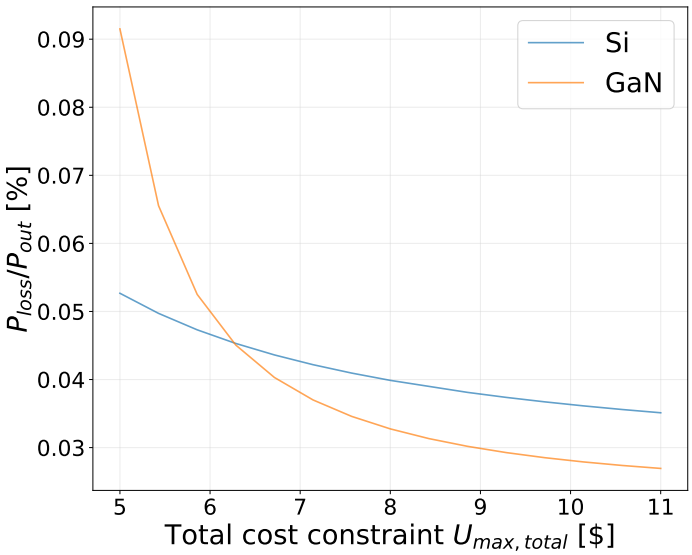


Figure 7.1: Results of the optimization step of the design tool given a range of $U_{max,total}$ for the buck converter example designed using Si or GaN power FETs. P_{loss}/P_{out} is $f(\bar{X}, \bar{O})$ evaluated at $\bar{X}_{opt}, \bar{O}_{opt}$.

current ripple. One may note the differences in parameter breakdown between different technologies under varying cost constraints. One unexpected result is where the lower cost constraint Si and GaN designs show $R_{on,1} < R_{on,2}$. This occurs because of the physical relationship between R_{on} and $C_{oss,0.1}$ of transistors, and then how this factors into the overall loss computation. As seen in Table 7.2, P_{Qoff} and P_{Qds} both contain the $C_{oss,0.1}$ term. When the optimization algorithm adjusts $\bar{O}_Q$, it sometimes finds that by decreasing $R_{on,1}$, the corresponding $C_{oss,0.1}$ prediction increases. Thus, although the P_{Qds} term increases, in some cases, the P_{Qoff} term decreases more substantially, thus leading to an overall lower total power loss.

The four specific design points Si_A, Si_B, GaN_A, GaN_B, were used to complete the design approach. For each design, the optimized parameters $\bar{X}, \bar{O}$ were used to filter the datasets and perform the component selection.

Table 7.4 shows the predicted power loss performance for various values of n components retained in the reduced datasets for transistors, inductors, and capacitors. The component selection step of the tool was chosen to consider the $n = 10$ components

	Si		GaN	
	A	B	A	B
P_{loss}/P_{out} [%]	5.3	3.5	9.0	2.7
$R_{on,Q1}$ [mΩ]	10	6	20	7
$R_{on,Q2}$ [mΩ]	13	6	48	4
f_{sw} [kHz]	150	50	1500	300
$\Delta i_L/I_L$	0.37	0.29	0.26	0.09
A_{C1} [mm^2]	11	9	9	10
A_{C2} [mm^2]	8	10	11	7

Table 7.3: $\bar{O}_{opt}$: Results for $\bar{O}$ after the optimization algorithm converged for the buck converter examples.

with the highest normalized scores, which results in a run-time of 2-4 minutes. This n value represents a good trade-off between run-time and the selected components leading to a highly efficient converter. Increasing the run-time slightly (from $<$ 1 min. to 2-4 min.) results in a significant improvement in predicted converter efficiency as the tool is now able to see a sufficiently large number of combinations even in the context of the specific loss equations, whereas a significant increase in run-time (from 2-4 min. to 20-30 min) leads to a relatively small predicted efficiency improvement. These results show how the ML model-based component parameter recommendations are a good starting point for component selection, and that the automatic component selection step can help mitigate errors in the parameter recommendations that occur as a result of ML model errors and limitations. The final step is the f_{sw} sweep given the selected components. The results for each of the four designs are shown in Table 7.5, where one may observe how adjusting f_{sw} yields small improvements in the power loss.

	P_{loss}/P_{out} [%]			
Design	At X_{opt}	$n = 5$	$n = 10$	$n = 18$
Si_A	5.3	7.4	5.3	4.9
GaN_A	9.0	8.1	7.9	7.4
Si_B	3.8	7.0	4.4	4.3
GaN_B	3.8	3.7	2.3	1.9

Table 7.4: Predicted P_{loss}/P_{out} after running the final step of the approach given various values of n highest-scoring components per dataset. The P_{loss}/P_{out} associated with the component combination leading to the lowest loss is shown here at the switching frequency f_{sw} returned by the optimization algorithm for the buck converter examples.

Design	Step	f_{sw} [kHz]	P_{loss}/P_{out} [%]
Si_A	Algorithm	150	5.3
	Sweep	100	4.9
GaN_A	Algorithm	1500	7.9
	Sweep	2500	7.0
Si_B	Algorithm	50	4.4
	Sweep	100	4.3
GaN_B	Algorithm	330	2.3
	Sweep	200	2.2

Table 7.5: Switching frequencies and correlated power loss at the converged f_{sw} versus after the f_{sw} sweep. In both cases, the P_{loss}/P_{out} calculations are computed using the parameters of the selected components for the buck converter examples.

Table 7.6 shows the break-down of individual loss, cost, and area contributions of Q_1, Q_2, L_1, C_1, and C_2 at the selected design points. The table compares the loss breakdown of the optimized parameters determined by the ML-based optimization step against the loss breakdown given the datasheet parameters of the selected components. It can be seen that the breakdowns are similar between the two results. In most cases, the largest contributor to overall loss, cost, and area is the inductor.

7.1.4 Experimental results

The selected components listed in Table 7.7 were used to build experimental hardware prototypes. The power stages of the assembled boards are shown in Fig. 7.2, and typical operating waveforms are shown in Fig. 7.3. Each prototype was tested across a range of operating frequencies f_{sw} and output power P_{out}, as shown in Fig. 7.4. Fig. 7.5 compares the experimentally measured total power loss against the predicted total power loss, calculated using the selected components and the optimized operating conditions. The predicted loss is broken down into individual loss contributions. One may note how the experimentally measured losses qualitatively match the predicted trends between the four designs: GaN_A had the highest loss, then Si_A, then Si_B, then GaN_B.

The difference in efficiencies between GaN_B and Si_B was not as high as expected. Since more of the cost budget was spent on GaN components, less was available to be

			Si				GaN			
			A ($\leq$ \$5)		B ($\leq$ \$11)		A ($\leq$ \$5)		B ($\leq$ \$11)	
Attribute		Component	X_{opt}	Selected	X_{opt}	Selected	X_{opt}	Selected	X_{opt}	Selected
Loss [W]	P_{Qc}	Q_1	0.10	0.11	0.05	0.11	0.18	0.16	0.04	0.06
		Q_2	0.41	0.34	0.18	0.36	1.35	0.47	0.18	0.17
	P_{Qq}	Q_1	0.03	0.03	0.01	0.009	0.03	0.07	0.03	0.03
		Q_2	0.02	0.03	0.01	0.009	0.01	0.07	0.02	0.03
	P_{Qoff}	Q_1	0.12	0.05	0.02	0.05	1.99	0.70	0.08	0.05
	P_{Qrr}	Q_1	0.57	0.76	0.27	0.17	0	0	0	0
	P_{Qds}	Q_1	0.02	0.05	0.02	0.02	0.07	0.31	0.10	0.11
	$P_{Lw,dc}$	L_1	0.55	1.2	0.74	0.52	0.36	0.14	0.62	0.18
	$P_{Lw,ac}$	L_1	0.38	0.12	0.11	0.20	0.17	0.27	0.04	0.13
	P_{Lc}	L_1	0.70	0.21	0.45	0.27	1.00	2.24	0.27	0.41
	P_{PCB}		0.21	0.25	0.25	0.27	0.26	0.31	0.25	0.26
		Total	3.12	3.15	2.11	2.63	5.42	4.73	1.63	1.40
Cost [\$]		Q_1	0.71	0.57	1.13	0.94	1.89	1.58	3.14	2.53
		Q_2	0.60	0.57	0.99	0.57	1.61	1.58	2.64	2.53
		L_1	3.11	1.97	8.16	7.37	1.10	0.41	4.76	4.02
		C_1	0.30	0.66	0.38	0.91	0.22	0.66	0.26	0.91
		C_2	0.28	0.26	0.34	0.96	0.18	0.26	0.19	0.96
		Total	5.00	4.03	11.00	10.75	5.00	4.49	11.00	10.95
Area [mm^2]		Q_1	16	11	19	11	0.81	0.81	0.81	0.81
		Q_2	18	11	20	11	0.81	0.81	0.81	0.81
		L_1	197	301	511	517	52	44	307	256
		C_1	11	5	9	8	9	5	10	8
		C_2	8	3	10	5	11	3	9	5
		Total	250	331	569	552	74	54	328	270

Table 7.6: Loss, cost, and area contribution breakdown by component and specific loss contribution. The expected performance was generated for the synchronous buck converter examples with the cost constraint $U_{max,total}$ equal to \$5 or \$11, using Si or GaN MOSFETs, given the commercially-available components shown in Table 7.7. In a few instances, due to availability, a component with very similar attributes to the recommended component was selected for the prototype. All calculations shown here are completed using f_{sw} returned by the optimization step. One may note the similarities between the results from the optimization step and those with the selected components. Subsection 7.1.2 discusses each power loss contribution for the buck converter example in detail.

Design	Component	Mfr. part no.	Parameters
Si_A	Q_1	PXN012-60QLJ	$V_{dss} = 60\text{V}$, $R_{on} = 11.5\,\text{m}\Omega$
	Q_2	PXN012-60QLJ	$V_{dss} = 60\text{V}$, $R_{on} = 11.5\,\text{m}\Omega$
	L_1	TMPA 1707SP-330MN-D	$L = 33\,\mu\text{H}$
	C_1	GRM31CC72A475ME11L	$C = 4.7\,\mu\text{F}$
	C_2	CL21A226MAYNNNE	$C = 22\,\mu\text{F}$
GaN_A	Q_1	EPC2052	$V_{dss} = 80\text{V}$, $R_{on} = 13.5\,\text{m}\Omega$
	Q_2	EPC2052	$V_{dss} = 80\text{V}$, $R_{on} = 13.5\,\text{m}\Omega$
	L_1	240-MGAH0603R68M-10CT-ND	$L = 0.68\,\mu\text{H}$
	C_1	GRM31CC72A475ME11L	$C = 4.7\,\mu\text{F}$
	C_2	CL21A226MAYNNNE	$C = 22\,\mu\text{F}$
Si_B	Q_1	SIS176LDN-T1-GE3	$V_{dss} = 70\text{V}$, $R_{on} = 10.9\,\text{m}\Omega$
	Q_2	PXN012-60QLJ	$V_{dss} = 60\text{V}$, $R_{on} = 11.5\,\text{m}\Omega$
	L_1	AMDLA2213Q-470MT	$L = 47\,\mu\text{H}$
	C_1	GRM32EC72A106ME05L	$C = 10\,\mu\text{F}$
	C_2	C3216X5R1E476M160AC	$C = 47\,\mu\text{F}$
GaN_B	Q_1	EPC2204	$V_{dss} = 100\text{V}$, $R_{on} = 5.6\,\text{m}\Omega$
	Q_2	EPC2204	$V_{dss} = 100\text{V}$, $R_{on} = 5.6\,\text{m}\Omega$
	L_1	SRP1510CA-100M	$L = 10\,\mu\text{H}$
	C_1	GRM32EC72A106ME05L	$C = 10\,\mu\text{F}$
	C_2	3216X5R1E336M160AC	$C = 33\,\mu\text{F}$

Table 7.7: Selected components for the buck converter prototypes shown in Fig. 7.2.

spent on the inductor. After analyzing the results, it was determined that most of the discrepancy in the loss model is due to terms involving the switching frequency f_{sw}, i.e. inductor ac losses and switching losses. This can be seen in Fig. 7.4, where the theoretical and experimental losses in the buck converter are compared when taken as a function of f_{sw} and P_{out}. There are more significant differences between experimental and theoretical values in Fig. 7.4(a), which shows efficiency as a function of f_{sw}. There was a noticeable inductor temperature rise, especially in cases of high $\Delta i_L/I_L$, which suggests that the inductor ac winding loss estimation method could be improved. Other differences in the observed efficiency versus the predicted efficiency can be attributed to Q_1 losses, particularly the estimation of $C_{ds,q}$, which uses an estimated $C_{oss,0.1}$ that differs from the datasheet-based value, and the Q_{rr} loss estimation, which uses estimated τ_c and τ_{rr} parameters. It is noted that the evaluation of design-oriented parameters and the power loss equation discussed in Chapter 4 and Subsection 7.1.2 are both simplified estimation approaches, which contribute to the differences between theoretical and measured losses.

The GaN transistors were particularly challenging for hardware prototyping. First, soldering the GaN transistors by hand requires high levels of magnification because the transistors are so small in size ($2.25\,\mathrm{mm}^2$ for the GaN_A design and $3.75\,\mathrm{mm}^2$ for the GaN_B design), as seen in Fig. 7.2(c,d). Slight increases in the solder temperature on the GaN pads can lead to flooding and shorting. In addition, careful consideration was required when designing the GaN power stage layout. The input capacitors needed to be as close to the drain of the transistor as possible, and the gate drive and power loops needed to be extremely tight. After making these adjustments, the GaN signals were noticeably cleaner.

Overall, from the results shown in Fig. 7.5, it can be concluded that the ML-based converter design method developed in this work can be useful in rapidly determining high-performing designs. However, additional measures can be taken to improve the theoretical loss models used in the tool.

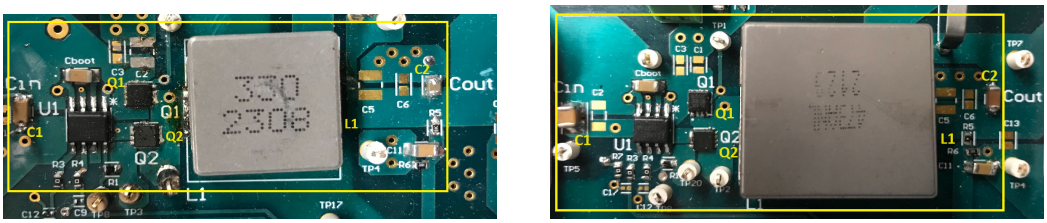


Figure 7.2: Experimental prototypes for the buck converter examples: (a) Si_A, (b) Si_B, (c) GaN_A, (d) GaN_B.

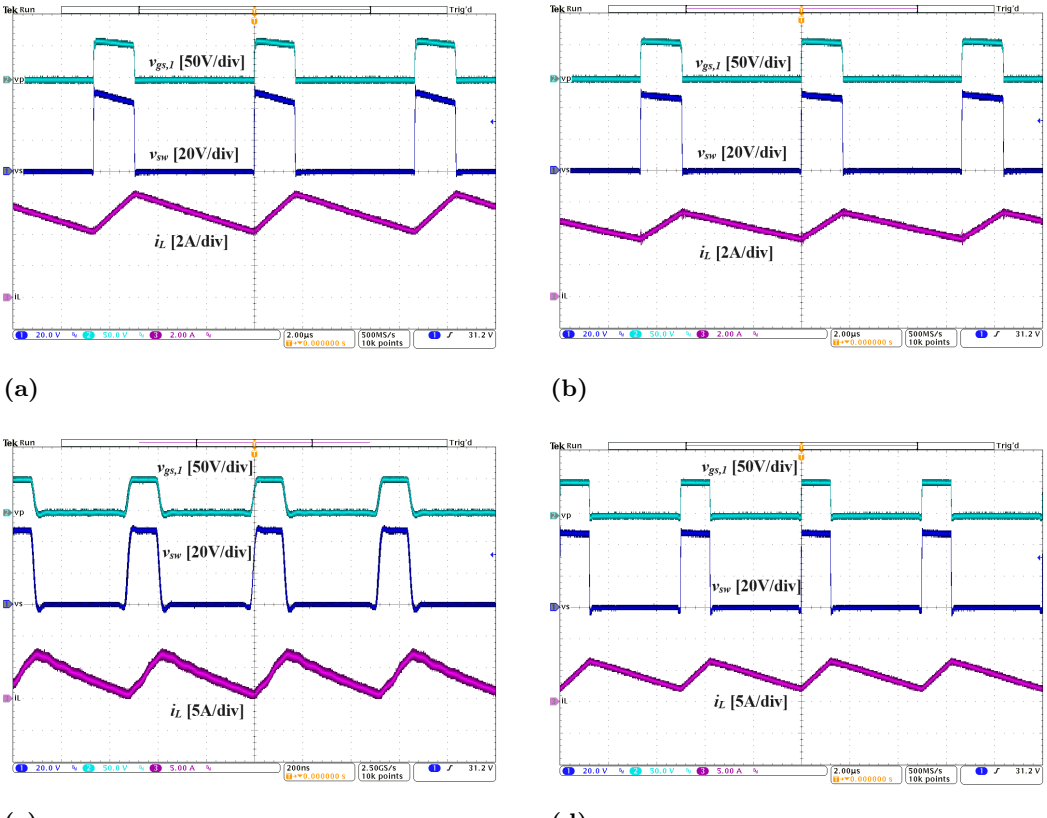


Figure 7.3: Operating waveforms for the buck converter examples: (a) Si_A, (b) Si_B, (c) GaN_A, (d) GaN_B.

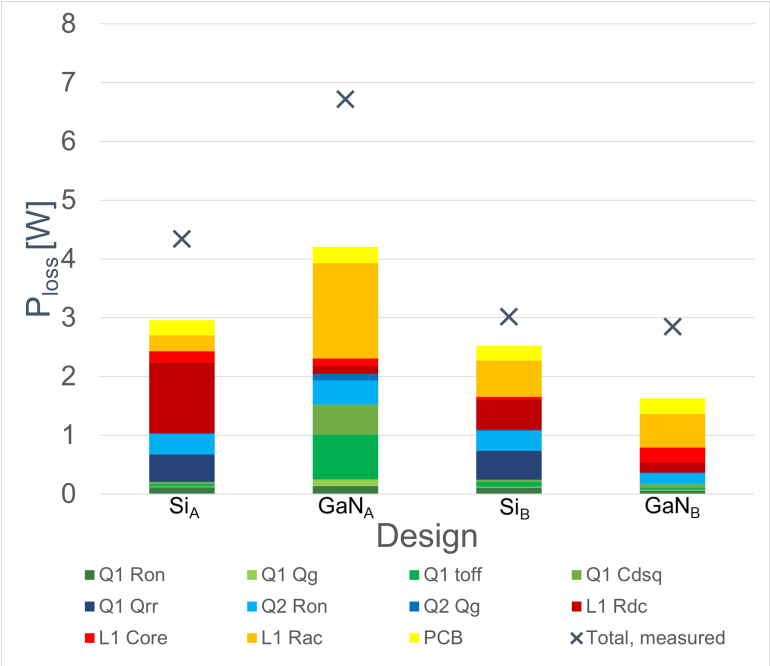
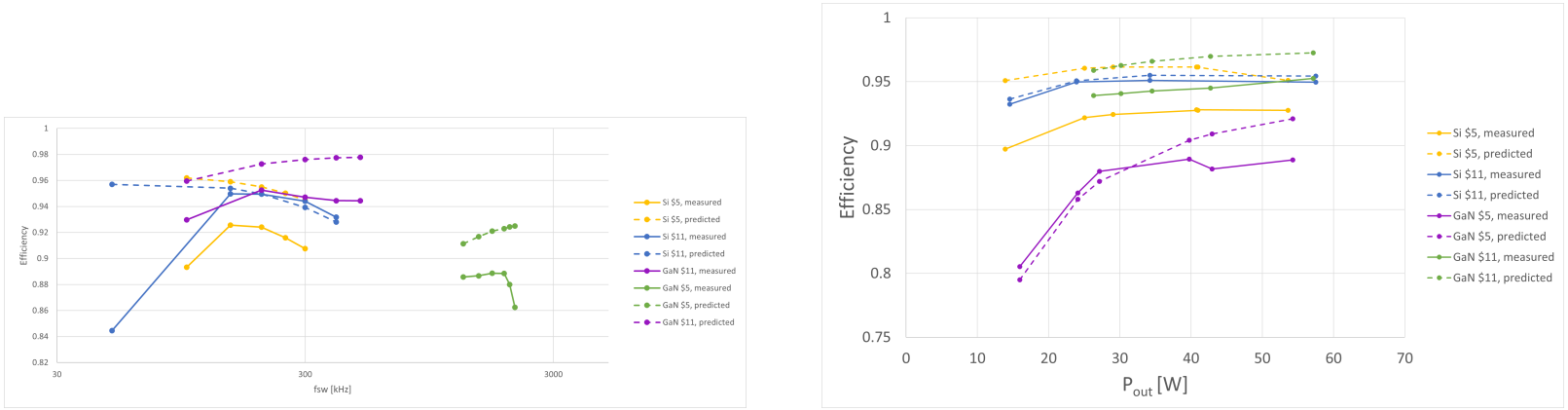


Figure 7.5: Comparison between experimentally measured losses and losses predicted by the loss model described in Subsection 7.1.2 for each of the four buck converter prototypes: Si_A, GaN_A, Si_B, GaN_B.

7.2 dc-dc boost converter

The second design example is of a 12-to-48 V, 100 W synchronous boost dc-dc converter as shown in Fig. 7.6, operating in hard-switched continuous conduction mode. In addition to the voltage and power requirements, the design-specific inputs include the design objective and constraints, which are to minimize the loss while observing the constraints of the BOM cost ($\leq$ \$5) and the BOM footprint area ($\leq 800\,\text{mm}^2$), using Si power MOSFETs. The design approach is applied to the boost example in a similar manner as to the buck example, and will be summarized in this section.

The tool takes the following set of topology-specific inputs for the boost converter. The component indices are the control switch Q_1, the synchronous rectifier Q_2, the inductor L_1, the input capacitor C_1, and the output capacitor C_2. The optimization variables are the on-resistance $R_{on,1}$ and $R_{on,2}$, the switching frequency f_{sw}, the inductor current ripple Δi_L, and the footprint areas A_{C1} and A_{C2} of each capacitor. The remaining topology-specific inputs are the set of standard boost steady-state solution equations



(a) **(b)**

Figure 7.4: Power losses in the buck example comparing the theoretical and experimental results for (a) power loss vs. f_{sw} and (b) power loss vs. P_{out}.

	Mfr. part no.	Parameters
$Q_{1,2}$	PXN012-60QLJ	$V_{dss} = 60\,\text{V}$, $R_{ds,on} = 11.5\,\text{m}\Omega$
L_1	SRP1265C-8R2M	$L = 8.2\,\mu\text{H}$, $I_{rated} = 13\,\text{A}$
C_1	GRM31CC72A475ME11L	$C = 4.7\,\mu\text{F}$, $V_{rated} = 100\,\text{V}$
C_2	GRM32EC72A106ME05L	$C = 10\,\mu\text{F}$, $V_{rated} = 100\,\text{V}$

Table 7.8: Selected components returned by the optimization tool, used to build the boost example hardware.

and the power loss model (similar to the model described in [0]).

After entering the inputs and running the tool, and considering component availability, the components selected for Q_1, Q_2, L_1, C_1, and C_2 are shown in Table 7.8. In the hardware prototype, the selected C_2 is replaced by EEU-HD1H470B. The boost converter prototype is shown in Fig. 7.7(a) and operating waveforms are shown in Fig. 7.7(b). Fig. 7.8 shows the loss breakdown predicted using the selected components' datasheet information and the power loss model using the developed design-oriented parameters as presented in [0]. It can be seen that the inductor R_{dc} and the transistor Q_{rr} provide the highest predicted contributions to the overall losses. The predicted loss using the selected components (4.8 W) is close to the loss from the optimization algorithm output (5.2 W) using the optimized theoretical component parameters. The predicted loss with the selected components is slightly lower than the optimization algorithm's predicted loss because the ML models capture trends in the component data, and specific components may exist that yield better performance. The exhaustive search on the limited dataset is beneficial to identify such components. The experimentally measured loss is somewhat higher (6.5 W). These discrepancies are similar to those discussed in Section 7.1, where the differences between theoretical and experimental are likely due to an underestimation in the prediction of the inductor ac loss, especially at high $\Delta I_L/I_L$ (here, $\Delta I_L/I_L = 0.39$), as well as additional transistor and PCB trace losses unaccounted for in the loss model.

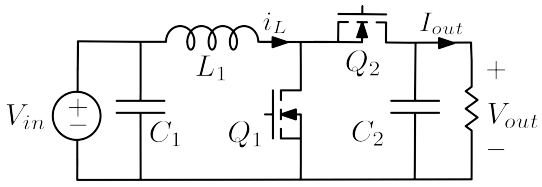
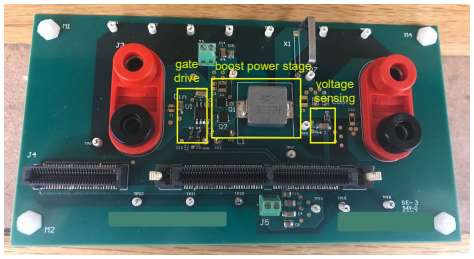
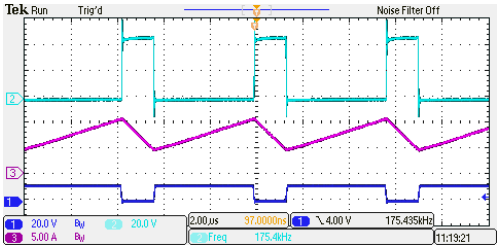


Figure 7.6: Circuit diagram of the boost converter.



(a)



(b)

Figure 7.7: (a) Boost converter prototype for a 12-to-48 V, 2 A output design operating in hard-switched continuous conduction mode. The design objective was to minimize the power loss given a BOM cost constraint of \$5 and a BOM area constraint of 800 mm^2. The components selected by running the tool are reported in Table 7.8. (b) Operating waveforms: switched-node voltage (teal), inductor current (purple), Q_1 gate control signal (blue).

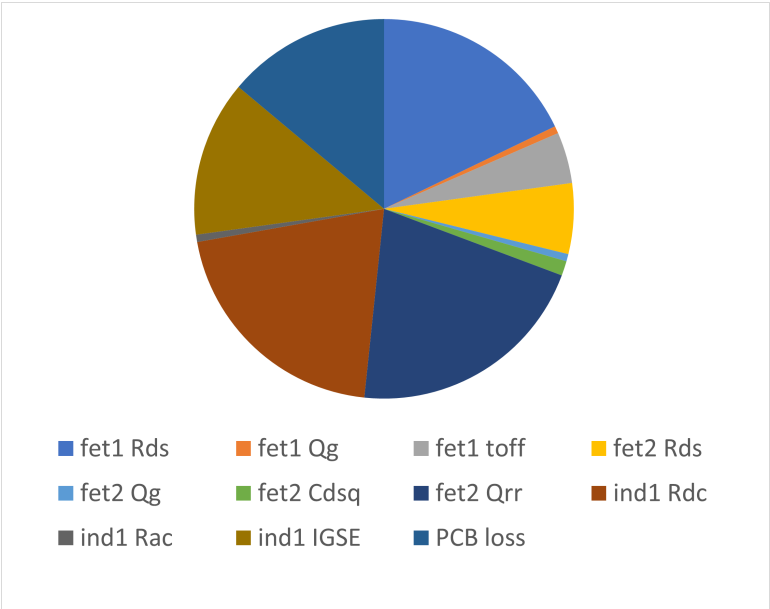


Figure 7.8: Theoretical loss distribution using the selected components for the Si-based boost converter example. Note that inductor R_{dc} and FET Q_{rr} are the largest contributions to overall loss.

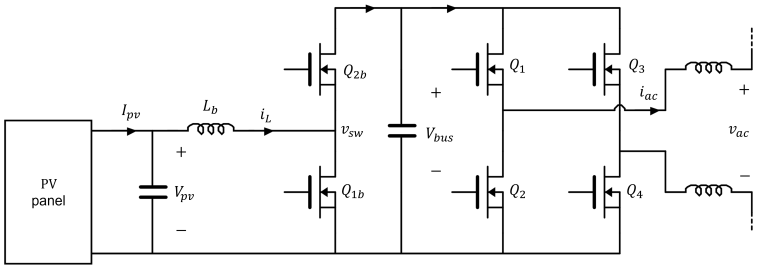


Figure 7.9: Low-voltage microinverter module intended for a cascaded H-bridge PV architecture. The nominal operating conditions are $V_{pv} = 33\,\text{V}$, $V_{bus} = 60\,\text{V}$, $V_{ac,rms} = 40\,\text{V}$, and $I_{ac,rms} = 10.8\,\text{A}$.

7.3 dc-ac microinverter

The third application example discussed in this thesis is of a two-stage, low-voltage microinverter example as shown in Fig. 7.9. This design example shows how the tool can be applied to more complex converter topologies. The microinverter module is supplied from a 430 W photovoltaic (PV) panel and is intended for a single-phase cascaded H-bridge PV architecture where the module ac outputs are connected in series to interface to the ac grid [0, 0, 0]. An advantage of this architecture is that all the components can be low-voltage and relatively low cost, resulting in high efficiency. Furthermore, the interleaved operation of the H-bridge inverters results in a multi-level ac waveform, thus reducing the size of the ac-side filter in the cascaded architecture.

The two stages of the microinverter module are a boost dc-dc stage followed by the H-bridge inverter. The module design specifications are shown in Table 7.9. To minimize switching losses, the boost stage operates in the zero-voltage-switching quasi-square-wave (ZVS-QSW) mode [0]. The switching frequency $f_{sw,b}$ changes with load to minimize root mean square (RMS) current and conduction losses while maintaining a sufficiently large current ripple to achieve ZVS-QSW operation [0]. The H-bridge stage is hard switched using a two-level sinusoidal PWM at a relatively low switching frequency $f_{sw,i} = 20\,\text{kHz}$, taking advantage of the multiplicative effect of the multi-level operation in the cascaded H-bridge architecture.

Table 7.9: Specifications for the microinverter module design example.

Specification	Value
V_{pv}	33 V
V_{bus}	60 V
$V_{ac,rms}$	40 V
$I_{ac,rms}$	10.8 A
$D_{sw,b}$	0.45
I_L	13.05 A
L_b	7.6 μH
ΔV_{input}	$0.34V_{pp}$
C_{input}	163 μF
ΔV_{bus}	5 V
C_{bus}	1.9 mF
$f_{sw,b}$	75 kHz
$f_{sw,i}$	20 kHz

7.3.1 Tool inputs

The design specifications as listed in Table 7.9 are used as the design-specific inputs $\bar{T}$ into the tool, in addition to the design objective, which is to minimize the loss while observing a BOM cost constraint $U_{max,total}$. In this design example, the only considered components are the power semiconductor components, which include the two transistors of the boost stage and the four transistors in the H-bridge inverter. This example illustrates how the tool can be used to focus on the selection of a subset of converter components.

The topology-specific inputs $\bar{T}$ are as follows. The component indices are the boost transistors Q_{1b}, Q_{2b} and the H-bridge transistors Q_{1-4}. All four transistors in the H-bridge inverter are assumed to be the same. The boost inductor L_b is a planar inductor designed separately, while the boost input capacitor C_1 and energy storage capacitor C_{bus} are selected manually using standard design practices $[0, 0, 0]$. The output ac inductors, which are not considered in the microinverter module design process, are selected separately based on system-level considerations in the cascaded H-bridge PV architecture.

The optimization variables are the on-resistances $R_{on,1b}, R_{on,2b}$ of the two MOSFETs

Table 7.10: Loss model equations for the microinverter module based on datasheet and estimated design-oriented parameters as described in Section 7.3.2. Here, k_T represents an increase in R_{on} at an elevated junction temperature, E_{Qrr}, E_{on} and E_{off} follow from a basic loss model presented in [0], $T_{sw,i} = 1/f_{sw,i}$, and $k_{sw,i} = 0.5 f_{sw,i}/f_{line}$ is the number of H-bridge inverter switching periods per one-half line period.

Loss Symbol	Loss Mechanism	Loss model equation
P_{Qc}	Conduction	$\sum_{j=1b,2b} k_T R_{on,j} I^2_{rms,j} + 2k_T R_{on,1-4} I^2_{ac}$
P_{Qrr}	Body-diode reverse recovery	$2f_{line} \sum_{k=1}^{k_{sw,i}} 2E_{Qrr}\left(kT_{sw,i}\right)$
P_{Qds}	Charge-equivalent capacitance	$2f_{line} \sum_{k=1}^{k_{sw,i}} 2E_{on}\left(kT_{sw,i}\right)$
P_{Qoff}	Off-transition	$f_{sw,b} E_{off,b} + 2f_{line} \sum_{k=1}^{k_{sw,i}} 2E_{off}\left(kT_{sw,i}\right)$
P_{Qg}	Gate-charge	$\sum_{j=1b,2b} Q_{g,j} V_{GG} f_{sw,b} + 4Q_{g,1-4} V_{GG} f_{sw,i}$

in the boost stage and the on-resistance $R_{on,1-4}$ of the four MOSFETs in the inverter stage. Taking the inductor design considerations into account, at the maximum output power, the boost-stage switching frequency is set to $f_{sw,b} = 75\,\text{kHz}$. The boost inductance L_b is such that the ZVS-QSW operation is achieved with a minimum conduction loss, which means that the inductor current ripple approximately equals the inductor dc current, $\Delta I_L \approx I_L$.

The final topology-specific input is the power loss model, which is briefly summarized in the next section.

7.3.2 Power loss model

The power loss model for the power semiconductors in the microinverter module combines the losses from the boost and the H-bridge inverter stages, as shown in Table 7.10. Note how the loss model equations use the datasheet and the design-oriented device parameters presented in Chapter 4.

Power semiconductor losses in the boost stage include the conduction and gate-drive (Q_g) losses. Since the boost stage operates in the ZVS-QSW mode, the relatively low

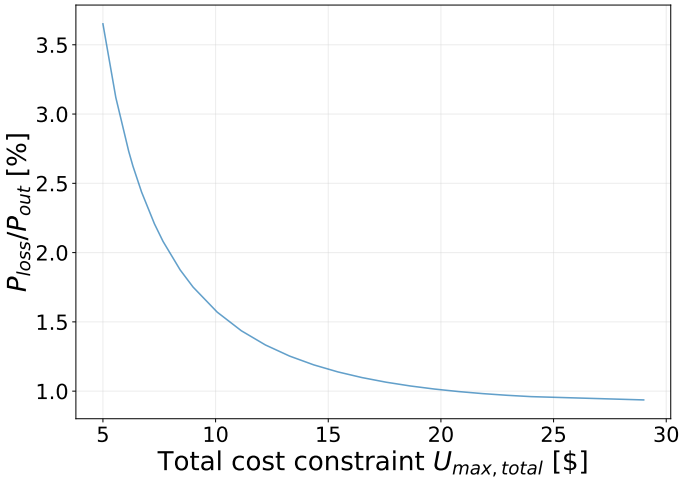


Figure 7.10: An application of the design tool for the microinverter example: at each BOM cost constraint $U_{max,total}$ for the power semiconductors, the optimized component parameters are used to predict the total power loss associated with the power MOSFETs Q_{1b}, Q_{2b} in the ZVS-QSW boost stage, and Q_{1-4} in the hard-switching H-bridge inverter stage.

switching loss is dominated by the turn-off of Q_1, which occurs when the inductor current is at its peak value of approximately $2I_L$.

To calculate the losses in the hard-switching H-bridge inverter stage, the switching ripple in the output ac line current is neglected. In each switching period, conduction and switching losses are considered, including the effects of body-diode reverse recovery, turn-off transition, device output capacitance, and gate charge.

The loss model iterates over a half-cycle of the line period, where f_{line} is assumed to be 60 Hz. The total energy lost over one half-line cycle is summed, and the power loss is obtained by multiplying the total energy loss by $2f_{line}$.

7.3.3 Optimization step results and component selection

Fig. 7.10 shows the power loss results of the optimization step after the optimization algorithm converged, given a varying constraint $U_{max,total}$ for the BOM cost of the power MOSFETs. One may note how the predicted power semiconductor loss drops to less than 1% as the allowed BOM cost $U_{max,total}$ of the power MOSFETs increases.

		Si FETs	
		U $\leq$ \$9	U $\leq$ \$14
Loss [W]	Q_{1b}	1.6	0.9
	Q_{2b}	1.0	0.6
	Q_{1-4}	4.9	3.6
	Total	7.5	5.1
BOM cost U [\$]	Q_{1b}	1.57	2.56
	Q_{2b}	1.94	3.19
	Q_{1-4}	5.49	8.24
	Total	9.00	14.00
BOM area A [mm^2]	Q_{1b}	25	31
	Q_{2b}	19	36
	Q_{1-4}	96	84
	Total	140	151

Table 7.11: Loss, cost, and area contribution breakdown by component contribution from the optimization algorithm results of the microinverter example. The expected performance was generated for the microinverter examples with a cost constraint $U_{max,total}$ equal to \$9 or \$14, using Si MOSFETs.

	Mfr. part no.	Parameters
Q_{1b}, Q_{1-4}	SIR104LDP-T1-RE3	$V_{dss} = 100\,\text{V}$, $R_{ds,on} = 6.1\,\text{m}\Omega$
Q_{2b}	SIDR668DP-T1-RE3	$V_{dss} = 100\,\text{V}$, $R_{ds,on} = 4.8\,\text{m}\Omega$

Table 7.12: Selected components for the microinverter example.

The results from the optimization step for $U_{max,total} = \$9$ and \$14 are summarized in Table 7.11. The theoretical device parameters optimized at $U_{max,total} = \$14$ were used to filter the database and perform the exhaustive search over a small subset of components from which a set of power semiconductor components was selected, as shown in Table 7.12. Fig. 7.11 shows the theoretical loss contribution breakdown with these selected components, based on the power loss model described in Subsection 7.3.2.

7.3.4 Experimental results

Using the transistors suggested by the tool, the separately designed planar inductor, and the manually selected input and energy storage capacitors, the microinverter module prototype was constructed and is shown in Fig. 7.12. The module was tested at operating conditions close to the specifications in Table 7.9. Figs. 7.13(a,b) illustrate the operation of the boost stage in the ZVS-QSW mode. At $V_{PV} = 33\,\text{V}$,

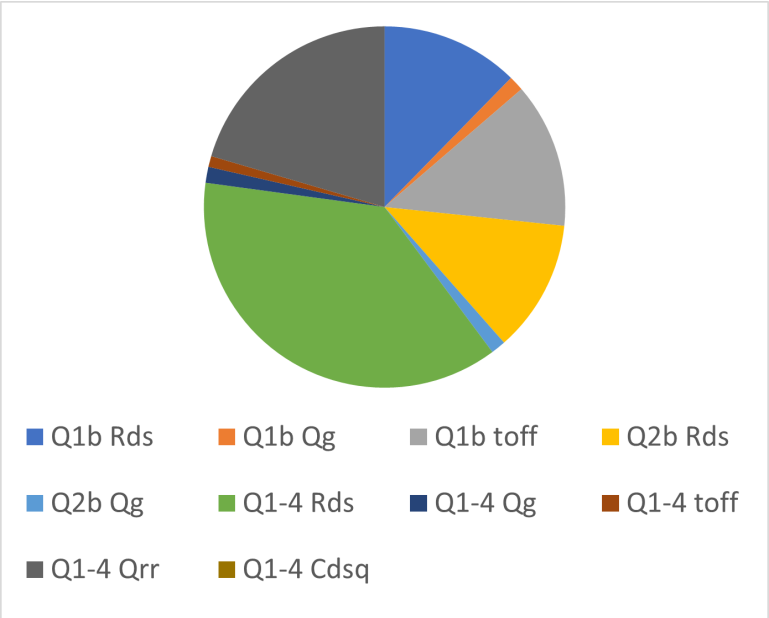


Figure 7.11: Theoretical loss breakdown for the Si-based microinverter design example with the selected components. This chart represents only the losses on the power MOSFETs in the boost stage and the H-bridge inverter stage, which sum to 6.1 W at the operating point specified in Table 7.9.

V_{bus} = 61.6 V, P_{bus} = 406 W, the boost stage efficiency is 97.8%, which is consistent with loss model predictions for Q_{1b} and Q_{2b} when losses attributed to the boost inductor, PCB traces, and capacitor ESR are accounted for. Fig. 7.13(c) illustrates the operation of the complete microinverter module, when operating at $V_{PV} = 33$ V, $V_{ac,rms} = 39.1$ V, $P_{ac} = 370$ W in a stand-alone mode with an additional ac-side LC filter and a resistive load.

7.4 Conclusions

In this chapter, multiple design examples are presented using the developed design approach. The examples include 48-to-12 V, 5 A output buck converters and a 12-to-48 V, 2 A output boost converter operating in hard-switched mode, as well as a two-stage 430 W low-voltage microinverter. The developed design tool was used to select components for some or all of the major components in these design examples. For the buck and boost examples, the tool design included two switch transistors, one inductor, and input and output capacitors. For the microinverter example, the tool designed for

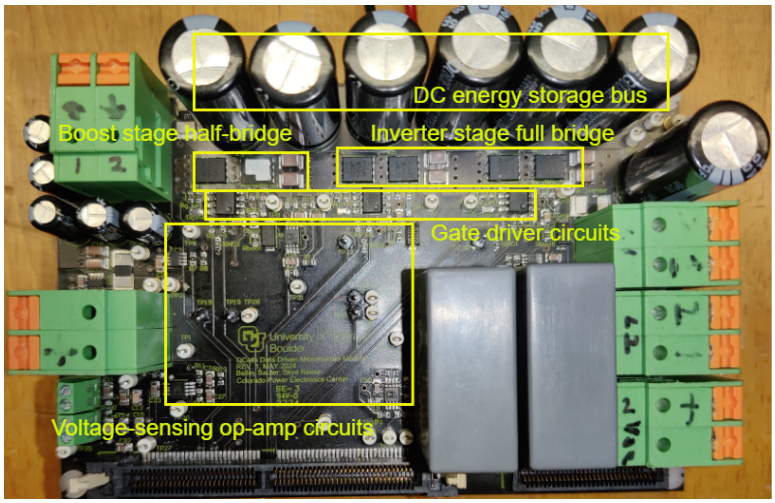
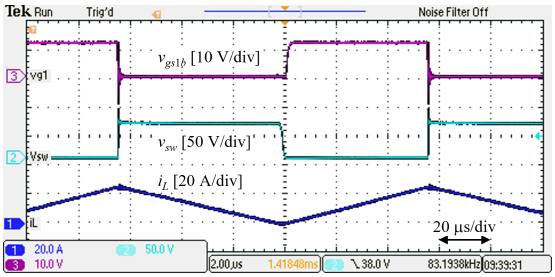
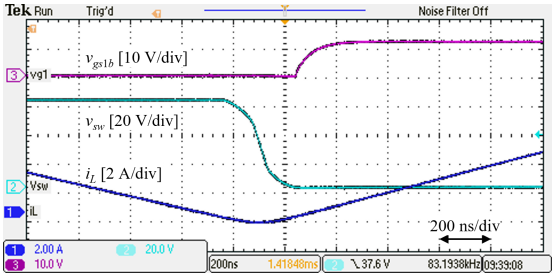


Figure 7.12: Microinverter module hardware prototype. The design objective was to minimize the power loss in the power semiconductors given a BOM cost constraint of $14. The components selected by running the tool are summarized in Table 7.12.

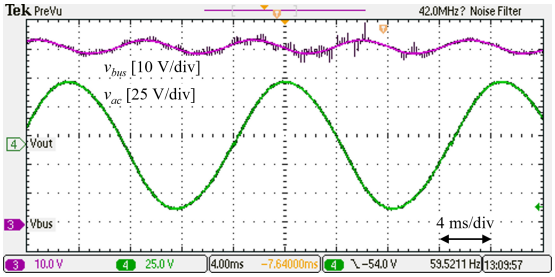
the two transistors in the boost and the four transistors in the H-bridge. The objective was to minimize power loss, given constraints on the BOM cost and footprint area. Various designs were considered and compared, using Si or GaN transistors, and high and low cost constraints. For each design, the tool was run and the results after the ML-based step were used to execute an exhaustive search on subsets of the component datasets and reach a final component combination. These components were used to build hardware prototypes, which were tested and revealed close performances to that of the theoretical results. The variety of included examples validates the flexibility of the developed tool.



(a)



(b)



(c)

Figure 7.13: Operating waveforms in the microinverter prototype module: (a) boost-stage waveforms at $V_{PV} = 33\,\text{V}$, $V_{bus} = 61.6\,\text{V}$, $P_{bus} = 406\,\text{W}$, (b) details of the Q_{2b}-turn-off to Q_{1b}-turn-on ZVS transition in the boost stage, (c) microinverter ac output voltage v_{ac} and bus voltage V_{bus} at $V_{PV} = 33\,\text{V}$, $V_{ac,rms} = 39.1\,\text{V}$, $P_{ac} = 370\,\text{W}$.

Chapter 8

Summary and future work

In this chapter, the main outcomes and conclusions resulting from this body of work are summarized in Section 8.1, followed by a discussion on future research directions in Section 8.2. There are future research steps for many of the elements comprising the overall ML-based power converter design approach, including dataset acquisition, design-oriented parameter and loss model development, ML modeling, experimental validation, and design implementation. Finally, Section 8.3 concludes this chapter.

8.1 Thesis summary

Machine learning (ML)-based design automation tools can be used to assist designers in the rapid selection of components in converter designs achieving high performance across a range of metrics including efficiency, cost, and size. The component selection process can be considered a combinatorial optimization problem. Various solution algorithms exist to solve such a problem, including exact and heuristic methods. This work presents a heuristic approach using ML models and a linear programming optimization algorithm to determine subsets of component datasets such that an exhaustive search can be performed in bounded run-time. This is a data-driven approach, which is developed into a tool used to select components for switching converter designs. The

approach significantly reduces design time and automatically generates high-performing designs that meet design constraints.

Data is gathered from distributor websites and component datasheets, as presented in Chapter 3. Data is gathered on FET transistors, surface-mount fixed-core MPP inductors, and Class II ceramic capacitors. These datasets are cleaned using various techniques including bounding, outlier removal, and Pareto dominance, which filter the datasets to components likely used by a designer for an optimized design. In Chapter 4, methods are presented to develop a more accurate power loss model that combines device physics-based modeling methods with datasheet parameters to improve accuracy. The developed methods model additional losses due to FET on-resistance, charge-equivalent capacitance, and reverse recovery charge, as well as inductor core and winding losses, and capacitor dc-voltage capacitance dependence. Chapter 5 trains ML models on the data and uses these ML models throughout an optimization algorithm. As discussed in Chapter 2, the selected optimization algorithm is COBYLA, a multivariate optimization that makes linear approximations of non-linear functions and can handle constraints and bounds on the variables. The algorithm converges to a set of optimized component parameters for every component in the design. These parameters are used to filter the original databases to that which can be computationally handled by an exhaustive search, as discussed in Chapter 6. The design approach is developed into a Python-based tool, and is structured such that it is relatively simple for another designer to use the tool to design for their specific topology and design specifications. The developed tool is run on buck, boost, and microinverter design application examples, as shown in Chapter 7. Components are selected from the recommended set of optimal combinations and used to build experimental prototypes. The performance of the prototypes is compared with the expected results from the tool and shown to have a close match in most cases. Differences are discussed between the theoretical and experimental loss measurements. The approach is shown to quickly generate component recommendations that achieve a design objective and meet all design constraints.

The main contributions of this thesis are summarized as follows. First, this work created a large database of machine-readable component data, as well as a methodology to significantly expand the dataset for any component information available on distributor sites or PDF datasheets. As the power loss modeling methods are expanded, more component parameters will be required, so having a way to make data machine-readable is beneficial. Methods are presented to filter the data to reduce the large datasets to smaller datasets of components which would likely be selected for an optimized design. This is useful because it can throw out components from the dataset that are less likely to be considered and reduce the variability in the data parameters.

Second, a set of ML models was developed and trained on the component data. These models provide a framework for making predictions given inputs and outputs specified using the data for a particular application. The ML models in this work were structured for use in the selected optimization algorithm, but the same ML framework can be customized for other applications. The results of the ML models on component data, as well as visual analysis of the data itself, confirmed expectations about the data. For example, given limited sets of inputs, the outputs can have a high standard deviation. Certain regression algorithms were selected for this application, but the specific algorithm can be easily changed.

Third, this work developed datasheet- and device-physics-based design-oriented parameters that can be computed using only datasheet data and operating conditions and are thus useful within a power loss model in automated approaches. The development of these design-oriented parameters involved gathering data only available in graphs, which is very difficult to acquire, as methods of accurate, entirely automatic graph digitization are currently lacking. The developed design-oriented parameters can be used for any converter topology, and depend on quantities such as initial values, operating conditions, and expected operation. Including design-oriented parameters to create a well-developed power loss model is also useful outside of the design approach presented in this work. For example, if a designer had a pre-selected set of components

and wanted to compute expected losses, they can input the datasheet parameters of the components as well as their desired operating conditions and are provided with an approximate predicted loss.

Fourth, this work applied a heuristic approach of solving discrete combinatorial optimization problems using ML models to solve the problem of converter component selection. This optimization framework using ML models could also be used for solving other power electronics problems, such as optimal custom magnetics design. The optimization step of the approach is followed by a scoring method to determine the highest performing components in the context of the specific design problem and requirements, and these components are used within a matrix-based exhaustive search. This heuristic approach is developed into a tool that is structured to be easily used by a designer to design any converter given the proper inputs about the topology and design. Thus, any designer can quickly get a set of commercially available manufacturer part numbers for every component in their design, where the components are selected in order to optimize the specified objective function. The designer can also be confident that the component set will meet all design constraints. In addition, the designer is provided information about operating conditions that are also intended to optimize the objective function given the selected components.

Finally, this work provides experimental validation of the tool. The tool was used to design many converters: four separate buck converter examples, one boost example, and one microinverter example. The sets of components returned by the tool were used to build these converters. The converters were then tested and the experimental results were compared to the theoretical results, highlighting areas where the current approach could be improved while also validating the ability of the approach to make good predictions and part recommendations.

8.2 Future Work

There are various future research objectives to further develop the various steps involved in the ML-based automated design approach presented in this work. This section discusses expanding the component database, continued work exploring ML modeling, improving the power loss model, performing additional hardware testing, and further integrating developing the tool.

8.2.1 Additional components

Many of the major contributions to power losses are included in the current version of this tool, primarily the contributions from transistors and inductors. However, adding components to the database would contribute to better representation of the total BOM cost and area of the PCB. For example, data could be included to represent the gate driver and sensing stages, as well as heat sinks in order to model thermal management. Transformer data would expand the tool to account for transformer-based topologies. More types of inductors and capacitors could be scraped and included in the datasets, for example inductors made from different core materials, or capacitors of other class types. More component information could also be collected from the graphs in the PDF datasheets, however this would require further graph digitization. Currently, various graph digitization tools exist on the internet or as downloaded applications. However, as previously discussed, these methods still involve a lot of manual input. For example, the user is required to specify the axes and to click individual points along the lines. A tool that could automatically read a graph would be highly useful for developing more accurate design-oriented parameters. The tool ideally could identify all necessary information to determine a dense series of data points representing each line in a graph, and turn that data into machine-readable format. In the meantime, more work could be put into manually gathering this graphical data.

In addition, more work can be put into normalizing component quantities based on test conditions. For example, in a typical MOSFET component datasheet, the R_{on} value

will list the V_{gs} and I_D conditions. In the current approach, these condition values are not considered and only the provided R_{on} value is taken into account. Future work includes developing a way to normalize the R_{on} value such that it can be directly compared across components.

The inclusion of inductors could be further elaborated by developing a method of custom inductor design. Rather than using the commercially available fixed inductors available on distributor sites, information on the core material, size, shape, winding layers and turns, and Steinmetz parameters could be used to determine optimal specific inductor design parameters. Once these parameters are determined by the tool, the designer could build an inductor using these parameters as a framework. This would allow for a more fine-tuned inductor design based on the operating conditions, thus further optimizing the inductor contributions to loss, cost, and size. However, acquiring the necessary inductor data and turning it into a machine-readable format is challenging and requires a considerable amount of additional research.

8.2.2 Loss model improvement

As previously discussed in this thesis, one significant improvement to the overall effectiveness of the tool would be to further develop the inductor ac loss estimation method. It was concluded that this method underestimates the measured losses, particularly when $\Delta i_L/I_L$ is high. Therefore, it would be worthwhile to adjust the estimation method to determine where the estimation values could be higher. For example, future work could include determining additional expected percentage loss correlated with inductor parameters and operating conditions, or determining which specific steps in the estimation method are leading to an underestimation of the loss.

In addition to improving inductor ac losses, improvements can be made to the estimation methods for transistor losses. For example, more data can be gathered and used to determine the γ value for the charge-equivalent capacitance computation or a more complex relationship between V_{dss} and γ may be shown to be a better representation

of the relationship. Both of these would lead to a better prediction of the C_{dsq} integral.

Thermal considerations could also be included. This could include scraping data and generating ML models for thermal management components such as heat sinks or forced air cooling, and accounting for thermal management within the PCB such as the area and cost taken by adding thermal vias and thermal relief pads. Additional thermal challenges that the tool could account for include modeling where there may be thermal runaway or how the component volume may change with thermal expansion. These complexities could be modeled given additional data and modeling mechanisms.

Additional loss values could also be included. This would add complexity to the tool because values would have to either be known, calculated, or estimated, but could contribute to a more accurate power loss model. For example, an additional power loss that could be included is capacitor equivalent series resistance (ESR) losses, which would require knowledge of the capacitor ESR that can be scraped from various locations (simulation tools, characterization sheets, etc.) for capacitors.

Finally, ML models could be used to improve the accuracy of the power loss models and the way the computed losses compare to the measured losses. One way this could be incorporated is by using simulation data. For example, given a specific topology, large quantities of data could be generated by running many iterations of that topology in simulation software while iterating over a range of operating conditions and component entries. Then the results from these power loss models could be combined with the power loss model using the datasheet data. In addition, experimental data could be used for training. This would require more labor on the part of the tool developer, but would be particularly useful. For example, many loss values could be measured after building the hardware and testing at various operating conditions, and swapping out component combinations. A neural network could be useful in considering how the combinations of certain components can affect the measured losses, and recognize patterns in the differences between the measured losses and those generated from the power loss model. When the tool is run, these differences can be used to generate more

accurate loss predictions using the optimized parameters and selected components, even if that specific hardware has not yet been built. As an example, extensive hardware testing was performed to generate the MagNet database [0], and this data was used to develop ML models that could more accurately predict core losses in magnetic components. These databases could be used for the design tool presented here, to have better loss model predictions and more options for inductor selections.

8.2.3 ML model development

Because ML models' accuracy largely depends on the quality and quantity of the datasets used for training, all of the dataset additions described in Subsection 8.2.1 would greatly benefit the ML models. Also mentioned in Subsection 8.2.1, the addition of components would require further ML model training in order to be used within the approach. For example, given input information about the gate transistor device type, input power supply, and transistor gate drive rating, ML models could be used to predict the gate driver cost, area, and number of required additional resistors. The number of additional resistors could then be used to make predictions about additional cost and area contributions resulting from including these resistors.

In addition to training more ML models, the ML models themselves can be further researched. This research found that regression ML model algorithms were sufficient for the application, and due to the high standard deviation in the data given a limited set of known inputs, the results when testing on the training data showed similar performance. However, as the tool is developed, incorporating more known inputs would lead to reduced variation in the outputs. Also, neural networks trained on component data could be investigated further and potentially included in the tool. Basic neural nets were developed and then trained and evaluated on the datasets but showed little improvement in performance scores.

8.2.4 Experimental validation

Additional hardware testing would be valuable to the validation of the tool and point out additional areas where the tool can be improved. For example, it is of interest to run the tool with a design objective to minimize the cost or area rather than total power loss. For example, one design could be to build a converter that still meets minimum efficiency and maximum area requirements but is the lowest possible cost. Or, a combination of metrics could be used in the objective function, such as a FOM of $U_{tot}\eta$. The tool and hardware testing could also be performed using SiC transistors for any of the designs. Using SiC transistors would be particularly interesting for high-power designs because SiC components tend to be more expensive and also cover higher V_{dss} compared to Si MOSFET counterparts.

As discussed in Chapter 2, the tool can easily be generalized to any converter topology, given the inputs based on the provided template. For example, further work may include running the tool and testing on the microinverter example when also including the inductor and capacitors in the list of component indices to optimize. Another interesting case study would be to input additional topologies into the library of loss models and then run and test various design examples.

Because the tool can run quickly and iterate over many different designs, the tool could be used to compare the expected performance when given a large array of similar topologies. For example, [0] discusses composite converter architecture synthesis, including input-series output-parallel (ISOP) and input-parallel output-series (IPOS) architectures developed from investigating all possible module interconnects. Seven valid converter architectures remained from these combinations. The tool developed here could be used to evaluate the expected performance of all seven architectures, without having to separately build and test each one. The tool could be used to guide the designer towards which three or four designs would operate most efficiently or take up the least space on the PCB, as possible example objective functions.

8.2.5 Design approach

The overall design approach can be improved in various ways. First, the tool implementation could be combined with circuit simulation tools such as LTSpice, so that designs developed by the design tool could be immediately validated in the simulation. This reduces the amount of required designer intervention. Many circuit simulation tools have their own loss models, so once the final components are selected, those component parameters could be input into the circuit simulation, and the results of the simulation would serve as an additional theoretical performance metric. Alternatively, the component parameters at each iteration of the optimization algorithm could be run through the circuit simulation, and the performance based on the simulation could guide the optimization algorithm towards the minimization of the objective function.

Second, this design tool could be combined with PCB layout software, such that a layout could be automatically created given the selected components. Combining this tool with PCB layout software would save the designer the step of searching for each component in the manufacturer part search section of the PCB design software and then placing each component.

Third, a graphical user interface (GUI) could be developed for this design tool. The GUI would be structured to have a place for a designer to input their power loss model based on an example template, or they could select a topology from one already in the library via a drop-down menu. There would also be a place for the designer to specify all of their design requirements, as discussed in Section 2.2. Then there would be a button to click 'run', and the tool would run the optimization, return the optimized parameters or, if a range of constraints or design parameters was given, a plot. Then the tool would proceed to run the component selection and give the designer lists of manufacturer part numbers. Adding a GUI would make running the tool easier for designers.

Fourth, developing a way to account for the uncertainty in the ML models would improve the accuracy of the overall approach. As discussed in Chapter 5, there is

an inherently high standard deviation in the data of the ML models given the set of inputs and outputs. This uncertainty affects the recommended optimized parameter efficacy when recommending an optimal set of components. There is also uncertainty in the developed power loss model, and how well it represents the actual losses. If these two sources of uncertainty were accounted for in the approach, the uncertainty could be returned to the user so they have a better sense of how reliable the results are. Furthermore, figuring out a way to offset the uncertainty would lead to overall better part recommendations to achieve the objective function.

Other heuristic combinatorial optimization algorithms could also be used to solve this design problem, and the results from these approaches could be compared to those of the ML-based design approach. For example, a greedy search could be implemented. In greedy search algorithms, the best component is always selected at each step, without considering the overall results of other options. The greedy search algorithm works in a top-down approach. This could be implemented to solve the component selection problem presented here by creating a FOM representative of each device, and then creating an order of which components to select first, for example L_1, then Q_1, then Q_2, and always selecting which component has the best FOM while also still allowing the design to meet the constraints. This would be a simple combinatorial optimization solution, but would not consider the fact that perhaps a slight trade-off in what is viewed as the 'best' inductor according to the FOM, in order to leave more cost allowed for the transistors, could result in an overall lower total P_{loss}/P_{out}.

A second combinatorial optimization algorithm that could be used is simulated annealing. This algorithm mimics the annealing of metals in order to find a minimum. At each step, the algorithm generates a new solution according to a probabilistic or random distribution, and decides whether to keep or toss the current solution. A solution with a worse minimization of the objective function is kept with a certain probability, which helps prevent against converging to local minima rather than global minima. As the 'temperature' drops, the algorithm refines its search and becomes more selective

of acceptance, ultimately converging when the temperature is low enough. Simulated annealing could be used to solve the presented problem by swapping through components based on what components are similar to the current solution of parameter values, while still taking some step in a parameter values and evaluating the results of a step in that direction. The algorithm would then continue to swap components, slowly cooling the temperature to only accept better solutions as the overall power loss is minimized.

Additional complexities could also be added into the tool, such as taking into account more information about the PCB layout including traces and board materials, or the required spacing margins between components. Also, currently, the component costs are single-unit distributor quotes, and using the data on bulk orders would show a better representation of cost because usually these components are bought in bulk. In addition, more cost options could be considered by combining the results from various distributors and selecting the best cost. These methods would improve the tool's representation of the total PCB area taken by the components in the design, and the total cost spent on the components and the PCB. However, this would make running the tool more complex, so these could be included as design options. The tool is currently structured such that it can be modular, and one can make their design as complex or simple as the template allows for, so integrating additional complexities as design options would be feasible.

An additional limitation occurs when extremely tight constraints are placed on the design. This can lead to convergence issues in the optimization algorithm. For example, if the area constraint is very small, and there are many components, the sum of the total area may exceed the area constraint when generating components that meet the cost constraint. It would be useful to add a function to return to the user after some number of iterations saying that their constraint is too tight and the design is not feasible.

8.2.6 Conclusions about future work

There are many future steps that can be taken to further develop this research. Each step in the presented design approach can be further elaborated upon. More data can be added to the datasets, which will allow for more accurate ML models, design-oriented parameter estimations, and more options in the exhaustive search. The design-oriented parameter estimations, particularly the inductor ac loss estimation method, can be further developed. The ML models can continue to be adjusted, and neural nets may be further considered. Additional hardware testing would provide insight into how the approach performs given other design objectives and topologies. Finally, the tool can be integrated into other automated circuit simulation and design process tools to create a more complete automated design and testing process for switching power converter design.

8.3 Conclusions

This chapter concludes this thesis by summarizing the presented design approach as well as each element within the approach and its specific research contribution. This chapter also includes an in-depth discussion of future research objectives. Improving the various components of this research presented in Chapters 2–7 would improve the ML-based design approach, which was the major contribution of this research. In conclusion, this thesis presented a successful implementation of a systematic, machine learning-based, automated design approach to design switching power converters. This approach was incorporated into a tool that is structured to handle the optimization of an objective function subject to constraints, and quickly return specific manufacturer part numbers for a given power converter design.

10

EasyEDA - Online PCB design & circuit simulator. https://easyeda.com/. [Accessed 06-09-2024].

Selenium Overview. https://www.selenium.dev/documentation/overview/. [Accessed 06-09-2024].

2002.

Jon Azurza, Grayson Zulauf, Johann Kolar, and Gerald Deboy. New figure-of-merit combining semiconductor and multi-level converter properties. *IEEE Open Journal of Power Electronics*, PP:1–1, 08 2020.

B. Jayant Baliga. *Fundamentals of Power Semiconductor Devices*. Springer, 2008.

B.J. Baliga. Power semiconductor device figure of merit for high-frequency applications. *IEEE Electron Device Letters*, 10(10):455–457, 1989.

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

Daniel Christen and Jürgen Biela. Analytical switching loss modeling based on datasheet parameters for MOSFETs in a half-bridge. *IEEE Transactions on Power Electronics*, 34(4):3700–3710, 2019.

Davide Cittanti, Enrico Vico, and Iustin Radu Bojoi. New fom-based performance evaluation of 600/650 V SiC and GaN semiconductors for next-generation EV drives. *IEEE Access*, 10:51693–51707, 2022.

Daniel Costinett, Dragan Maksimovic, and Regan Zane. Circuit-oriented treatment of nonlinear capacitances in switched-mode power supplies. *IEEE Transactions on Power Electronics*, 30(2):985–995, 2015.

Daniel Delahaye , Supatcha Chaimatanan, and Marcel Mongeau. Simulated annealing: From basics to applications, Jan 1970.

R. Erickson, M. Madigan, and S. Singer. Design of a simple high-power-factor rectifier based on the flyback converter. In *Fifth Annual Proceedings on Applied Power Electronics Conference and Exposition*, pages 792–801, 1990.

R. Erickson and D. Maksimovic. *Fundamentals of Power Electronics*. Springer, 3rd edition, 2020.

Faete Filho, Yue Cao, and Leon M. Tolbert. 11-level cascaded H-bridge grid-tied inverter interface with solar panels. In *2010 Twenty-Fifth Annual IEEE Applied Power Electronics Conference and Exposition*

(APEC), pages 968–972, 2010.

Nikolas Förster, Philipp Rehlaender, Oliver Wallscheid, Frank Schafmeister, and Joachim Böcker. An open-source transistor database and toolbox as a unified software engineering tool for managing and evaluating power transistors. In *IEEE APEC*, pages 1747–1751, 2022.

K. J. Goodrick, A. Butler, Tom Byrd, and D. Maksimović. Machine learning estimators for power electronics design and optimization. In *2021 IEEE Design Methodologies Conference (DMC)*, pages 1–8, 2021.

Kyle J. Goodrick, Jianglin Zhu, and Dragan Maksimović. Systematic optimization of multiple-output dc distribution architectures. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 9(5):5703–5717, 2021.

Thomas Guillod, Panteleimon Papamanolis, and Johann W. Kolar. Artificial neural network (ANN) based fast and accurate inductor modeling and design. *IEEE Open Journal of Power Electronics*, 1:284–299, 2020.

C.P. Henze, H.C. Martin, and D.W. Parsley. Zero-voltage switching in high frequency power converters using pulse width modulation. In *APEC '88 Third Annual IEEE Applied Power Electronics Conference and Exposition*, pages 33–40, 1988.

Matthias Kasper, Ralph M. Burkart, Gerald Deboy, and Johann W. Kolar. ZVS of power MOSFETs revisited. *IEEE Transactions on Power Electronics*, 31(12):8063–8067, 2016.

Il-Jung Kim, S. Matsumoto, T. Sakai, and T. Yachi. New power device figure of merit for high-frequency applications. In *Proceedings of International Symposium on Power Semiconductor Devices and IC's: ISPSD '95*, pages 309–314, 1995.

S.B. Kjaer, J.K. Pedersen, and F. Blaabjerg. A review of single-phase grid-connected inverters for photovoltaic modules. *IEEE Transactions on Industry Applications*, 41(5):1292–1306, 2005.

P.O. Lauritzen and C.L. Ma. A simple diode model with reverse recovery. *IEEE Transactions on Power Electronics*, 6(2):188–191, 1991.

Haoran Li, Seungjae Ryan Lee, Min Luo, Charles R. Sullivan, Yuxin Chen, and Minjie Chen. MagNet: A machine learning framework for magnetic core loss modeling. In *2020 IEEE 21st Workshop on Control and Modeling for Power Electronics (COMPEL)*, pages 1–8, 2020.

Haoran Li, Diego Serrano, Thomas Guillod, Evan Dogariu, Andrew Nadler, Shukai Wang, Min Luo, Vineet Bansal, Yuxin Chen, Charles R. Sullivan, and Minjie Chen. MagNet: An open-source database for data-driven magnetic core loss modeling. In *IEEE APEC*, pages 588–595, 2022.

Haoran Li, Diego Serrano, Thomas Guillod, Shukai Wang, Evan Dogariu, Andrew Nadler, Min Luo, Vineet Bansal, Niraj K. Jha, Yuxin Chen, Charles R. Sullivan, and Minjie Chen. How MagNet: Machine learning framework for modeling power magnetic material characteristics. *IEEE Transactions on Power Electronics*, 38(12):15829–15853, 2023.

Xinze Li, Xin Zhang, Fanfan Lin, and Frede Blaabjerg. Artificial-intelligence-based design for circuit parameters of power converters. *IEEE Transactions on Industrial Electronics*, 69(11):11144–11155, 2022.

Magnetics. Powder cores. `https://www.mag-inc.com/Products/Powder-Cores`.

Wilmar Martinez and Camilo Cortes. Improved design automation combinatorial algorithms for multi-objective optimization of high voltage-gain dc-dc converters. In *2021 23rd European Conference on Power Electronics and Applications (EPE'21 ECCE Europe)*, pages P.1–P.8, 2021.

Christoph Marxgut, Florian Krismer, Dominik Bortis, and Johann W. Kolar. Ultraflat interleaved triangular current mode (TCM) single-phase PFC rectifier. *IEEE Transactions on Power Electronics*, 29(2):873–882, 2014.

G. Mclachlan. Mahalanobis distance. *Resonance*, 4:20–26, 06 1999.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

M. J. D. Powell. A view of algorithms for optimization without derivatives. In *Report DAMTP 2007/NA03. CMS, University of Cambridge, Cambridge CB3 0WA, UK*, April 2007.

Skye Reese, Thomas Byrd, John Haddon, and Dragan Maksimovic. Machine learning-based component figures of merit and models for dc-dc converter design. In *2022 IEEE Design Methodologies Conference (DMC)*, pages 1–6, 2022.

Skye Reese and Dragan Maksimovic. An approach to dc-dc converter optimization using machine learning-based component models. In *2022 IEEE 23rd Workshop on Control and Modeling for Power Electronics (COMPEL)*, pages 1–8, 2022.

Skye Reese, Bailey Sauter, Subodh Khandelwal, Aakash Kamalapur, Thomas Byrd, John Haddon, and Dragan Maksimovic. Loss estimation and design of dc-dc converters using physics- and data-based component models. In *2023 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 82–89, 2023.

Skye Reese, Bailey Sauter, Abhinav Kumar, Shoudong Hu, Vignesh Iyer, and Dragan Maksimovic. Machine learning-aided design of switched-mode power converters. In *2024 IEEE Energy Conversion Congress and Expo (ECCE)*, pages 1–7, 2024.

Subham Sahoo and Frede Blaabjerg. Data-driven controllability of power electronics under boundary conditions - a physics-informed neural network based approach. In *2023 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 2801–2806, 2023.

Charles R. Sullivan, Bradley A. Reese, Aaron L. F. Stein, and Phyo Aung Kyaw. On size and magnetics: Why small efficient power inductors are rare. In *2016 International Symposium on 3D Power Electronics Integration and Manufacturing (3D-PEIM)*, pages 1–23, 2016.

Charles R. Sullivan, Bradley A. Reese, Aaron L. F. Stein, and Phyo Aung Kyaw. On size and magnetics: Why small efficient power inductors are rare. In *2016 International Symposium on 3D Power Electronics Integration and Manufacturing (3D-PEIM)*, pages 1–23, 2016.

Chandan Suthar, V. Inder Kumar, Vahid Yousefzadeh, Montu Doshi, and Dragan Maksimović. A family of two-module composite dc-dc converters. In *2022 IEEE 23rd Workshop on Control and Modeling for Power Electronics (COMPEL)*, pages 1–8, 2022.

Fanghao Tian, Diego Bernal Cobaleda, and Wilmar Martinez. Artificial-intelligence based dc-dc converter efficiency modelling and parameters optimization. In *2022 24th European Conference on Power Electronics and Applications (EPE'22 ECCE Europe)*, pages 1–7, 2022.

Fanghao Tian, Diego Bernal Cobaleda, and Wilmar Martinez. Automatic data extraction based on semiconductor datasheet for design automation of power converters. In *2022 International Power Electronics Conference (IPEC-Himeji 2022- ECCE Asia)*, pages 922–927, 2022.

Fanghao Tian, Diego Bernal Cobaleda, Hans Wouters, and Wilmar Martinez. Parameter design optimization for dc-dc power converters with deep reinforcement learning. In *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1–7, 2022.

C.D. Townsend, T.J. Summers, and R.E. Betz. Control and modulation scheme for a cascaded H-bridge multi-level converter in large scale photovoltaic systems. In *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 3707–3714, 2012.

K. Venkatachalam, C.R. Sullivan, T. Abdallah, and H. Tacca. Accurate prediction of ferrite core loss with nonsinusoidal waveforms using only steinmetz parameters. In *2002 IEEE Workshop on Computers in Power Electronics, 2002. Proceedings.*, pages 36–41, 2002.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau,

Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

Vishay. Si7450dp datasheet, 2024. [Online; accessed November 5, 2024].

Deqiang Wang, Matthias Preindl, Fei Peng, Jin Ye, and Ali Emadi. DC-bus design with hybrid capacitor bank in single-phase PV inverters. In *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 2425–2430, 2017.

Hongfang Wang, Fred Wang, and Junhong Zhang. Power semiconductor device figure of merit for high-power-density converter design applications. *IEEE Transactions on Electron Devices*, 55(1):466–470, 2008.

Song Wang, Yi Murphey, Wencong Su, Mengqi Wang, VAN HAI BUI, Fangyuan Chang, Can Huang, Lingxiao Xue, Felipe Leno da Silva, and Ruben Glatt. An intelligent system for automatic selection of dc-dc converter topology with optimal design. In *AAAI Workshop on AI for Design and Manufacturing (ADAM)*, 2022.

Bailu Xiao, Lijun Hang, Jun Mei, Cameron Riley, Leon M. Tolbert, and Burak Ozpineci. Modular cascaded H-bridge multilevel PV inverter with distributed MPPT for grid-connected applications. *IEEE Transactions on Industry Applications*, 51(2):1722–1731, 2015.

Zichao Ye, Seth R. Sanders, and Robert Carl Nikolai Pilawa-Podgurski. Modeling and comparison of passive component volume of hybrid resonant switched-capacitor converters. *IEEE Transactions on Power Electronics*, 37(9):10903–10919, 2022.

Mengfan Zhang, Xiongfei Wang, and Qianwen Xu. Data-driven modeling of power-electronics-based power system considering the operating point variation. In *2021 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 3513–3517, 2021.

Shuai Zhao, Frede Blaabjerg, and Huai Wang. An overview of artificial intelligence applications for power electronics. *IEEE Transactions on Power Electronics*, 36(4):4633–4658, 2021.