Machine Learning Enabled Fast Multi-Objective Optimization for Electrified Aviation Power System Design

Derek Jackson (*), *Student Member IEEE*
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR, USA
jacksder@oregonstate.edu

Syrine Belakaria (*)
School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA, USA
syrine.belakaria@wsu.edu

Yue Cao, *Member IEEE*
School of EECS (EE)
Oregon State University
Corvallis, OR, USA
Yue.Cao@oregonstate.edu

Janardhan Rao Doppa
School of EECS (CS)
Washington State University
Pullman, WA, USA
jana.doppa@wsu.edu

Xiaonan Lu, *Member IEEE*
Department of ECE
Temple University
Philadelphia, PA, USA
xiaonan.lu@temple.edu

***Abstract*—With the rise of more electric and all-electric aviation power systems, engineering efforts of system optimization shift to the electrical domain. A substantial amount of time and resources are dedicated to finding the best system architecture and design specifications to meet energy efficiency goals and physical constraints. Current processes utilize models of power system components to determine the optimal designs. However, such modeling is computationally expensive as numerous iterations are required to settle on an optimal design. This paper proposes a machine learning (ML) enabled constrained multi-objective optimization solver to drastically reduce the amount of design iterations required for Pareto set discovery for power systems. The process contributes significantly to design automation. A heavy-duty vertical-takeoff-landing (VTOL) unmanned aerial vehicle (UAV) power system is selected to demonstrate the efficacy and limitation of ML enabled optimization. Two extreme trials were run: 1) a search throughout the entire design space with only 9% valid designs within constraints; 2) a search throughout the valid design space. While Trial 1 was unsuccessful in discovering the Pareto front, Trial 2 uncovered all Pareto optimal designs with a 99% reduction of iterations compared to a brute force method.**

***Keywords*—Power Electronics, Power System Design, Machine Learning, Multi-Objective Optimization, Design Automation, Pareto Front, Aviation, UAV, VTOL**

I. Introduction

Modern aviation power systems are trending towards more electric or all-electric. Boeing 787, a more electric aircraft (MEA), has been in use for a few years; Amazon Prime Air autonomous unmanned aerial vehicles (UAVs) delivery has become a reality; NASA N+3 electric passenger aircraft is under development, to name a few [1]. As aerial vehicles become more electrified, power systems will require multiple power electronic converters, electric machines, energy storage, wiring, and cooling devices. Optimizing system performance, such as energy efficiency, weight, and size, can be approached at a component level or a system level. While recent technological advancements have allowed for power converters to reach 99%+ efficiency [2], such designs may not be feasible in any application due to size, weight, temperature constraints, or integration requirement with the rest of the system. Therefore, proper sizing of individual components is critical for system-level optimization.

For an aircraft's design, an electric power system team usually works with a multi-disciplinary optimization (MDO) team by providing necessary modeling pieces of subsystems, including power converters, electric machines, batteries, and so on. The MDO team then runs the models on numerous combination scenarios, sometimes on the order of thousands, either through brute force looping or Monte Carlo search [3]. Depending on the complexity or interface with other physical models, such simulation runs can take from hours to several days, to map the entire design space. After the simulations, engineers determine a Pareto front to find the system optimality, usually circling a few final candidates and then picking one by experience. When a subsystem model is modified, or a new mission is imposed, such iteration processes occur several times before an ultimate design can be generated. This represents a substantial amount of engineering time and effort.

Computer algorithms for solving general Multi-Objective Optimization (MOO) problems with a reduced number of simulations exist, discussed in [4]-[5]. Machine learning (ML) algorithms have been investigated and show promising results to reduce the time and resources for Pareto set discovery [6-11]. Circuit level design of power electronics utilizing machine learning has achieved a 90% reduction in the number of simulations needed to optimize design parameters [12]. However, there has been little research in the development of a machine learning-based search method to optimize design parameters at the system level of a power system.

This paper proposes a machine learning algorithm called ***M**ax-value **E**ntropy **S**earch for **M**ulti-objective **O**ptimization with **C**onstraints (MESMOC)* to reduce the number of simulation iterations required to discover the Pareto set. Experiments using a prior version of the ML algorithm without constraints, i.e., MESMO, consistently outperform state-of-the-art algorithms at providing an accurate, computationally-efficient, and robust optimization solver [9]. This work builds upon MESMO with

(*) indicating co-first authors

Belakaria and Doppa's work was supported by the U.S. National Science Foundation (NSF) grant IIS-1845922.

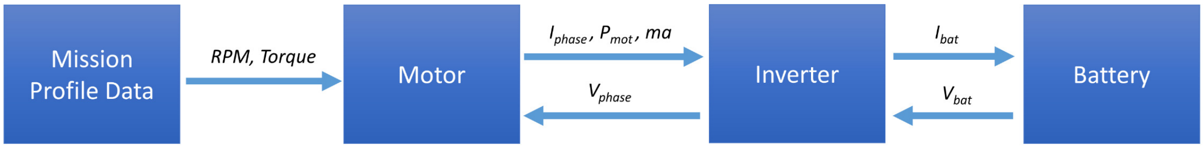

Fig. 1. Block diagram of UAV power system showing the interfaces between each component model.

the development of the constraints add-on. The ML algorithm requires a physical model. A high level static or averaged model is developed for each component in the power system, including multiple physics domains in electrical, thermal, and mechanical. Plenty of research exists on the development and experimental validity of both static and dynamic models of power subsystems. A multi-timescale parametric electrical battery model is described in [13], and [14]-[16] demonstrates the integration of multiple subsystems for UAVs and MEA. Once the physical models are combined to form a desired system architecture, MESMOC can treat the simulation as a black-box function where the outputs are optimization objectives and constraints, and the design parameters serve as the function inputs. MESMOC then evaluates the input design parameters to maximize the information gain about the optimal Pareto front in each iteration until an optimal Pareto set is found.

Due to the scope of this paper, the focus will be on the optimization of a vertical-takeoff-landing (VTOL) UAV power system. However, the proposed ML-based power system design approach can be abstracted to a variety of complex applications, such as MEA, on/off-road vehicles, ships, grid-connected buildings, renewable energy systems, etc. This paper, using a UAV system, serves to demonstrate the efficacy of the approach, especially, the drastic reduction of the number of simulation iterations towards converging to an optimal design.

II. Power System Modeling and Simulation

As the intent of this paper is to explore the use of ML to reduce the number of simulations for electrical power system optimization, technical details of the power system models will not be the focus but are discussed in separate literature [14]-[16]. Actually, the type of power system modelling is of little importance as the optimization algorithm treats it as a black-box function. Still some level of the physical understanding allows to better comprehend the application of the ML algorithm. In this paper, a time-based static simulation capturing averaged power calculations on the order of seconds is used.

The UAV system architecture consists of a central Li-ion battery pack, hex-bridge DC-AC inverters, PMSM motors, and necessary wiring, as shown in Fig. 1. A set of variable design parameters, such as the battery pack configuration and motor size, are included in the system models. This set, known as the design space, will be searched by the machine learning algorithm to find the optimal designs. Table 1 summarizes the design parameters and their sweeping ranges.

A mission profile, defining the workload of the UAV, is used for every simulation run. The mission profile, shown in Fig. 2, contains 30 minutes of thrust values normalized to the total craft mass, representing a flight to and from a location. The mass of the craft frame and its cargo is held constant for all designs. Additional mass is added to the total craft mass depending on the number of cells in the battery pack, motor sizing, and the number of motors. A single motor and an inverter are simulated to reduce repetitive calculations, where the number of motors N_{motor} scales thrust and battery current demand. A single simulation time-step begins with the power and current requirements for the motor and propagates through the inverter to the battery to calculate the next time-step's battery voltage and SOC. Note that power electronics mass is assumed constant for this study, since the semiconductor weight variation is relatively small. Other design details may be included, such as heat sinks or filters. However, this paper focuses on the development of the ML-physical integrated framework rather than a high-fidelity model.

TABLE I. UAV Design Space Ranges

Design Parameter	***Range***
Battery cells in series, N_{series} (#)	[8:18]
Battery cells in parallel, $N_{parallel}$ (#)	[12:96]
Quantity of motors, N_{motors} (#)	[6:10]
Motor stator winding length, h_{stator} (mm)	[4:32]
Motor stator winding turns, N_{turns} (#)	[145:250]

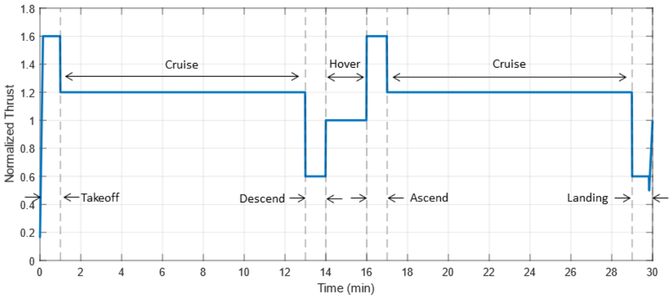

Fig. 2. Mission profile thrust values for UAV simulation.

To represent the design of a PMSM motor, two parameters control the motor sizing: 1) N_{turns} is the number of wire-turns for a stator coil; 2) h_{stator} is the height of the stator coils and governs the machine fill factor. The number of pole pairs is held constant for this study. Motor design is accomplished by perturbing reference values of stator resistance, synchronous inductance, back EMF constant, and mass using the aforementioned design parameters. The motor reference values are experimentally measured from a specific PMSM family. For a given N_{turn} and h_{stator}, the maximum wire gauge for the stator is selected while still satisfying the fill factor limit.

Algorithm 1 MESMOC Algorithm

```
Input: input space $\mathfrak{X}$; $K$ blackbox objective functions $f_1(x), f_2(x), \cdots, f_K(x)$; $L$ blackbox constraints $C_1(x), C_2(x), \cdots, C_L(x)$; and maximum no. of iterations $T_{max}$
Initialize Gaussian process models $\mathcal{M}_{f_1}, \mathcal{M}_{f_2}, \cdots, \mathcal{M}_{f_K}$ and $\mathcal{M}_{c_1}, \mathcal{M}_{c_2}, \cdots, \mathcal{M}_{c_L}$ by evaluating at $N_0$ initial points
for each iteration $t$ = $N_0 + 1$ to $T_{max}$ do
    Select $\mathbf{x}_t \leftarrow \arg max_{\mathbf{x} \in \mathfrak{X}} \quad \alpha_t(\mathbf{x})$
            s.t $(\mu_{c_1} \geq 0, \cdots, \mu_{c_L} \geq 0)$
    $\alpha_t(.)$ is computed as:
        for each sample $s \in 1, \cdots, S$:
            Sample $\tilde{f}_i \sim \mathcal{M}_{f_i}, \quad \forall i \in \{1, \cdots, K\}$
            Sample $\tilde{C}_i \sim \mathcal{M}_{c_i}, \quad \forall i \in \{1, \cdots, L\}$
            //Find Pareto front of cheap MOO over $(\tilde{f}_1, \cdots, \tilde{f}_K)$ constrained by $(\tilde{C}_1, \cdots, \tilde{C}_L)$
            $\mathcal{Y}_s^* \leftarrow \arg min_{x \in \mathcal{X}} (\tilde{f}_1, \cdots, \tilde{f}_K)$
                s.t $(\tilde{C}_1 \geq 0, \cdots, \tilde{C}_L \geq 0)$
        Compute $\alpha_t(.)$ based on the $S$ samples of $\mathcal{Y}_s^*$ as given in Equation (1)
    Evaluate $\mathbf{x}_t$; $\mathbf{y}_t \leftarrow (f_1(\mathbf{x}_t), \cdots, f_K(\mathbf{x}_t))$ ;$\mathbf{c}_t \leftarrow (C_1(\mathbf{x}_t), \cdots, C_L(\mathbf{x}_t))$
    Aggregate data: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{x}_t, \mathbf{y}_t, \mathbf{c}_t)\}$
    Update models $\mathcal{M}_{f_1}, \mathcal{M}_{f_2}, \cdots, \mathcal{M}_{f_K}$ and $\mathcal{M}_{c_1}, \mathcal{M}_{c_2}, \cdots, \mathcal{M}_{c_L}$
    $t \leftarrow t + 1$
end for
return Pareto front of $f_1(x), f_2(x), \cdots, f_K(x)$ based on $\mathcal{D}$
```

Modeling of the motor follows the work from [15]. Calculations of the back EMF, electrical and mechanical power losses, and modulation index depend on the motor RPM, torque, and the voltage level of the inverter. Motor RPM and torque come from the mission profile thrust values, where an experimentally derived curve fit relates the thrust to RPM and torque. Steady-state motor temperature is calculated from power losses, surface area of the motor, and heat transfer coefficient approximations. For system integration, per-phase current, required modulation index, and total power used are sent to the inverter model.

The DC-AC inverter uses a three-phase hex-bridge topology with specifications from the datasheet of a Toshiba TPH4R008NH MOSFET and TPH4R10ANL as the diode. Similar to [15], PWM based switching power loss calculations use an averaged switching current, derived from the current of one motor phase. Methods to find switching voltage rise and fall times are from [17], which results in worst-case scenario estimations. MOSFET and diode conduction losses depend on the one-phase current and modulation index from the motor model. Adding various power losses to the motor output power, battery current is found. An accurate model of a Li-ion battery pack must consider the SOC, terminal voltage, and the current demand. This simulation uses a multi-timescale parametric electric battery model, based on [14]. The work builds upon Randle's equivalent circuit and utilizes multiple RC time constants to model the transient behavior of a cell's terminal voltage and provides the battery cell impedance for power loss calculations. In this model, voltage and impedance are dependent on SOC. There are two design parameters considered in the battery: 1) N_{series} represents the number of cells in series for one stack, i.e., the battery pack voltage. 2) $N_{parallel}$ represents the number of cell stacks connected in parallel. Inverter current demand serves as an input to the battery and is used along with $N_{parallel}$ to determine the current of an individual cell. Individual cell current, along with voltage and internal resistance, is used to calculate power losses for each time-step. The new SOC and battery pack voltage can then be derived from the energy consumed and current demand of the present time-step.

III. MESMOC Algorithm

To overcome the computational overhead of determining an optimal power system design, MESMOC aims to reduce the number of simulations required to find the Pareto optimal set of solutions. A solution is called Pareto optimal when one objective cannot be improved without compromising another objective. In the context of power system design, a Pareto optimal set contains designs that have the best combination of total energy consumption, weight, and cost.

By treating the physical models and simulation as a black-box function, Bayesian Optimization (BO) [17], an effective framework for solving expensive black-box function evaluations, can be employed. BO begins by building cheap surrogate models (e.g., Gaussian Process [18]) using simulation results. Gaussian Processes (GPs) are effective surrogate models for multi-objective BO. The desired objective functions are modeled using K independent GP models. Each surrogate model is learned by using past simulation runs as training data.

$$\alpha(\mathbf{x}) \simeq \frac{1}{S}\sum_{s=1}^{S}\left[\sum_{j=1}^{K}\frac{\gamma_s^{f_j}(\mathbf{x})\phi(\gamma_s^{f_j}(\mathbf{x}))}{2\Phi(\gamma_s^{f_j}(\mathbf{x}))} - \ln\Phi(\gamma_s^{f_j}(\mathbf{x})) + \sum_{j=1}^{L}\frac{\gamma_s^{c_j}(\mathbf{x})\phi(\gamma_s^{c_j}(\mathbf{x}))}{2\Phi(\gamma_s^{c_j}(\mathbf{x}))} - \ln\Phi(\gamma_s^{c_j}(\mathbf{x}))\right] \qquad (1)$$

where $\gamma_s^{c_j}(x) = \frac{y_s^{c_j *} - \mu_{c_j}(\mathbf{x})}{\sigma_{c_j}(\mathbf{x})}$, $\gamma_s^{f_j}(x) = \frac{y_s^{f_j *} - \mu_{f_j}(\mathbf{x})}{\sigma_{f_j}(\mathbf{x})}$, $y_s^{c_j *}$ and $y_s^{f_j *}$ are the maximum values of constraint $\tilde{C}_j$ and function $\tilde{f}_j$ reached after the cheap multi-objective optimization over sampled functions and constraints. ϕ and Φ are the p.d.f and c.d.f of a standard normal distribution respectively.

Subsequently, these models are used to define an acquisition function, which will be used to score the utility of each candidate design and then select the design with highest utility for design evaluation. This acquisition function, defined as α, will be used to intelligently select the next sequence of design parameters for evaluation to accelerate the discovery of the optimal Pareto front. Rather than using an input space entropy-based acquisition function such as the state-of-the-art BO algorithm in [7], MESMOC utilizes an output space entropy-based acquisition function. Output space entropy search allows for much tighter approximations, is significantly cheaper to compute, and naturally lends itself to robust optimization. MESMOC's acquisition function maximizes the information gain about the optimal Pareto front, which is equivalent to expected reduction in entropy over the optimal Pareto front.

Two key algorithmic steps of the acquisition function are (a) computing the Pareto front samples; and (b) computing entropy with respect to a given Pareto front sample. Pareto front samples are computed by sampling the posterior GP models via random Fourier features for use in cheap MOO over the *K* sampled functions. The sample Pareto fronts are used to compute the information gain. The algorithm will select the input design that maximizes the information gain to be evaluated next. The information gain equation is given in equation (1). Details about the mathematical derivation will not be covered due to the scope of the paper, but can be found in [9]. A complete description of the MESMOC algorithm is given in Algorithm 1. The blue colored steps correspond to computation of the output space entropy-based acquisition function via sampling.

IV. Physical Model-ML Integration

Power system design is a constraint heavy optimization problem within a large design space. Even the comparably simple electrical architecture of a UAV can include hundreds of thousands of design parameter combinations. However, the design space consists of many invalid parameter combinations. Some designs may not be capable of providing the motor the necessary power for flight, the power rating of the motors or electronics may be too low and cause overheat, or there is insufficient energy storage to complete the mission, etc. A set of constraints are thus required to ensure a valid design.

Throughout a simulation, each constrained model variable is monitored. For example, a maximum temperature for the motors and inverters is selected to ensure no overheating. The modulation index of the inverter switching control is constrained as the DC bus voltage can only be utilized a limited amount. In the large design space, many designs do not include energy storage capable of supplying the required current demand. Too much current drawn from the battery at any instant will significantly decrease the battery terminal voltage, which in-turn increases the load current further, creating a positive feedback loop. A minimum battery cell voltage is thus required for a valid design. The battery pack must also contain enough energy to complete a mission. A Depth of Discharge (DOD) limit is then set to match the common maximum discharge of Li-ion batteries. Table II outlines the set of constraints used for optimization. With these constraints, the set of valid designs are only a small subset of the entire design space.

As MESMOC treats the simulations as a black-box function, it has no knowledge of what occurs during the simulation. Even if a design exceeds one of the constraints, the algorithm will still use the simulated results in its search of the Pareto front. An invalid design simulation must therefore provide useful information as well. Unfortunately, each electrical subsystem is modeled in a way that assumes valid operation. If a constrained variable exceeds the limit by too much, the models will breakdown and cause the simulation to return results with values that can be misleading to the algorithm, thereby decreasing its ability to find the Pareto front. To mitigate this issue, a 'soft limit' is set for the previously discussed constraints. During a simulation, a constrained variable which exceeds a certain value is suppressed exponentially as to remain within the operating bounds of the model. A hyperbolic tangent-like curve is employed as the 'soft limit' which will only activate when the constrained variables exceed the limits listed in Table II. This technique is more favorable than holding these variables at a 'hard limit' (saturation) as little information of the Pareto front is gained when many unique designs return the same results.

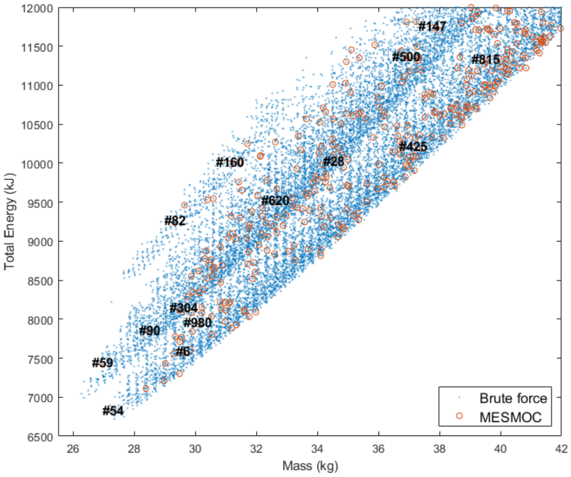

Fig. 3. Mass vs energy plot of the brute force and MESMOC results using the *entire* design space.

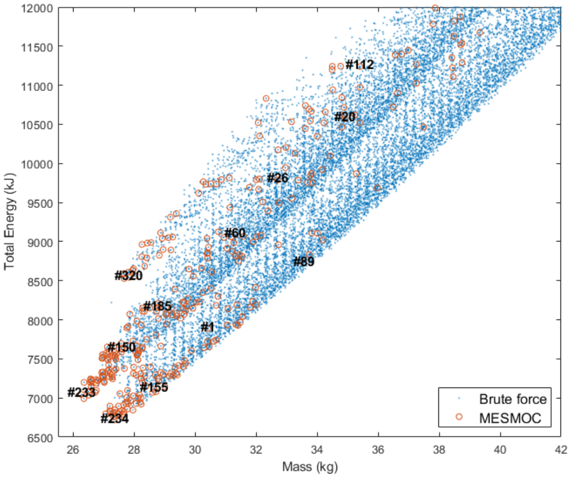

Fig. 4. Mass vs energy plot of the brute force and MESMOC results using the *valid* design space.

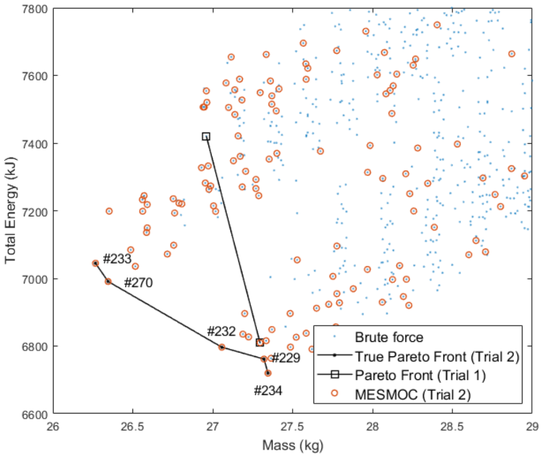

Fig. 5. Mass vs energy plot showing a subset of the designs found by brute force, MESMOC, and the ML discovered Pareto fronts. The iteration number of the Pareto points for trial 1 are included.

At the end of a simulation, results are sent to the machine learning algorithm for analysis. The results include constrained variables along with the optimization objectives. The surrogate models for each objective are updated in MESMOC and the next set of design parameters are chosen, which will maximize the information gain about the Pareto front.

TABLE II. DESIGN CONSTRAINTS AND LIMITS

Constraint	*Design Limit*
Maximum final DOD	75%
Minimum cell voltage	3.0V
Maximum motor temperature	125°C
Maximum inverter temperature	120°C
Maximum modulation index	1.3

V. EXPERIMENTAL RESULTS AND DISCUSSION

To demonstrate the efficacy of the ML-based power system MO optimization solver, minimization of total energy used and vehicle mass are selected as the objectives. For a design to be valid, the simulated UAV must be capable of completing the specified mission without violating the constraints in Table II. A reference base is required to determine the accuracy of the Pareto front found by MESMOC. Thus, a brute force approach simulated a total of approximately 250,000 combinations of design parameters. Out of the entire design space, only 9% of design combinations passed all the constraints. As the valid points are only a small subset of the design space, two separate Pareto search trials were done over the entire set and the subset of passing designs, respectively.

A. Trial 1: Entire Design Space

Employing MESMOC on the entire design space resulted in none of the Pareto optimal points discovered after more than 1000 iterations. Due to the sparsity of valid designs only 50% of the simulated parameters passed the constraints, highlighting the difficulty of this algorithm with a highly constrained design space. Interestingly, the best Pareto front was found within the first 100 iterations. Fig. 3 shows these results compared to the brute force method. The most optimal designs are located at the bottom left corner, as the axes are the two objectives to minimize.

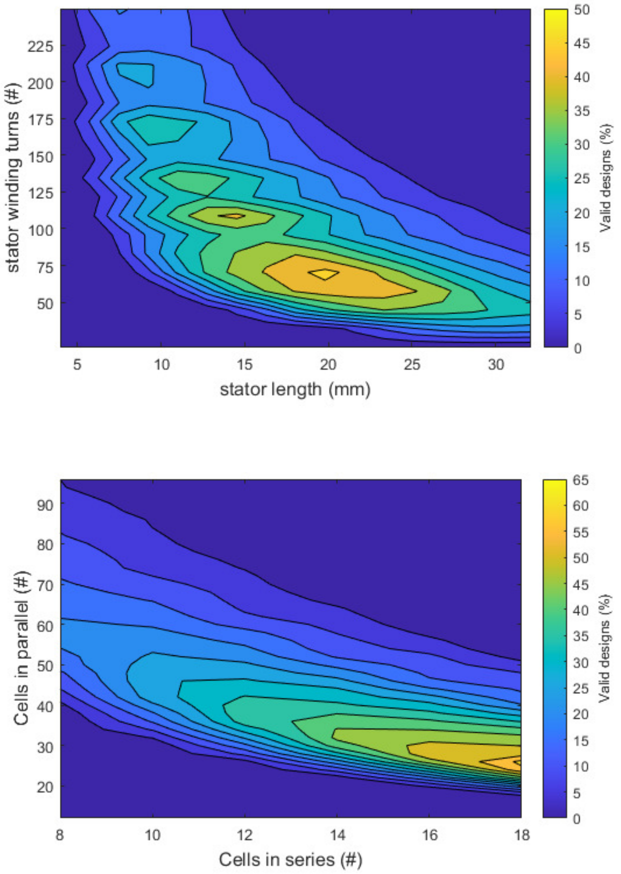

Fig. 6. Percent of designs which pass all constraints for motor sizing parameters (top) and battery sizing parameters (bottom).

B. Trial 2: Valid Design Space

By allowing MESMOC to search the subset of valid design parameters, it discovered all 5 points of the Pareto front in only 270 iterations. This implies that 1% of the iterations are required compared to the brute force. This considerable reduction in ML-based iterations still retains the search accuracy.

Similar to Fig. 3, a mass vs. energy plot in Fig. 4 shows the results of valid designs from the brute force approach along with the points MESMOC tested. A few points are numbered to show the iteration steps of the ML search. The set of points considered as the Pareto front can be seen in Fig. 5, a close up of the optimal design region of the mass vs. energy plot. Fig. 5 also includes the Pareto front found in Trial 1 to show its distance from the true Pareto front.

Note that in these plots the discrete boundaries seen on the distribution of brute force points are due to a few factors: 1) Fig. 3,4,5 show only valid design points and many designs are not valid due to the temperature and DOD constraints, or an inability to provide sufficient voltage to the motor to maintain the necessary RPM. 2) The energy requirement and vehicle mass of n-motored UAVs, all shown in the plot, are inherently different. 3) The battery parameters, $N_{parallel}$ and N_{series}, cause discrete changes to the vehicle mass and therein the total energy used.

C. Discussion

In application, the subset of valid designs is unknown to the optimization algorithm. However, it serves as a demonstration of the capability of MESMOC. In a non-constrained search, MESMOC drastically reduces the number of iterations required to discover the Pareto front.

The primary challenge MESMOC has with this multi-objective optimization is the quantity of invalid points in the design space. Only 50% of the points sampled by the algorithm were valid, with an unsuccessful discovery of the true Pareto front after more than 1000 evaluations. The ratio of valid designs for a given motor size or battery pack configuration, shown in Fig. 6, helps realize the sparsity of the design space. 50.1% of all motor sizes never resulted in a valid design. Within the motor sizes which can result in a valid design, more than half of the possible designs will still fail due to other design parameters. Similarly, 65.4% of battery pack configurations never resulted in a valid design. By removing the chance of sampling an invalid design, all Pareto optimal points were found in 270 iterations, which is about 0.11% of the entire design space. Therefore, the two trials suggest any reduction of invalid designs would improve Pareto front discovery. Note that these two trials are extreme cases for a majority non-valid design space and an all-valid design space.

VI. Conclusions and Future Work

This paper demonstrated the potential of ML algorithms to considerably reduce the number of design iterations to discover the Pareto front for MOO of power electronic systems. Utilizing a physical model simulation, the capability of MESMOC was demonstrated with a search through the entire design space and the valid sub-space. The unsuccessful Pareto front discovery due to 91% of designs failing the constraints showcased the limitations of the algorithm. A more robust optimization search is enabled by reducing the number of invalid designs within the design space. For example, when the search is limited to valid designs only, the capability of MESMOC is demonstrated by a successful Pareto front discovery with a substantial reduction in iterations. This ML enabled search is a promising solution to saving many hours of engineering effort to determine an optimal power system design, although challenges still exist.

Future work to investigate the reliability of the search and ways to improve full Pareto front discovery is under way. A reduction in the number of invalid points in the design space is expected to increase this discovery. Additional work will improve the physical models. The simulation currently utilizes only static component models of the power system. Dynamic models of each power system component are to be added, which will allow for a more detailed analysis of system behaviors such as current and voltage transients, motor controls, and so on.

Acknowledgement

The authors would like to thank Alastair Thurlbeck for his efforts in developing the UAV base models.

References

[1] B. Sarlioglu and C. T. Morris, “More electric aircraft: review, challenges, and opportunities for commercial transport aircraft,” *IEEE Trans. Transportation Electrification*, vol. 1, no. 1, pp. 54-64, 2015.

[2] D. Rothmund, T. Guillod, D. Bortis, J. W. Kolar, “99.1% Efficient 10 kV SiC-Based Medium-Voltage ZVS Bidirectional Single-Phase PFC AC/DC Stage”, *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 7, no. 2, 2019.

[3] S. Ziemer, M. Glas, G. Stenz, “A Conceptual Design Tool for multi-disciplinary aircraft design,” *Aerospace Conf.*, 2011, pp. 1-13.

[4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] Joshua Knowles, “Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation,* vol. 10, no. 1, pp. 50–66, 2006.

[6] A. Deshwal, N. K. Jayakodi, B. K. Joardar, J. R. Doppa, and P. Pande, “MOOS: A Multi-Objective Design Space Exploration and Optimization Framework for NoC enabled Manycore Systems,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 5s, art. no. 77, 2019.

[7] Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. “Predictive entropy search for multi-objective bayesian optimization”. in *Proc. International Conference on Machine Learning (ICML)*, 2016, pp. 1492–1501.

[8] Lyu, Wenlong, et al. "Multi-objective Bayesian optimization for analog/RF circuit synthesis." in *Proc. Design Automation Conference (DAC)*, 2018, pp. 1-6.

[9] S. Belakaria, A. Deshwal, and J. R. Doppa, “Max-value Entropy Search for Multi-Objective Bayesian Optimization,” in *Proc. Advances in Neural Information Processing Systems (NeurIPS) Conference*, 2019, pp. 7823-7833.

[10] S. Belakaria, A. Deshwal, N. Jayakodi, and J. R. Doppa, “Uncertainty-aware Search Framework for Multi-Objective Bayesian Optimization,” in *Proc. of AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 10044-10052.

[11] N. Jayakodi, S. Belakaria, A. Deshwal, J.R. Doppa, “Design and Optimization of Energy-Accuracy Tradeoff Networks for Mobile Platforms via Pretrained Deep Models,” *ACM Transactions on Embedded Computing Systems (TECS)*, 2020, vol. 19, no. 4, pp. 1-24.

[12] Z. Zhou, S. Belakaria, A. Deshwal, W. Hong, J. R. Doppa, P. Pande, and D. Heo, “Design of Multi-Output Switched-Capacitor Voltage Regulator via Machine Learning,” to appear in *Proc. IEEE/ACM Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020, pp. 502-507.

[13] Y. Cao, R. C. Kroeze, and P. T. Krein, “Multi-timescale parametric electrical battery model for use in dynamic electric vehicle simulations,” *IEEE Trans. on Transportation Electrification*, vol. 2, no. 4, pp. 432-442, 2016.

[14] Y. Cao, M. A. Williams, B. J. Kearbey, A. T. Smith, P. T. Krein, and A. G. Alleyne, “20x-Real Time Modeling and Simulation of More Electric Aircraft Thermally Integrated Electrical Power Systems,” in *Proc. IEEE International Conf. Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles* (*ESARS-ITEC*), 2016, pp. 1-6.

[15] A. Thurlbeck and Y. Cao, “Analysis and modeling of UAV power system architectures,” in *Proc. IEEE Transportation Electrification Conf. (ITEC)*, 2019, pp. 1-8.

[16] Y. Cao and A. Thurlbeck, “Heavy-duty UAV Electric Propulsion Architectures and Multi-timescale Multi-physics Modeling,” in *Proc. AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, 2019, pp. 1-13.

[17] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. “Taking the human out of the loop: A review of Bayesian optimization,” in *Proc. IEEE*, 2016, vol. 104, no. 1, pp. 148–175.

[18] Christopher KI Williams and Carl Edward Rasmussen. “Gaussian processes for machine learning,” *MIT Press*, vol. 2, 2006.