AI-Driven Design Optimization of Engineering Systems: A Case Study on Turboshaft Engines

Peeyush Pankaj[1] and Satish Thokala[2]

[1,2]*MathWorks, Bengaluru, Karnataka, 560103, India*

ppankaj@mathworks.com
sthokala@mathworks.com

ABSTRACT

In the pursuit of optimizing complex engineering systems, the exploration and thorough understanding of the design space become imperative, particularly when dealing with multi-objective systems characterized by an array of independent variables. This paper presents a comprehensive analysis on the design space mapping of such intricate systems, utilizing a turboshaft engine as a representative case study.

Our initial methodology phase involves the employment of a physics-based model to generate a synthetic dataset. This dataset reflects the intricate interplay of various system parameters that underpin the engine's operation. The synthesized data serves as a foundation for the subsequent development of a Machine Learning or Deep Learning-based surrogate model. This surrogate AI model is meticulously crafted to encapsulate the multiple inputs and outputs inherent in the turboshaft engine's functioning, thereby facilitating an efficient and accurate exploration of the design space.

The core of our investigation revolves around AI surrogate model utilization for conducting multi-objective optimization. This optimization process is not merely focused on enhancing specific performance metrics but is also geared towards identifying a comprehensive family of feasible design solutions. Such an approach enables the delineation of the entire design space, offering invaluable insights into the trade-offs and synergies among different design objectives. Through this methodology, we can uncover a wide spectrum of viable design alternatives, thereby providing a robust framework for decision-making in the engineering design process.

1. INTRODUCTION

Aerospace system optimization is a critical engineering area focusing on improving the performance, efficiency, and cost-effectiveness of various components and processes within the aerospace industry. This involves applying mathematical and computational techniques to design, analyze, and refine systems such as aircraft, spacecraft, propulsion systems, and avionics.

Design optimization involves identifying the optimal design parameters that meet project specifications. Engineers often employ design of experiments (DOE), statistical analysis, and optimization techniques to assess trade-offs and pinpoint the best design.

In many design scenarios, numerous parameters must be considered. Some parameters may influence performance metrics in a nonlinear manner, while others might only assume discrete values. Furthermore, there are typically multiple, often conflicting, requirements and objectives to satisfy. Adjusting one parameter at a time manually can lead to less-than-ideal outcomes and evaluating every possible option in the design space can be prohibitively time-consuming.

Design optimization tackles these issues by using numerical optimization methods to automatically find the best solutions within given constraints. This approach navigates the design space more efficiently than exhaustive searches. The iterative process of design refinement is automated, reducing the time required and the potential for human error. Engineers also apply statistical techniques to explore parameter sensitivities and gain insights into the design space before and after optimization, ensuring the robustness of the optimal solutions.

An aircraft sub-system, such as the engine, must meet multiple design objectives and a wide array of independent variables. It becomes challenging to execute trade studies and optimize the design in such a way that all the design requirements are satisfied in quick time. Traditional methods of design space exploration can be labor-intensive and inefficient.

This paper introduces an innovative approach that combines physics- based modeling with advanced AI techniques to efficiently map the system design space. Using a turboshaft engine as a case study, we generate synthetic data through a physics-based model, which then informs the development of a Machine Learning or Deep Learning-based surrogate model. This AI surrogate model facilitates accurate multi-objective optimization, uncovering a wide range of feasible design solutions. Our methodology not only enhances performance metrics but also provides deep insights into the trade-offs and synergies between different design objectives. The results demonstrate the potential of this hybrid approach to revolutionize engineering design by offering a robust framework for optimizing complex systems.

2. WORKFLOW

In this research paper, a generic workflow is discussed to systematically approach data generation with experiment design, evaluation of multiple AI models with validation, conducting multi-objective optimization, and mapping the full family of feasible design solutions through a design space map, as shown in Figure 1.

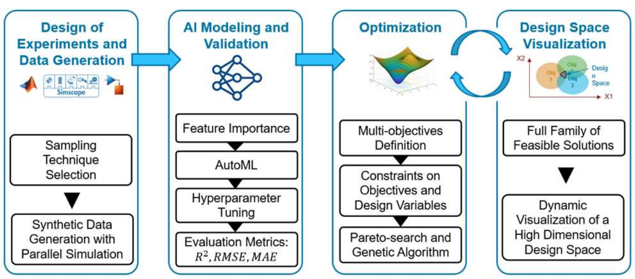

Figure 1. Generic workflow depicting systematic approach to data generation, data driven modeling, optimization, and design space mapping.

The workflow and analysis described in this paper are implemented using MATLAB®. We will highlight the salient aspects of each step of the workflow in the subsequent sections.

2.1. Design of Experiments

The first step to uncovering the best possible engineering product/asset design is to take a systematic approach to data generation, starting with running simulations and experiment design. A physics-based model is created to generate the data.

2.1.1. Physics-based Model for Synthetic Data

A first principles-based turboprop engine model is developed using Simulink® and Simscape™, as shown in Figure 2.

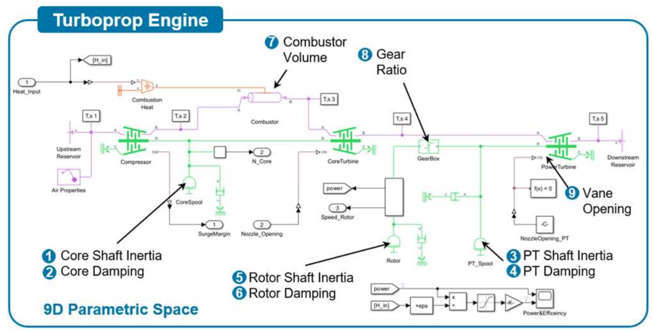

Figure 2. First principles-based turboprop engine model highlighting independent design variables.

This model serves as a representative system for our analysis, rather than an exact replica of an actual engine. The system comprises two aerodynamically coupled spools, the core rotor, the power turbine module, and a propeller shaft connected to the power turbine shaft via a bevel gear. The model incorporates nine independent parameters, including shaft inertias and the structural damping for each of the three rotors, combustor volume, gear ratio, and vane opening. The compressor and turbine maps are pre-defined, as shown in Figure 3.

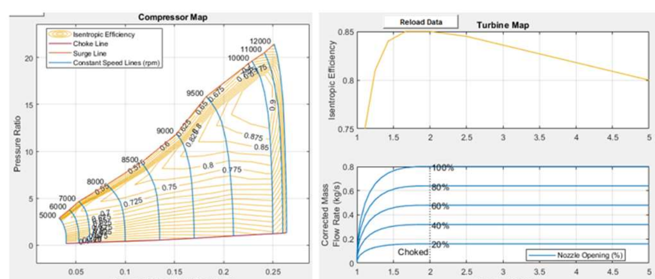

Figure 3. Predefined Compressor and Turbine maps.

These parameters were selected to capture the essential dynamics and interactions within the turboprop system. For this workflow demonstration, we simulated the engine under a throttle ramp and hold condition over a duration of 1200 seconds, as shown in Figure 4. The throttle position is ramped to 600 seconds and held at a constant value for another 600 seconds. During this period, all system objectives were calibrated for steady-state operation when the throttle was in the hold condition.

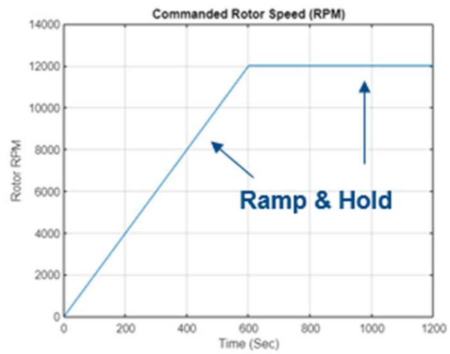

Figure 4. Core rotor speed being plotted with respect to time. Core rotor speed reacts to the ramp and hold throttle position profile.

The outputs monitored during the simulation include core thrust, power turbine thrust, and shaft mechanical power. These outputs provide critical insights into the system's performance and serve as the basis for developing our AI-based surrogate model.
By employing this synthetic dataset, we ensure that our methodology is robust and generalizable, capable of adapting to various complex engineering systems. The generated data encapsulates the intricate system parameter interplay, thereby laying a solid foundation for subsequent optimization and analysis.

2.1.2. Sampling

To systematically explore synthetic data generation, we employed a Design of Experiments (DOE) approach for executing the simulations. One common method in DOE is full factorial sampling, where all possible combinations of the independent parameters are tested. While comprehensive, full factorial sampling is often impractical for complex systems with numerous parameters due to its exponential growth in the number of required simulations, as shown in Table 1. In this case study we have nine independent system parameters. Even if we define three levels for each parameter, the full factorial design sampling will generate 20,000 sampling points, which can lead to excessive computational costs and time inefficiencies.

Table 1. Number of sampling points for a full-factorial design with nine independent parameters and multiple levels on each parameter.

Levels	# of Simulations
10	1B
9	387M
7	40M
5	2M
3	20K

Given the computational constraints, we opted for Latin Hypercube Sampling (LHS), a more efficient and scalable technique. LHS ensures that the entire range of each parameter is sampled, offering a more representative distribution of the design space with fewer sampling points compared to full factorial sampling. For our analysis, we selected 2,000 sampling points using LHS, as referenced in Figure 5. This choice strikes a balance between computational feasibility and the need for thorough exploration of the design space.

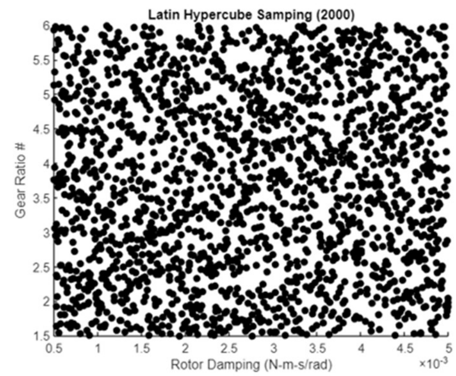

Figure 5. Latin Hypercube Sampling space for nine independent parameters but limited to 2,000 simulations.

Using LHS, we can efficiently generate a diverse simulation set that captures the variability and interactions among the nine independent parameters. This approach enhances the robustness of our synthetic dataset, thereby improving the accuracy and reliability of the subsequent AI-based surrogate model.

2.1.3. Parallel Simulation and Synthetic Dataset

To expedite the simulation process, multiple simulations were executed in parallel, leveraging parallel computing. We significantly reduced the time required to generate the synthetic dataset by utilizing multiple processing cores simultaneously. This parallel execution not only enhances efficiency but also ensures the extensive set of 2,000 sampling points is processed in a timely manner. As a result, we got a 2,000 time-series data table consisting of nine predictors and three responses.

2.2. AI Modeling

In the previous step, a dataset was created concerning each of the nine independent tunable parameters. Now that the dataset is ready, the next step is to train an AI model.

2.2.1. Feature Ranking

In this section, we conducted an analysis to determine the relative importance of various design parameters on system performance. Utilizing the F-Test, a statistical method that assesses each feature's significance in explaining the response variable variance, we identified three parameters as particularly influential: PT Nozzle Opening, Gear Ratio, and PT Damping. These parameters stood out among the nine considered, demonstrating a higher impact on the Power Turbine Thrust, as shown in Figure 6.

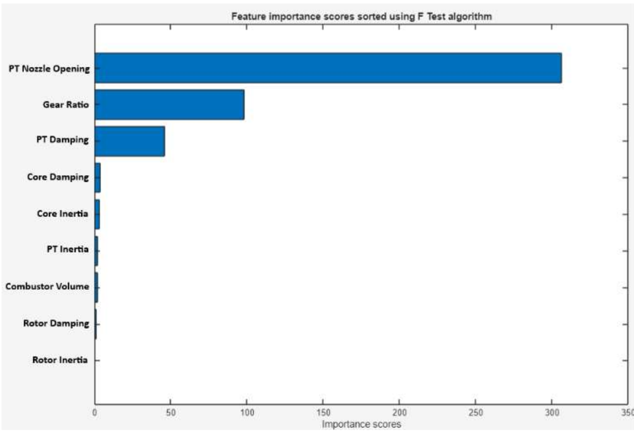

Figure 6. Feature importance scores for Power Turbine Thrust response using F-Test.

The identification of these key parameters is crucial for several reasons. First, it allows for a more focused design space exploration, prioritizing variables that most significantly affect outcomes. Second, we can efficiently solve both the optimization problem and the design space mapping by concentrating on influential parameters. Lastly, understanding the importance of features aids in simplifying the model, potentially reducing complexity without sacrificing predictive accuracy. This insight into feature significance not only informs the design and optimization strategies but also provides a foundation for further investigations into system dynamics.

2.2.2. AutoML

Accurately modeling system responses is crucial in optimizing complex engineering designs. This case study focused on three design objectives: shaft mechanical power, core thrust, and power turbine thrust. We employed automated machine learning (AutoML) to train over 20 regression models for each objective using a synthetic dataset. These models included variants of linear regression, support vector machines, decision trees, ensemble methods, gaussian process regression, and neural networks. A 20% holdout validation set was used to ensure unbiased performance evaluation of the AI model on the unseen data. Models were assessed with adjusted R^2, accounting for the predictor count and offering a precise performance measure. Apart from adjusted R^2 values, we also considered RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) metrics to measure goodness of fit. Figure 7 represents the observed root mean squared error (RMSE) values on the test data for multiple trained machine learning models.

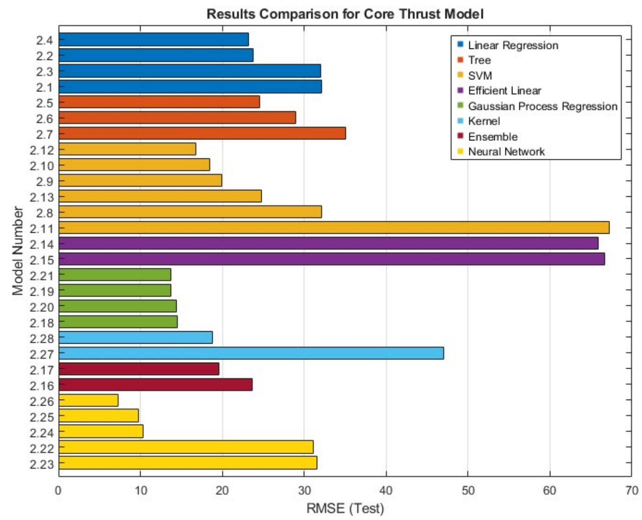

Figure 7. Root Mean Squared Error comparison over test dataset for multiple machine learning models.

2.2.3. Hyperparameter Tuning

As observed in the previous results, the Artificial Neural Networks (ANN) model outperforms others for the core thrust predictions. The trained ANN has three hidden layers with 10 neurons each. This architecture uses ReLU activation level. We set up the hyperparameter tuning experiment with Bayesian Optimization for the ANN architecture while varying the number of fully connected layers, activation function, and regularization strength. The minimum MSE value observed on the validation data is reported with each iteration, as shown in Figure 8.

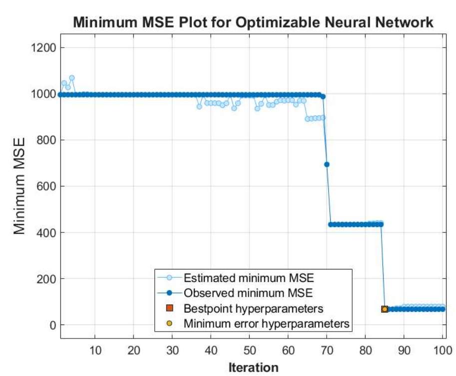

Figure 8. Minimum Mean-Squared-Error plot regarding Iterations during Hyperparameter Tuning of the ANN model.

The MSE value dropped from 52.58 to 45.97 for the optimized network, and the RMSE value improved from 7.25 to 6.78 on the test data as reported in Table 2, whereas the MAPE is similar to the unoptimized network. The optimized network has two fully connected layers with 20 neurons in the first layer and one neuron in the second layer, with ReLU activation and regularization strength set as 6.5e-5.

Table 2 Results Summary table for multiple machine learning models trained on core thrust response.

Model Type	Training				Test			
	RMSE	MSE	R^2	MAPE	MSE	RMSE	R^2	MAPE
ANN optimized	19.07	363.60	0.9343	5.32%	45.97	6.78	0.9910	1.91%
ANN Trilayered	15.10	227.98	0.9588	3.52%	52.58	7.25	0.9897	1.83%
ANN Bilayered	16.68	278.33	0.9497	4.26%	96.13	9.80	0.9811	2.55%
ANN Wide	10.44	109.07	0.9803	2.99%	105.72	10.28	0.9793	2.81%
GPR Rationale Quadratic	14.70	215.99	0.9609	4.37%	186.97	13.67	0.9633	3.87%
GPR Matern 5/2	14.74	217.31	0.9607	4.42%	188.91	13.74	0.9630	3.91%
GPR Exponential	16.06	257.89	0.9534	4.77%	207.15	14.39	0.9594	4.1 %
GPR Exponential^2	15.50	240.30	0.9566	4.75%	208.97	14.46	0.9590	4.22%
SVM Medium Gaussian	18.60	345.99	0.9374	5.81%	282.17	16.80	0.9447	5.12%
SVM Cubic	18.72	350.62	0.9366	5.75%	341.37	18.48	0.9331	5.75%
Kernel Least Squares Regression	21.91	480.18	0.9132	6.97%	355.04	18.84	0.9304	5.91%
Ensemble Bagged Trees	22.09	488.12	0.9117	6.61%	383.63	19.59	0.9248	5.74%

ANN excelled over other family of models, achieving adjusted R^2 values of 0.98 and above on both training and test datasets for all three design objectives, as shown in Figure 9 below.

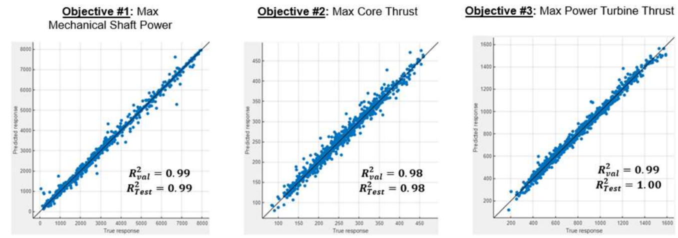

Figure 9. Predicted response (y-axes) vs. Ground Truth (x-axes) for all three design objectives.

While the machine learning models demonstrate high accuracy when comparing predicted values to ground truth, Figure 9, shows that 2,000 observations in a 9-dimensional space may only capture a coarse representation of the response surfaces, especially if any system responses exhibit non-linearity. In scenarios where the number of simulations for synthetic data generation is limited, it is crucial to validate the model's accuracy by examining cross-sections of the multidimensional space. This involves visually overlaying the predicted response surfaces onto the ground truth, as shown in Figure 10.

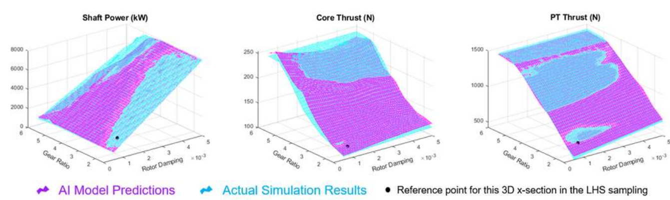

Figure 10. 3-D Response surfaces highlighting the great fit between simulated and predicted responses. Only one point (black dot) from this 3-D cross section of the 9-D parametric space has been used to train the AI models.

In this visual representation, a black dot indicates an individual observation used for training the machine learning models. The magenta-colored surface represents the ML-predicted response surface, while the blue surface depicts the re-generated or re-simulated ground truth. This 3D cross-sectional view of the 9D parametric space illustrates the impressive accuracy of the regression model, despite being trained with limited data. The overlay demonstrates how effectively the model captures the underlying response dynamics, validating its robustness in approximating complex systems. To facilitate this validation, we conducted additional simulations of the plant model to generate accurate 3D representations of response surfaces for specific cross-sections of the 9D design space. These simulations were supplementary to the original 2,000 observations and were not used for re-training the models. However, there is no restriction on using this data for future model refinement.

Given their distinct dynamics, the study also addressed the need for different models for transient versus steady-state operations. We evaluated the trade-offs between training separate models for each response and using a single deep-learning model for multiple responses. We selected separate models for each system response to achieve better accuracy. This approach highlights the importance of tailored model selection and validation in capturing engineering system complexity.

2.3. Parameter Optimization

The parameter optimization step can be interchangeably used with design space exploration, as highlighted earlier in the workflow. This is a crucial step in refining complex engineering designs. This research employed a multi-objective solver, specifically a Pareto-search algorithm, to identify optimal values for nine design variables that satisfy all three system objectives. Multi-objective optimization inherently involves trade-offs between different objectives, and the solver often provides multiple solutions. In our case, the optimization routine utilizing AI models for predicting system responses yielded 70 potential solutions, as depicted in the scatter plot in Figure 11. Traditionally, engineers might employ surrogate optimization by directly integrating physics-based simulation models to capture system responses. However, this approach can incur significant computational costs, especially when the system involves many degrees of freedom.

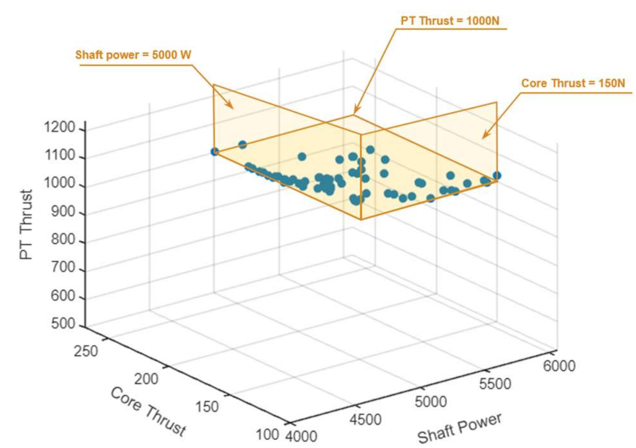

Figure 11. Optimization solver outcomes mapped against a 3-D objectives plot. The objectives are constrained such that the optimization solver looks for solutions in the highlighted cuboid.

The scatter plot reveals a range of solutions, with a cuboidal region representing the area of interest for all three design objectives. This region highlights the lower bounds of the objectives, with uncapped upper bounds. These solutions can be further filtered based on a bias towards specific system responses and can be transformed into a parametric configuration file for reintegration into the design cycle.

While machine learning regression models in this approach exhibit a 1-2% error, they offer significant efficiency advantages, particularly when identifying operational regimes rather than exact solutions. This optimization approach using machine learning models on synthetic datasets is highly efficient regarding computational time. However, challenges arise in cross-disciplinary settings, where one engineering team's design decisions impact another. In such scenarios, a full family of potential design solutions may help over optimization results. An interchangeable workflow between optimization and design space mapping is recommended, in such cases.

2.4. Design Space Mapping

Human beings can typically visualize in up to three dimensions, which is why design space mapping is valuable, especially when coupled with sensitivity analysis. This method involves identifying the top features in the dataset that most significantly influence system responses. For example, we focus on the two most sensitive parameters while fixing all other design parameters to create a 2D cross-section of the design volume. Although one can extend this visualization to 3D by including the top three features, we describe the 2D design space for simplicity.

Using regression models, we predict the three system responses and overlay them as contour lines on the 2D plot. In Figure 12, red contour lines represent shaft mechanical power, blue lines represent power turbine thrust, and black lines represent core thrust. The intersection of these contour lines reveals the full family of feasible design solutions or the design space. Within this space, designers and engineers can make conservative choices by selecting design points slightly inside the decision boundaries, fully aware of the marginal inaccuracy of the regression models.

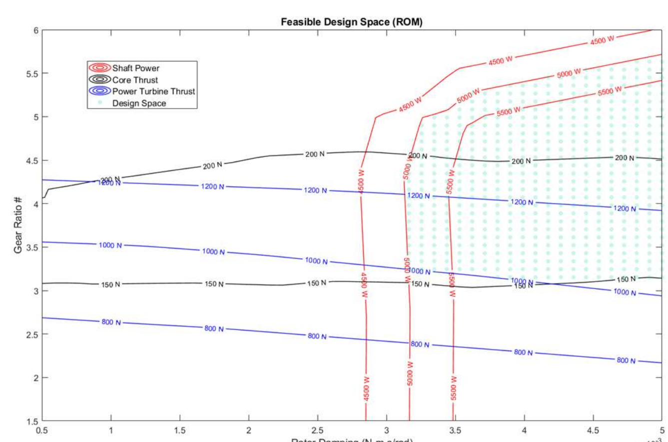

Figure 12. 2-D cross-section of a 9-D parametric space revealing the feasible design space for the chosen features of interest.

Objectives contours are overlayed to unveil the full family of feasible design solutions (design space), which is shown as dotted regions in green. As we vary any of the lower-ranked features, this 2D design space representation will adjust accordingly, either shrinking or expanding, as demonstrated in Figure 13 where the power turbine nozzle is opened up to 52% as compared to 42% in the previous result.

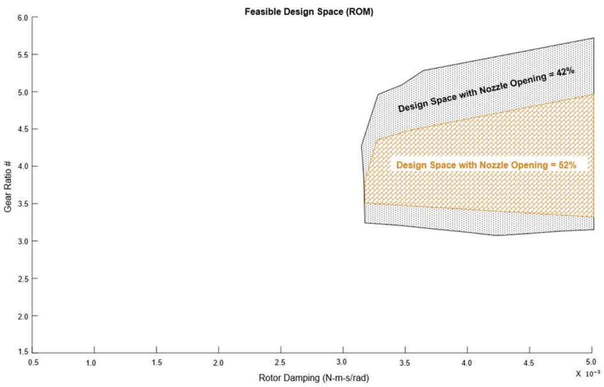

Figure 13. Design Space changes dynamically in the 2D cross-sectional view of the high dimensional parametric space as we change the other important features.

Importantly, this process does not require any additional computational resources once the machine learning models are trained, making it an efficient tool for exploring design possibilities.

3. CONCLUSION

This analysis marks a significant advancement in optimizing multi-objective systems by integrating data-driven insights and advanced optimization techniques. We began our research by strategically conducting trade studies on physics-based models, meticulously preparing data that captures the system's complexities. This groundwork enabled us to perform feature ranking, identifying key parameters that drive system responses, and enhancing the explainability of our AI models.

The development of a machine learning-based AI surrogate was pivotal, allowing designers to achieve top-notch accuracy in results without the need for further physical simulations. The use of AutoML was crucial in selecting and fine-tuning the most effective models, ensuring our surrogate model was both robust and precise.

Our AI-based approach to optimization, combining machine learning, demonstrated significant efficiency gains over traditional physics-based optimization methods. This not only accelerated the process multi-folds but also made it more adaptable, removing the time-consuming barriers of repeated optimization cycles.

Design space mapping provided a transformative way to visualize the high-dimensional parametric space, focusing on the most sensitive regimes within the design volume. This technique offered clarity in interpreting optimization results and revealed the full spectrum of feasible design solutions.

In essence, this version of our methodology sets a new benchmark for efficient, insightful design processes in complex engineering systems. By seamlessly integrating strategic data preparation, feature ranking, AI surrogates, multi-objective optimization, and advanced visualization techniques, we provide a comprehensive framework that enhances both the efficiency and depth of engineering design and decision-making.

4. FUTURE SCOPE

The current research focuses on steady-state design objectives. In the future, this work can be expanded to address transient-state design. To ensure predictive accuracy in transient-state applications, the physics-based model would require calibration with a trained AI model. Additionally, residual analysis can be included to get deeper understanding of prediction errors across the data range.

This research could also be extended to analyze the prediction speed and the memory footprint of the AI model on various deployment platforms, such as embedded hardware (virtual sensors) or cloud environments. Selecting a suitable AI model is critical to achieving optimal performance.

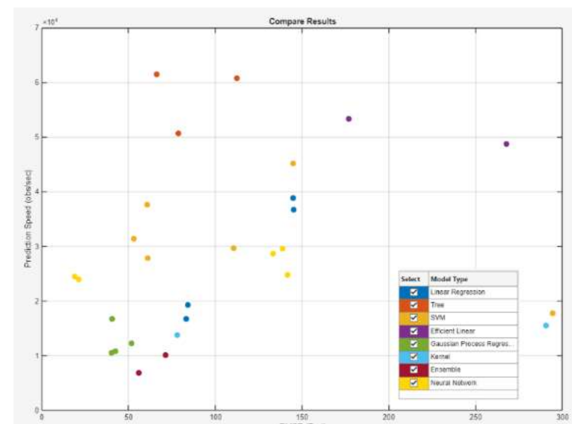

Figure 14. Performance vs. Prediction latency of various AI models.

For example, the Decision Tree models are faster, whereas the Neural Network models offer moderate speed with lower RMSE. Model comparison for prediction speed vs. accuracy is shown in Figure 14. Designers can use this information to perform trade-off analyses and choose the appropriate model based on specific deployment needs.

REFERENCES

1. Jasper Sneok, Hugo larochelle, and Ryan P.Adams, "Practical Bayesian Optimization of Machine Learning Algorithms", Advances in Neural Information Processing Systems 25, NIPS 2012.
2. Daniele Peri, "Machine Learning Algorithms in Design Optimization", https://arxiv.org/abs/2203.11005

3. M. Inoue, H. Matsumoto and H. Takagi, "Acceptability of a Decision Maker to Handle Multi-objective Optimization on Design Space," 2020 Joint 11th International Conference on Soft Computing and Intelligent Systems and 21st International Symposium on Advanced Intelligent Systems (SCIS-ISIS), Hachijo Island, Japan, 2020, pp. 1-6, doi: 10.1109/SCISISIS50064.2020.9322679.
4. https://www.mathworks.com/help/releases/R2024b/simscape/ug/brayton-cycle-gas-turbine-with-custom-components.html
5. https://www.mathworks.com/help/gads/paretosearch-algorithm.html
6. https://www.mathworks.com/products/reduced-order-modeling.html

DEFINITIONS/ABBREVIATIONS

DOE	Design of Experiments
ANN	Artificial Neural Networks
WNN	Wide Neural Network
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
AutoML	Automated Machine Learning
LHS	Latin Hypercube Sampling

BIOGRAPHIES

Mr. Peeyush Pankaj Peeyush Pankaj is a Principal Application Engineer at MathWorks, where he collaborates with engineering teams across the Automotive, Aerospace, and Industrial Automation (IAM) sectors to develop and deploy AI-powered solutions for complex system challenges. With over 14 years of experience, he enables organizations to integrate AI, machine learning, simulation, and data analytics into their workflows—driving innovation in areas such as, but not limited to predictive maintenance, visual inspection, big data and fleet analytics, and enterprise-scale intelligence.

Prior to MathWorks, Peeyush built a strong foundation in the aviation industry, working on the design, testing, and certification of aircraft engines. He holds 25 patents in jet propulsion technologies and prognostic health monitoring, reflecting a career grounded in technical excellence and practical impact. Peeyush holds a master's degree in advanced mechanical engineering from the University of Sussex, UK, and is deeply engaged with emerging trends in industrial AI and digital transformation. His ability to bridge domain expertise with scalable AI practices makes him a trusted advisor to engineering organizations navigating the shift toward smarter, data-driven systems.

Satish Thokala is Aerospace and Defense Industry manager at MathWorks®. His area of expertise is Avionics systems design for both military and civil aircrafts. In the current role, he is responsible to analyze technology landscape of Aerospace industry and develop strategies to increase the adoption of MATLAB® and Simulink® for mission and safety critical systems. Before joining MathWorks, he worked at premier aerospace organizations with a total experience of >22 years. Early in the career, Satish contributed to the design and development of communication radios and field trials of the same on military aircrafts. He led large engineering groups that developed software for cockpit displays, engine controls, AR/VR solutions and participated in the DO-178 certification audits.

Satish has delivered talks at 30+ international events, including the G-31 committee in 2022, where he spoke about Digital Engineering. In 2023, he presented Digital Thread and Design at the PLMSS workshop and participated as a panelist on integrated aircraft health management at the IVHM conference. Satish reviewed 10+ technical papers and published 2 papers. Satish co-authored a book on "Integrated Aircraft Health Management for Beginners", published by Aeronautical Society of India.