Harnessing Multi-fidelity Design, Analysis, and Optimization Techniques to Advance Fusion Reactor Design

By Greta Hibbard, Jacob Schwartz

I. ABSTRACT

The design of fusion reactors necessitates a comprehensive assessment of interacting components to ensure engineering and economic feasibility. Complex systems, like fusion power plants, are often preliminarily modeled using low - fidelity 'systems codes' for efficient design space exploration. This paper investigates applying aerospace-derived multi-fidelity techniques to incorporate higher-fidelity models while expediting system convergence and optimizing plant design. Multi-fidelity analysis utilizes high-fidelity models to locally calibrate low-fidelity models. We automate the integration of high-fidelity and low-fidelity models using Gaussian process regression, a Bayesian machine learning method. Low-fidelity power law models are 'trained' using high-fidelity models. Gaussian Process Regression is used to estimate the error of the low-fidelity model as design points vary from the initial training data set. If the error is large for a given design point, the point is added to the training set for the high-fidelity model, and the system is re-evaluated. It was found that Gaussian Process Regression as used in this algorithm offers greater advantages at higher fractional uncertainties (148 high-fidelity model evaluations saved after 300 total points requested when u = 0.3) while being less advantageous at lower fractional uncertainties (5 high-fidelity model evaluations saved after 300 total points requested when u = 0.1). It was additionally observed that constricting input parameters to smaller value ranges resulted in more saved high – fidelity model evaluations compared to previous tests (155 high-fidelity model evaluations saved after 300 total points requested when u = 0.3).

II. INTRODUCTION

The design of fusion reactors necessitates a detailed analysis of many interacting components. This paper evaluates how applying multi-fidelity modeling to fusion reactor design may advance and optimize the design process. It additionally aims to automate the leveraging of different fidelity levels used in multi-fidelity design analysis. This is done using Bayesian machine learning method Gaussian Process Regression. Fusion rates produced by theoretical data are used as the specific parameter to be calculated in this paper. This technique can be applied to various design challenges within the fusion community.

Modeling complex engineering system, such as a fusion reactor, as high-fidelity models require high computational run time to ensure high model accuracy. Small changes in the design may affect several other interacting components, resulting in a bias toward less design innovation; this discourages a complete, broad exploration of the design space. Conversely, low-fidelity modeling takes a more prototypical perspective. Low-fidelity models capture how different physical, engineering, and economic parameters may behave from a simplified perspective, but lack levels of high precision. An alternative design technique – multi-fidelity design analysis – has successfully modeled similarly complex engineering systems in the aerospace industry [Sobieszczanski-Sobieski]. Multi-fidelity modeling aims to combine high-fidelity and low-fidelity models to minimize run time while maximizing model accuracy.

III. THEORY

a. Low-Fidelity, High-Fidelity, and Multi-Fidelity Modeling

This project utilizes a method of design analysis known as multi-fidelity modeling. Multi-fidelity modeling is used in the aerospace industry to model complex engineering systems with many interacting components. The goal of multi-fidelity modeling is to combine the high model accuracy of high-fidelity models with the low computational run time of low-fidelity models. This work seeks to automate this process to take advantage of the benefits of each model as efficiently as possible.

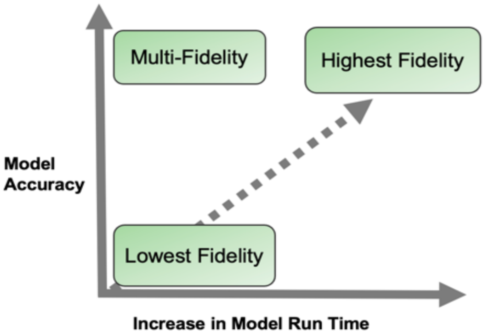

Figure 1 Multi-Fidelity Modeling

Low-fidelity models are cheaper, simplified models of an engineering system. These models typically occur early in the design process to model and predict how a system may behave at the most superficial level. They permit a swift exploration of the design space because they are easy to change and computationally quick to run. Where low-fidelity models excel in speed, they lack accuracy. High-fidelity models ensure high model accuracy but are computationally expensive. High-fidelity models are developed later in the design process to model the actual system in the highest level of precision achievable.

Multi-fidelity combines these two approaches to include the computationally inexpensive benefits of low-fidelity modeling and preserve the accuracy that high-fidelity modeling has to offer. This project utilizes Gaussian Process Regression, to automate leveraging the low and high-fidelity models. It passes between these systems to find where to allow the less computationally expensive job to do a “good enough” job and when the high-fidelity model is computationally worth running to ensure model accuracy.

b. Gaussian Process Regression

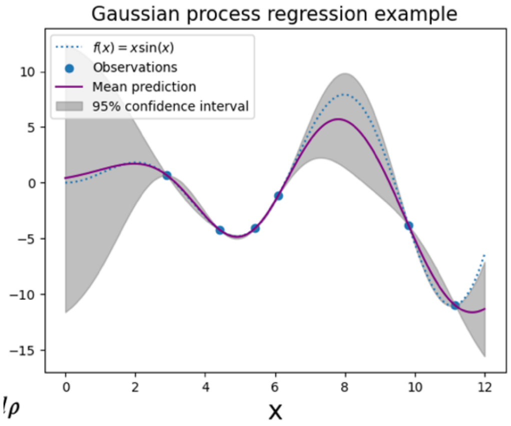

Figure 2 GPR example

Gaussian Process Regression (GPR) is a Bayesian approach to machine learning. A GPR model uses prior training knowledge (sets of known inputs and outputs) to make new output predictions for new inputs. The GPR model assumes there is some function f that describes the relationship between each input and output point. Two critical aspects of a GPR model are the mean function and covariance function. Together these functions make up the kernel of the GPR model. The kernel gives the model its predictive capabilities. The covariance matrix ensures points close together in the input space will be close together in the output space. This paper utilized a constant kernel multiplied by a radial basis function kernel.

(1.) Radial basis function kernel $k(\mathbf{x}_n, \mathbf{x}_m) = \exp(-||\mathbf{x}_n - \mathbf{x}_m||^2/2L^2)$,

(2.) Constant kernel $k(\mathbf{x}_n, \mathbf{x}_m) = \kappa$,

Gaussian Process Regression seeks to fit a function to a given set of observations to predict the function value for new inputs. There are an infinite number of functions that may fit these observation points, GPR seeks to evaluate the probability associated with each function and find the mean function from these possible functions, as illustrated in Figure 2.

(3.) Gaussian Probability Distribution Function $P_X(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

GPR was utilized in this project to automate the integration of low and high-fidelity models to create a multi-fidelity modeling system. Training points were calculated using the high-fidelity model. Each training point has an input ‘X’ vector and an associated output ‘Y’ vector. The GPR model makes outputs ‘Y’ test predictions for new input ‘X’ test points from these training points.

The GPR model evaluates the standard deviation from the purple mean prediction line (Fig. 2) over those predictions. This machine learning method is beneficial to the scope of this project because of its capability to make predictions based on a few (in the scope of this paper, 7 training points for over 200 test points) input parameters. This allows us to train the model with fewer runs of the more computationally expensive model instead of methods that require more training data which is beneficial to the scope of the project because it saves high-fidelity model iterations. As more test points are added, the GPR model has a more defined survey of the design space, allowing it to strengthen its predictive capabilities.

c. Fusion Rate

Multi-fidelity design analysis applies to many specific research areas in plasma physics and physics research generally. This paper tests the feasibility of applying this process to a simple problem. To apply multi-fidelity modeling, there must be both a high and low-fidelity model of the system. This work used a calculation of the total rate of fusion reactions (FR) in a toroidal plasma as a toy problem for testing this method.

The high-fidelity model (FR_high) involves integration over the plasma volume and calculating the local fusion rate for each differential volume. This formula provides a more accurate representation of the fusion rate in the plasma but is more computationally expensive than the low-fidelity model.

−1

(4.) $$\text{High-Fidelity Fusion Rate} = \int_0^1 \frac{dV}{d\rho}(\rho)\,\langle\sigma v\rangle\,(T)\,n^2\,d\rho$$

FR_high is built from density, temperature, and volume radial-based profiles of the plasma. The radial based profiles are generated in the form:

(5.) $$x(\rho) \doteq x_0(1-\rho^2)^{\alpha_x}$$

Variables alpha_n and alpha_T are created for density and temperature. These variables are the exponential term for each radial based profile. They control the profile shape. Alpha_n and alpha_T make up part of the input vector for the GPR model. The final element of the input vector is T_0, the central temperature. The local fusion rate is calculated for each volume element as

(6.) $$FR_{sec} = \langle\sigma V\rangle(T)^{*}\,n^2$$

Sigma V is the fusion rate coefficient, which is dependent on the temperature of each volume element. These sectional fusion rates are integrated over the entire volume of the plasma to give the total amount of fusion reactions.

The low-fidelity model (FR_low) is calculated based on volume-averaged density and volume-averaged temperature. This provides a quick, approximation of the fusion rate of a plasma volume.

(7.) $$\text{Low-Fidelity Fusion Rate} = V\,c\,\langle n^2 \rangle \langle T \rangle^{\beta}$$

Volume averaged density is calculated as

(8.) $$\langle n \rangle = \frac{n_1 V_1 + n_2 V_2}{V_1 + V_2}$$

Volume averaged temperature is calculated as

(9.) $$\langle T \rangle = \frac{n_1 T_1 V_1 + n_2 T_2 V_2}{n_1 V_1 + n_2 V_2}$$

IV. Workflow

The integration of high and low-fidelity models was achieved using an algorithm described in figure 3. The model initially requests that the user input various parameters (as detailed in the fusion rate section of this paper) specific to the fusion machine for the fusion rate to be evaluated. These parameters are alpha_n, alpha_T (exponential terms of radial based profiles), central temperature (T_0), central density (n_0), the plasma's minor radius (a), the plasma's major radius (R), and the level of fractional uncertainty the user is willing to accept. The machine takes in these inputs and generates data for the Gaussian Process Regressor model to be trained on. We will refer to these as "training points."

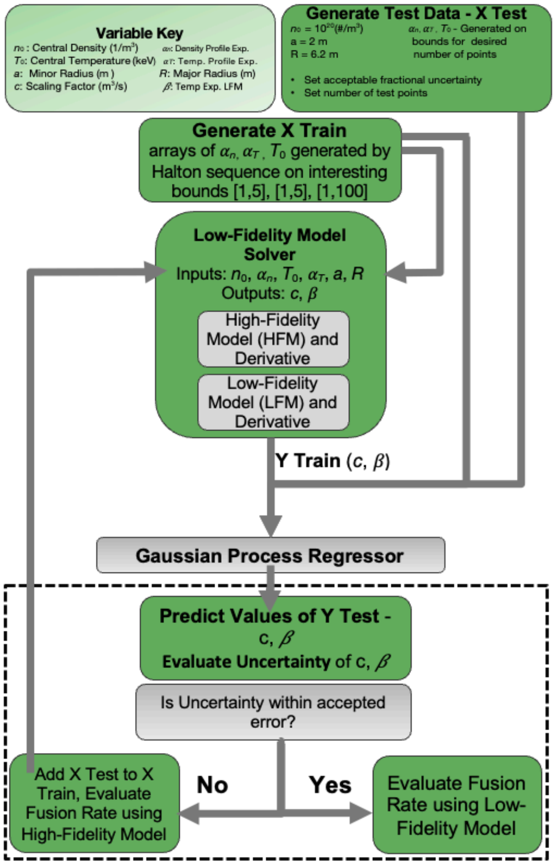
Variable Key
n_0 : Central Density (1/m^3)
α_n: Density Profile Exp.
T_0: Central Temperature (keV)
α_T: Temp. Profile Exp.
a: Minor Radius (m)
R: Major Radius (m)
c: Scaling Factor (m^3/s)
β: Temp Exp. LFM
Generate Test Data - X Test
n_0 = 10^{20}(#/m^3)
a = 2 m
R = 6.2 m
α_n, α_T , T_0 - Generated on bounds for desired number of points
• Set acceptable fractional uncertainty
• Set number of test points
Generate X Train
arrays of α_n, α_T , T_0 generated by Halton sequence on interesting bounds [1,5], [1,5], [1,100]
Low-Fidelity Model Solver
Inputs: n_0, α_n, T_0, α_T, a, R
Outputs: c, β
High-Fidelity Model (HFM) and Derivative
Low-Fidelity Model (LFM) and Derivative
Y Train (c, β)
Gaussian Process Regressor
Predict Values of Y Test - c, β
Evaluate Uncertainty of c, β
Is Uncertainty within accepted error?
No
Yes
Add X Test to X Train, Evaluate Fusion Rate using High-Fidelity Model
Evaluate Fusion Rate using Low-Fidelity Model

Figure 3 Algorithm Work Flow

The training points have an X vector component and a Y vector component. The X component of each point consists of an alpha n, alpha T, and central temperature value. These values are generated using a Halton sequence over an interesting range for each parameter. A Halton Sequence is a method of random point generation commonly used in quasi-Monte Carlo integration and optimization [Rus]. This method of point generation was selected instead of random generation because Halton Sequences provide more uniform distribution than random sequences. The interesting ranges for alpha_n, alpha_T, and T_0 are determined by bounds it can be expected that these values to be within. Alpha_n and alpha_t are both exponential terms, so they are constrained from 1 to 5. The central temperature is allowed to range from 1 to 100 keV. The other parameters, central density, minor radius, and major radius are taken to be constant, user specified values throughout the process.

The algorithm utilizes the high-fidelity model to calculate the associated Y outputs for the X input training points. This is done by numerically calculating the high-fidelity fusion rate based on the equation (Eq. 1) and its derivative. We set these values equal to the equations for the low-fidelity fusion rate (Eq. 4) and its derivative. Derivatives are taken for both levels of fidelity with respect to central temperature. Derivatives are calculated to set up a problem with two equations and two unknowns to solve for these unknowns with a nonlinear solver. The algorithm calls a nonlinear solver to calculate the values for c and Beta. These c and Beta parameters together are considered the train Y output for the training input points. Note that the high-fidelity model is evaluated for each training point.

The same initial input parameters are used to generate test data, X (alpha n, alpha T, central temperature) points for which we want to use the GPR model to estimate Y values (c and Beta). Values for alpha_n, alpha_T and T_0 were randomly generated on the same ranges as the training data. The Gaussian Process Regressor produces predicted Y outputs and an associated standard deviation for each point as detailed in the GPR section of this paper.

The machine compares the user-specified uncertainty to the standard deviation at each point. If the GPR error is acceptable, the low-fidelity model calculates the fusion rate based on the predicted c and Beta values. This is considered a saved evaluation of the high-fidelity model, the user-declared accuracy is preserved while saving model run time. If the produced error is greater than the acceptable error, the machine knows that to preserve accuracy, and it must use the high-fidelity model. Because the high-fidelity model has been run, it is beneficial to the GPR model to add these test points to the training set. This allows the machine to expand its knowledge of the design space with each evaluation of the high-fidelity model.

V. Results

The computational speedup using this algorithm was assessed under various allowed uncertainty levels and conclusions were drawn based on the efficiency. The first test varied the allowed levels of uncertainty as an input parameter and compares how many times the high-fidelity and low-fidelity model were evaluated for a given number of test points. This test shows the tendency to require more evaluations of the high-fidelity model for lower uncertainties. As lower levels of uncertainty are allowed, there is a lower chance for a "hit" (a previously calculated training point within required uncertainty) within the acceptance region to occur in the GPR model. In figure 4 the black line represents the number of high-fidelity iterations, and the blue line represents the number of low-fidelity iterations as a function of the allowed fractional uncertainty. It is observed that as allowable uncertainty decreases, the number of high-fidelity model evaluations increases. As expected, the algorithm saves more high-fidelity evaluations when the margin of error is higher. When the level of accepted uncertainty is high, such as 0.3 fractional uncertainty, the high-fidelity model is run half as often as it would without multi-fidelity design analysis. When the level of accepted uncertainty is low, such as 0.05

fractional uncertainty, it is observed that the high-fidelity model is iterated for nearly every test point.

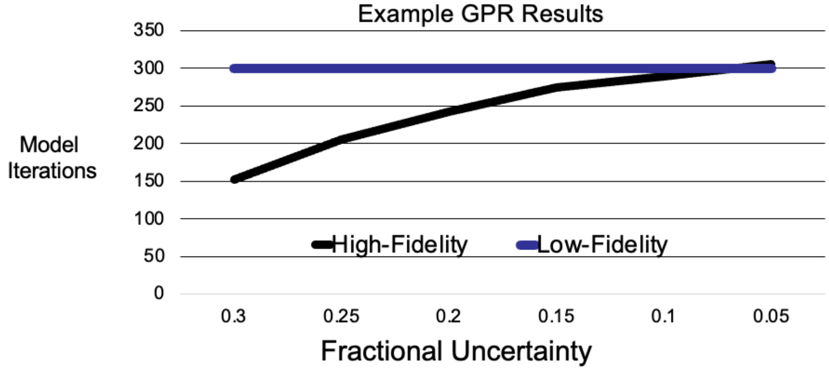

Figure 4 Model Iterations vs Allotted Fractional Uncertainty

The next test more closely tracks the training of the GPR model over time. The model was initially trained on 7 points evaluations of the high-fidelity model. Test points were added in batches of ten. Each time the points were added, the GPR model was ran. The amount of low-fidelity and high-fidelity models were recorded. This allows for the number of evaluations of high-fidelity model saved to be observed. This test is run at different allotted uncertainties to log how many "hits" on previously run models occur. An evaluation saved is an instance where the GPR model finds a close enough point in the training set that can make an appropriate prediction for the new point, allowing the low-fidelity model to be evaluated instead of the high-fidelity model while still maintaining the same level of accuracy. As the GPR model adds more points to the training data, there is a broader set of trained data that may be in the acceptable region of error. Gaussian Process Regression, as used in this algorithm, offers more significant advantages at higher fractional uncertainties (148 high-fidelity model evaluations saved after 300 total points requested when u = 0.3) while being less advantageous at lower fractional uncertainties (5 high-fidelity model evaluations saved after 300 total points requested when u = 0.1).

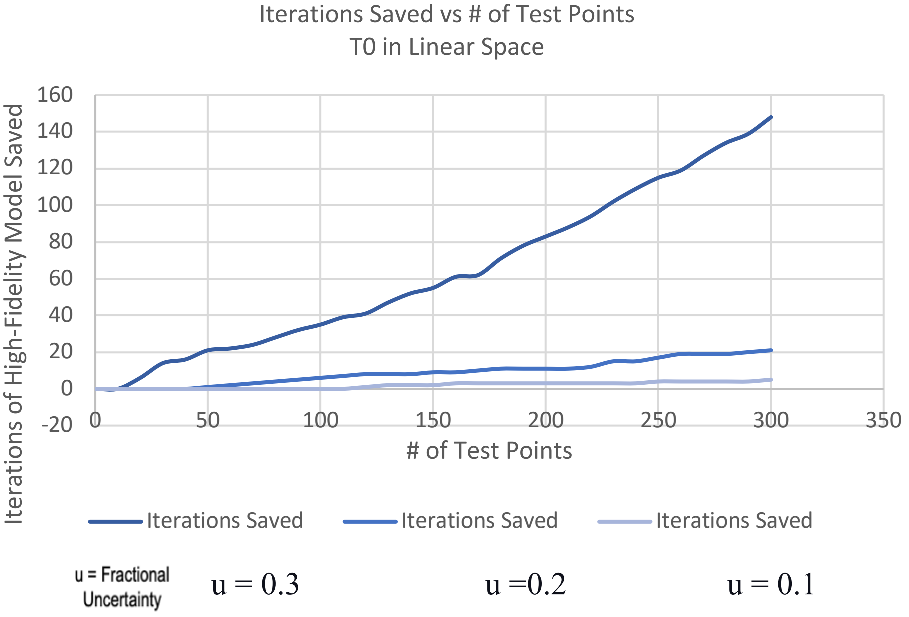

Figure 5 Iterations Saved vs # of Test Points for T0 in Linear Space

It is likely that the central temperature element of the input test data is the element that is most often causing high-fidelity model iterations. Comparing alpha_n, and alpha_T, both with ranges 1 to 5, to T_0, with range 1 to 100, the GPR model is less likely to find a "hit" within acceptable uncertainty to evaluate the low-fidelity model with. To test this, T_0 was taken into log space. The new range of T_0 in log space was 0.00 to 4.61. The second test was conducted again, but with T_0 in log space.

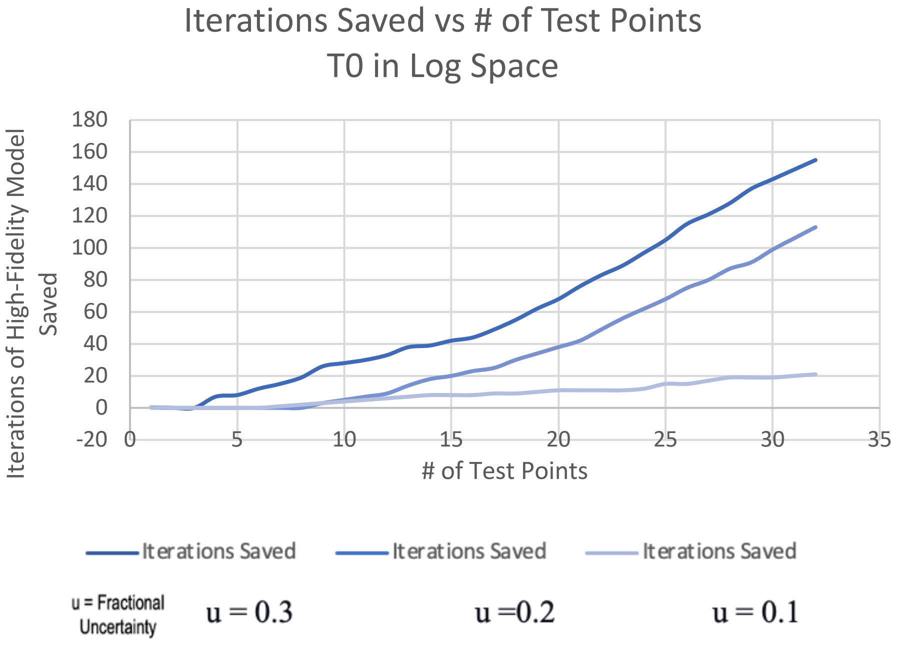

Figure 6 Iterations Saved vs # of Test Points for T0 in log Space

The contour map figures below are visual representations of the Gaussian Process Regressor Model. The "Y" output consists of two parameters, c , and Beta. Figure 6 shows the predicted value of log($c*10^{16}$ m^{-3}s). Figure 8 shows the predicted value of Beta. It is observed in both cases that the predicted values for both variables have a greater tendency to change with change in central temperature than alpha_T. Figures 7 and 9 show the standard deviation of log($c*10^{16}$ m^{-3}s) and Beta respectively. As shown in figure 2, it is characteristic of Gaussian Process Regression to have regions of less standard deviation around observed points. In these contour maps, this is illustrated by figures 7 and 9. We can locate each training point based on where the lowest regions of standard deviation are.

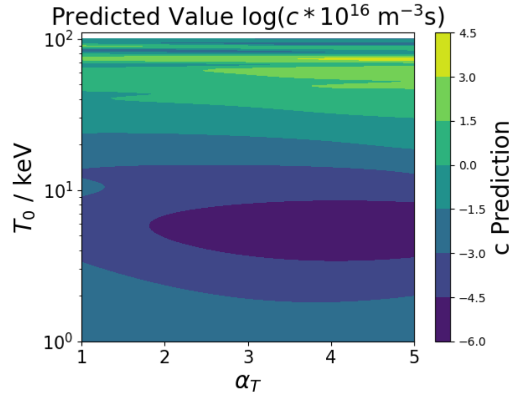

*Figure 6 Predicted Value log (c*10^16 m^-3s)*

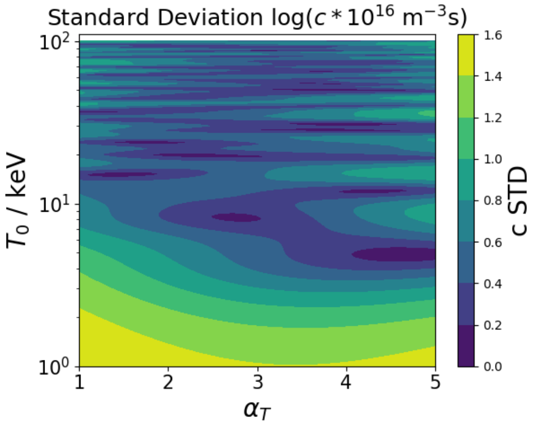

*Figure 7 Standard Deviation log (c*10^16 m^-3s)*

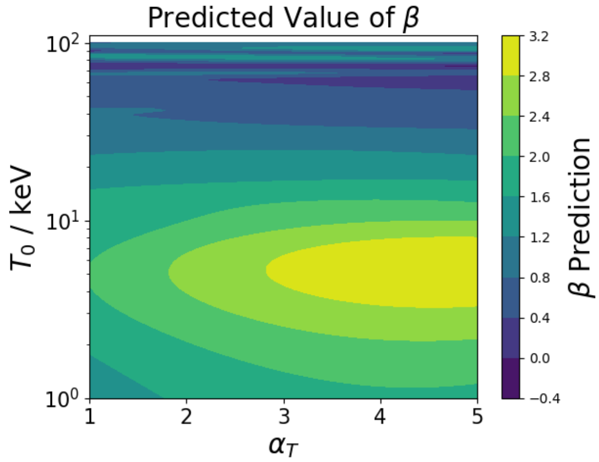

Figure 8 Predicted Value of Beta

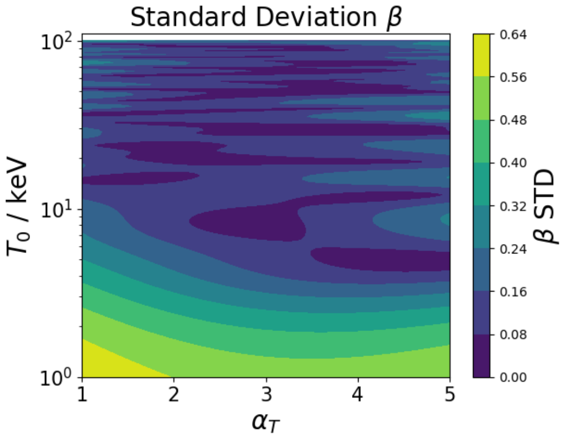

Figure 9 Standard Deviation of Beta

VI. Conclusions and Discussion for Future Work

Applying multi-fidelity modeling using Gaussian Process Regression is more beneficial than solely computing with high-fidelity modeling when a more considerable margin of error is acceptable. This method of design analysis proved more successful when the ranges of the input parameters were smaller, as observed in comparing the cases of T_0 in linear space versus T_0 in log space. In all cases, as numbers of test points increased, multi-fidelity modeling using Gaussian Process Regression did reduce the number of high-fidelity model evaluations and save computational time.

General efforts should be made to minimize computational expenses in the code. As the code exists currently, there is room for improvement in computational efficiency.

For future research, it would be interesting to apply this sort of modeling to a real-life fusion reactor design process. In a real-life application, there will be more certainty on realistic ranges of the parameters. Similarly to the case of T_0 in linear space versus T_0 in log space, we can expect the program to save more high-fidelity iterations and computational time for smaller ranges of input values.

Additionally, it is hypothesized that multi-fidelity modeling could prove successful with predictive methods beyond Gaussian Process Regression. Replacing the GPR model with a different Bayesian method and comparing computational speed up to the GPR model would be an interesting study.

As previously mentioned, the method of multi-fidelity modeling could be utilized in fields beyond fusion reactor design. Testing this method in different research applications is additionally an interesting path for future research.

Works Cited

About This Jupyter Book — Learning from Data. https://danielrphillips.github.io//LearningFromData/about.html. Accessed 21 Sept. 2023.

Bowcutt, Kevin. “A Perspective on the Future of Aerospace Vehicle Design.” *12th AIAA International Space Planes and Hypersonic Systems and Technologies*, American Institute of Aeronautics and Astronautics, 2003. *DOI.org (Crossref)*, https://doi.org/10.2514/6.2003-6957.

Coleman, M., and S. McIntosh. “BLUEPRINT: A Novel Approach to Fusion Reactor Design.” *Fusion Engineering and Design*, vol. 139, Feb. 2019, pp. 26–38. *DOI.org (Crossref)*, https://doi.org/10.1016/j.fusengdes.2018.12.036.

Kovari, M., et al. “‘PROCESS’: A Systems Code for Fusion Power Plants – Part 2: Engineering.” *Fusion Engineering and Design*, vol. 104, Mar. 2016, pp. 9–20. *DOI.org (Crossref)*, https://doi.org/10.1016/j.fusengdes.2016.01.007.

Lion, J., et al. “A General Stellarator Version of the Systems Code PROCESS.” *Nuclear Fusion*, vol. 61, no. 12, Dec. 2021, p. 126021. *DOI.org (Crossref)*, https://doi.org/10.1088/1741-4326/ac2dbf.

Martins, Joaquim R. R. A., and S. Andrew Ning. *Engineering Design Optimization*. Cambridge University Press, 2021.

Natsume, Y. “Gaussian Process Kernels.” *Medium*, 23 Aug. 2022, https://towardsdatascience.com/gaussian-process-kernels-96bafb4dd63e.

Rasmussen , C. E., & Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Rus, Jacob. “Halton Sequence.” *Observable*, 25 May 2018, https://observablehq.com/@jrus/halton.

Scikit-Learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Sobieszczanski-Sobieski, J., and R. T. Haftka. “Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments.” *Structural Optimization*, vol. 14, no. 1, Aug. 1997, pp. 1–23. *DOI.org (Crossref)*, https://doi.org/10.1007/BF01197554.