

Turbine Blades - Design of Computer Experiment

Yuxuan Chen, Xige Huang, Gaojia Xu, Guanqi Zeng

4/18/2022

1 Background and Introduction

In contemporary society, power is needed everywhere in our daily life. Although not familiar to most of us, to create power, the generator needs momentum from the engine that produces kinetic energy from some sources. One type of combustion engine, the gas turbine, is a crucial component of the power generating process across all industries, such as aircrafts, trains, and electricity plants. Connected to the generator, the gas turbine drives the generator to function by pressurized gas: the air is drawn to the turbine from its suction and heated by fuel source combustion; then, as the heated air expands, the motion is made by the turbine and is connected to the generator to produce electricity. Transforming pressure energy to kinetic energy, the turbine blades are the key element of a gas turbine. As it extracts energy from the high temperature and high pressure gas, it is also the limiting part of a gas turbine, as it needs to withstand the severe environment. To protect the turbine blades, an internal cooling system is incorporated in the turbine, extracting cooling air from the compressor and passing through the airfoils to cool the blades.

While blade deformation to some extent is acceptable, an extreme deformation can largely reduce the length of the engine life cycle, and the life of the blades themselves will decrease by half only by a 30 degree celsius off in the blade temperature prediction. As a result, the design of the turbine blades requires careful consideration on the choice of materials and the cooling schedule.

In this project, we aim to find an optimal design for the turbine blades to minimize blade stress and deformation. Since physical experiments are expensive to prototype and run, computer experiments are used instead to reduce costs. The deterministic black-box simulator makes use of a finite-element model (FEM) solver which models the physical deformation of the blade during operation. There are six inputs in the simulator, which will be elaborate on the next section. There are mainly two goals we need to achieve: firstly, we want to train an surrogate model on stress that yields good prediction performance over the design space that can be used for a variety of downstream tasks, including model calibration and system control; secondly, we want to identify good blade material properties and cooling conditions which minimize maximum blade stress over the turbine blade. More importantly, we also need to consider the fact that a maximum displacement greater than $d^* = 1.3 \times 10^3$ will result in immediate failure of the engine; therefore, we need to take the black-box constraint into account while improving emulation accuracy and optimizing the black-box simulator.

Another main challenge that need to be carefully taken care of in every stage of our design and computer experiments is that running the simulator is fairly time-consuming. Therefore, given limiting computational resources and also to make our algorithms and results reproducible, the project is limited to a maximum of 150 computer experiment runs.

2. Method

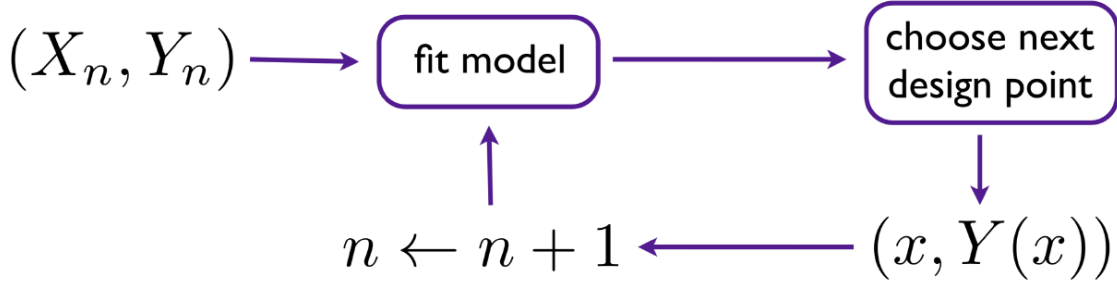
In this section, we will describe our problem method based on mathematical formulation. We are given a black-box simulator that takes six input, presented in table 1 with respective desired design range (given by the collaborator):

Table 1: Simulator input parameter and desired design range

variable	range
Young’s modulus	$[200 \times 10^9, 300 \times 10^9]$
Poisson’s ratio	$[0.1, 0.49]$
Coefficient of thermal expansion (CTE)	$[5 \times 10^{-6}, 1.5 \times 10^{-5}]$
Thermal conductivity	$[5, 15]$
Interval cooling air temperature	$[50, 350]$
Pressure load on suction	$[1 \times 10^5, 4.5 \times 10^5]$

For both of the goals, i.e., prediction accuracy improvement and optimization under constraint, we need accurate surrogate models (on both *stress* and *constraint*) because directly evaluating the black-box simulator is not computationally feasible. Because the black-box function is very likely to be highly complicated, Gaussian Process (GP), is a good choice to fit such complex response surface.

Moreover, we will conduct sequential designs for actively improvement prediction accuracy while searching for the optimal design, and also for optimization, instead of a one-shot design. Sequential selection has the advantages of being more practical, more computationally reasonable, and also being better behaved numerically and faster (SRG). The brief workflow of sequential design is presented in figure 1; we will elaborate more in the following sections.



All technical detailed of implementation can be found in our github repository.

2.2 Space-Filling Design

Because we known nothing about the response surface we are modeling, we need a good space-filling design that spread out points with the aim of encouraging a diversity of data for better estimation of the parameter. One problem is, model-based designs are highly relied on good specification of the hyperparameters, which is unknown and must be estimated from data observed at the chosen design sites (Zhang et al., 2020). This creates a chicken-or-egg problem because we hope to use data, which we’ve not yet observed since we’re still designing the experiment, to learn hyperparameter settings (SRG). To alleviate such paradox, we adopt the beta-distribution design (Zhang et al., 2020), which performs well in sequential settings, as our initial design. In short, the distribution of pairwise distances of a betadist distribution follows a beta distribution (and we use Beta(2,5)).

Ideally, we will need a testing grid to evaluate the model accuracy. However, we have limited number of runs and thus are not able to generate a dense grid to evaluate the model. As an alternative, we will generate a *maximin* design. Maximin design has the advantage of points being more spread out evenly in the hyperspace, though with one main problem of points being pushed more to the boundaries. We will use the maximin design for test set to approximate RMSE of the model on the response surface.

2.3 Active Learning Cohn (ALC)

Following from the workflow from figure 1, after obtaining the initial betadist design with size 15, we evaluate these points by the simulator function to obtain a complete training set $D_{n_0} = (X_{n_0}, \mathbf{y}_{n_0})$, where $\mathbf{y}_{n_0} = [\text{stress}, \text{displacement}]$. We first fit the initial surrogate Gaussian Process model on *stress* and *displacement*. Then, we solve the criterion function $J(x)$ but under the constraint $\mu_{disp}(x) < 1.3 \times 10^{-3}$ to derive the next point x_{n+1} . Here, we denote the posterior mean of the surrogate GP as $\hat{d}(x)$. We then update the GP for both *stress* and *displacement* with $D_{n+1} = D_n \cup (x_{n+1}, \mathbf{y}_{n+1})$ and repeat.

In the sequential design, there are mainly two criteria $J(x)$ to be considered: the Active Learning Mackay (ALM) and the Active Learning Cohn (ALC). The ALM criterion aims to find the next point to maximize predictive variance based on the current dataset. Specifically,

$$x_{n+1} = \operatorname{argmax}_{x \in \mathcal{X}} \sigma_n^2(x)$$

On the other hand, the ALC criterion is more aggregate.

$$x_{n+1} = \operatorname{argmin}_{x \in \mathcal{X}} \int_{x \in \mathcal{X}} \tilde{\sigma}_{n+1}^2(x) dx$$

where $\tilde{\sigma}_{n+1}^2(x)$ here is the deduced variance based on design X_{n+1} combining X_n and a new input location x_{n+1}^\top , rather than the predictive variance.

One thing to note on the constraint on displacement, since the surrogate GP model may not yield good predictive performance at the beginning, and as a consequence, the constraint based on the prediction of displacement may not be actually effective in constraining the solve of criterion into the design space we desired. To incorporate the predictive uncertainty into our constraint, we start with setting the constraint as $\delta(x) < 1.3 \times 10^{-3}$ where $\hat{\mu}_{disp} - n * \hat{\sigma}_{disp} < \delta(x) < \hat{\mu}_{disp} + n * \hat{\sigma}_{disp}$, which is equivalent to restricting a predictive interval rather than the predictive mean. We then gradually tighten the constraint during the sequential designs. In our implementation, we set $n = 1.96$ for the first 10 iterations, $n = 1.64$ for the next 10 iterations, and $n = 0$ for the rest.

2.4 Optimization

For the second goal of optimization, we want to find the global minimum of the black-box function, the simulator, with the help of GP model got from last part. The overall workflow is analogous to the algorithm of ALC except that the criterion $J(x)$ is based on Gaussian Process predictive mean $\mu_n(x) = E\{Y(x)|D_n\}$. There are several criterion used in optimization, we adopted *Expected Improvement*(EI) here. EI can be seen as an improvement on EY, where we wish to find $x^* = \operatorname{argmin}_{x \in \mathcal{X}} f(x)$. In contrary, EI is specified as

$$\text{EI}(x) = (f_{\min}^n - \mu_n(x)) \Phi\left(\frac{f_{\min}^n - \mu_n(x)}{\sigma_n(x)}\right) + \sigma_n(x) \phi\left(\frac{f_{\min}^n - \mu_n(x)}{\sigma_n(x)}\right)$$

Therefore, EI takes both predictive mean and uncertainty factor (i.e., probability of improvement) into account.

When taken the constrain on displacement into account, we extend EI to *Integrated Expected Conditional Improvement*

$$\text{IECI}(x_{n+1}) = - \int_{x \in \mathcal{X}} \mathbb{E}\{I(x | x_{n+1})\} \prod_{j=1}^m p_n(x) dx$$

where $p_n(x)$ stand in for the predicted probability, from fitted surrogate model classifier(s), that input x satisfies the constraint; $p_n(x)$ will downweights reference x -values not satisfying the constraint. Because we have a real-valued constraint that displacement has to be lower than 1.3×10^{-3} , $p_n(x)$ can be specified as

$$p_n(x) = \Phi\left(-\frac{\mu_n(x)}{\sigma_n(x)}\right)$$

where μ_n and σ_n is the predictive mean and variance from the surrogate GP for displacement.

3. Result

Among the total of 150 runs, we have allocated 10 runs of simulator to generate the initial design, 30 runs to generate the test set to validate model accuracy, 50 runs to perform ALC search to improve optimization, and 50 runs to perform EI for optimization.

From figure 2 we can see that RMSE continues to decrease, despite some small jumps during the progress, and reaches a minimum of 360136441 after the full 50 sequential runs. We have tested the final GP model for stress on all observations with displacement $< 1.3 \times 10^3$, the resulting RMSE is 378951104. One thing to notice is that after 50 sequential runs, the RMSE does not have a clear trend to converge. We believe that RMSE can continue to decrease if more sequential runs are performed.

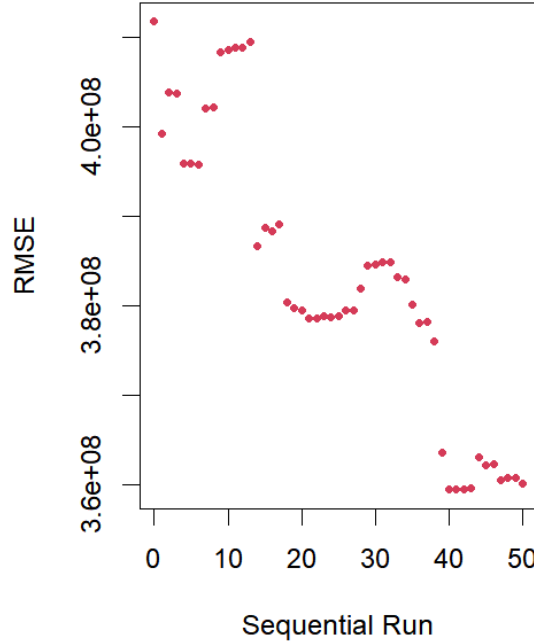


Figure 1: out-of-sample RMSE over 50 ALC (under constraint) acquisitions

Based on our results from the constraint ALC, we have found that the surrogate GP for displacement is actually not very reliable because most of the sequential point searched under the constraint does not actually satisfy the constraint. We conducted optimization both taking the constraint into account, i.e., optimizing over IEI, and not taking the constraint into account, i.e., optimizing over EI. We have found that optimizing over EI yields a smaller minimum value of stress, 371498114.19, with displacement of $7.0661e-04$ actually falls under the constraint; the associate six inputs values are 221221200540, 0.470, 5.783×10^6 , 10.567, 165.892, 433805.6 optimizing over ICEI yields result where both stress and displacement larger than those from optimizing over EI. This result may be due to the inaccuracy of prediction from our surrogate GP for displacement.

Figure 3 shows the trend of best observed value over the number of EI runs (without acutally deal with the constraint in the algorithm).

4. Conclusion and Discussion

After a total of 150 runs of simulator, we have conducted ALC to trained the final GP model on stress to yield a optimal RMSE of 378951104 on observations with displacement $< 1.3 \times 10^3$. We have also used

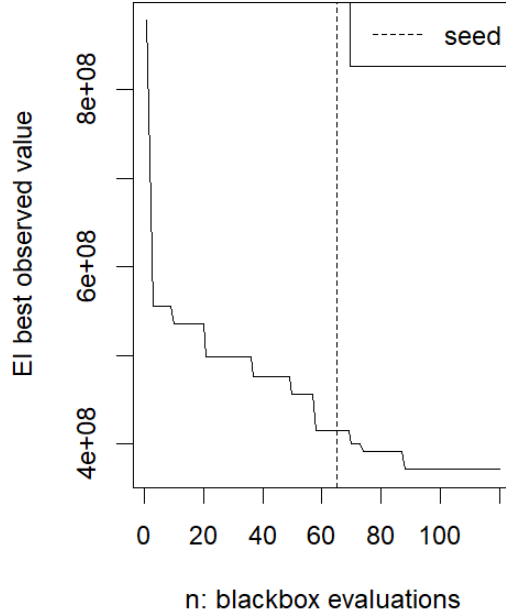


Figure 2: EI progress in terms of best observed value

surrogate GP models to minimize the black-box simulator to find the minimum maximum blade stress of 371498114.19, with displacement of $7.0661e-04$ actually falls under the constraint; the optimal setting for the six parameters is: Young’s modulus=221221200540, Poisson’s ratio = 0.470, Coefficient of thermal expansion (CTE)= 5.783×10^6 , Thermal conductivity=10.567, Internal cooling air temperature=165.892, Pressure load on suction=433805.6.

One main problem of our results is that the RMSE of our final model given the optimal design is still relatively high. We believe that this is mainly due to the fact that the GP we have fitted with the initial design does not yield good rmse. There is potential for improvement if hyperparameters are tuned more appropriately. Also, if there are more number of runs, we could fit the initial GP with initial designs generated with different space-filling method.

Moreover, the optimization algorithm yields better results when we explicitly ignored the constraint while conducting sequential optimization. This can also be attributed to the fact that our model on displacement even after the ALC runs to improve accuracy does not have satisfactory performance in prediction. Ignoring the constraint surely does not often yields results that happens to satisfy the constraint, although it happens in our case. This can be considered our risky attempt to circumvent the fact that the model accuracy cannot be improved drastically given limited computational resources and time.

5. Reference

1. Zhang, Y., Tao, S., Chen, W., and Apley, D. (2018). A latent variable approach to Gaussian process modeling with qualitative and quantitative factors. Preprint on arXiv:1806.07504.
2. Gramacy, R. B. (2020). Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences. Chapman and Hall/CRC.