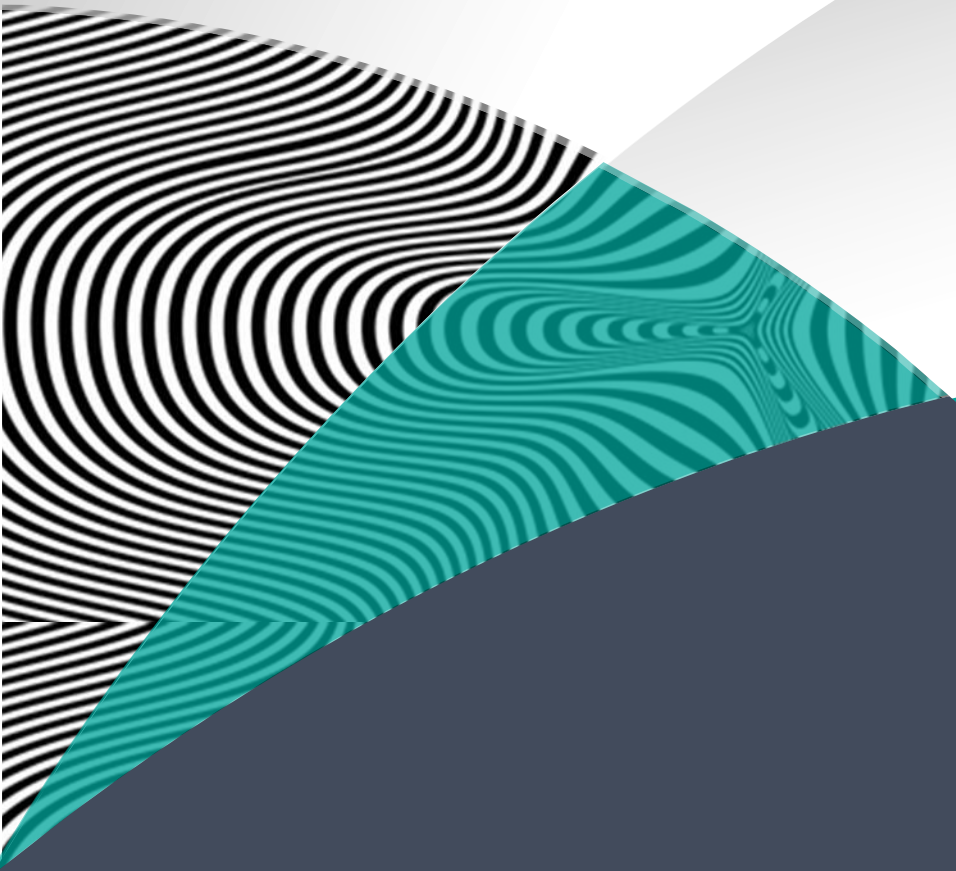
Linear and Non-linear Programming: MATH 404

Dr. Ahmed Abdelsamea

Eng. Mohssen Elshaar



2D Bayesian Optimization in Tuning PI Controller

University of Science and Technology, Zewail City, Fall 2022

Table of Contents

Abstract

When it comes to black-box functions, Bayesian optimization is a sequential design technique that does not make any functional form assumptions. Usually, it is used to improve difficult-to-evaluate functions. In this paper, 2D Bayesian optimization is used to tune the PI controller gain for the pitch angle in a UAV. The purpose is to minimize the settling time. Short- period mode approximation is used. Three different approaches are compared.

Introduction

Since the 1960s, the statistics and machine learning communities have refined a Bayesian approach to optimization. Bayesian optimization routines rely on a statistical model of the objective function, whose beliefs guide the algorithm in making the most fruitful decisions. These models can be quite sophisticated, and maintaining them throughout optimization may entail significant cost of its own. However, the reward for this effort is unparalleled sample efficiency. For this reason, Bayesian optimization has found a niche in optimizing objectives that:

- are costly to compute, precluding exhaustive evaluation,
- lack a useful expression, causing them to function as “black boxes,”
- cannot be evaluated exactly, but only through some indirect or noisy mechanism, and/or
- offer no efficient mechanism for estimating their gradient.

Closely related to optimal design of experiments, dating back to Kirstine Smith (1918). As Bayesian optimisation, studied first by Kushner (1964), then by Mockus (1978), and more recently by Jones, et al. (1998).

The Bayesian approach to optimization allows us to relax all of these assumptions when necessary, and Bayesian optimization algorithms can deliver impressive performance even when optimizing complex “black box” objectives under severely limited observation budgets. Bayesian optimization has proven successful in settings spanning science, engineering, and beyond, including of course hyperparameter tuning (turner, 2021).

On the contrary to classical mathematical optimization, the objective function is not required. Rather, only access to a mechanism revealing some information about the objective function at identified points on demand is needed (Garnett, 2023).

Problem definition

UAV

Unmanned aerial vehicles (UAVs) have become increasingly popular due to their many civilian and military applications. Extending the envelope of autonomous flight capability to extreme aggressive manoeuvring has been a growing research topic. In recent years, the continuous development in engineering-related fields, such as automatic systems, flight control and the aerospace industry as a whole, has contributed to the rapid growth of the area of Unmanned Aerial Vehicles (UAV), making it an appealing research topic in both military and civil applications. In terms of civil applications, it is important to mention those related with agricultural services, marine operations, natural disaster support, etc. Different types of controllers can be designed for UAVs. The simplest ones are linear PID based on linearized models of UAVs.

Due to their simplicity, ease of implementation and robust performance, a decentralised and linear control scheme based on four proportional-integral-derivative controllers (PID) has been chosen to design the attitude tracking controller. Controller for pitch angle is implemented using standard PID controller with feedback of estimated pitch angle from a complementary filter, resulting in the following control law:

$$u = K_p\left((\theta_d - \theta) + \frac{1}{T_i}\int(\theta_d - \theta)dt + T_d(\dot{\theta}_d - \dot{\theta})\right)$$

$$e(t) = \theta_d - \theta$$

where K_p , K_i ,*and* K_d are all non-negative, denote the coefficients for the proportional, integral, and derivative terms respectively (sometimes denoted P, I, and D). θ_d is associated with the reference angle or desired pitch, while u is the controller action: u is associated with the deflection angle of elevator.

In the standard form of the equation, K_i and K_d are respectively replaced by $\frac{K_p}{T_i}$ and K_pT_d; the advantage of this being that T_i and T_d have some understandable physical meaning, as they represent an integration time and a derivative time respectively.

- K_pT_d is the time constant with which the controller will attempt to approach the set point.
- $\frac{K_p}{T_i}$ determines how long the controller will tolerate the output being consistently above or below the set point. It is mostly set to zero and will be treated this way throughout this work.

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right)$$

There are many different manual methods for tuning a controller that involves observing the process response after inflicting controller setpoint changes. One method involves increasing the amount of setpoint change and repeating the procedure until the process enters a state of steady-state oscillation. The PID controller learns how the process responds to a change in setpoint, and suggested PID settings. (Mortenson, 2022)

In this paper, we aim to tune the PI controller gains for UAV- VTOL pitch angle in short-period approximation using Bayesian optimization.

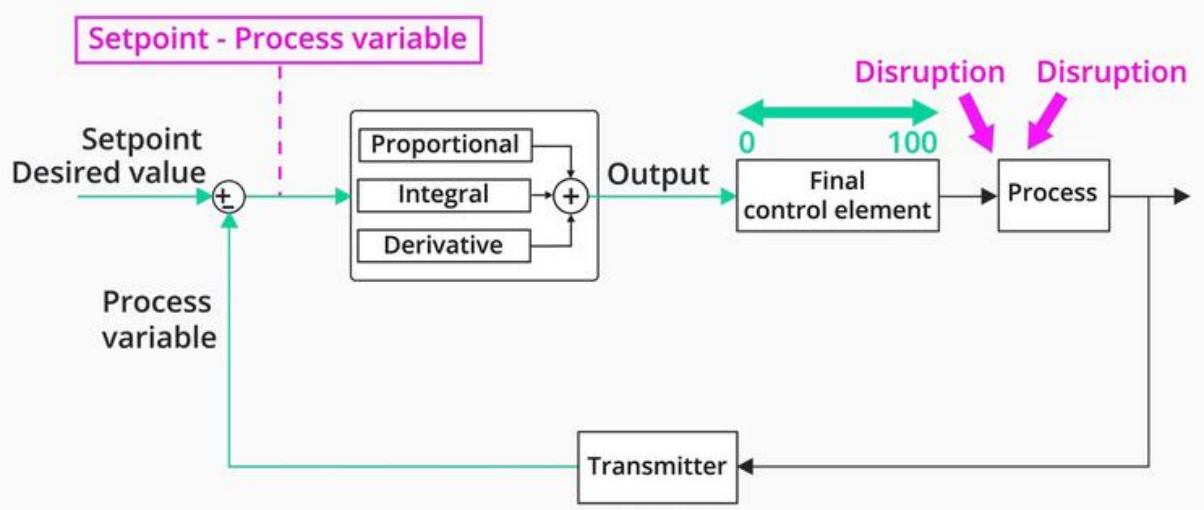


Figure 1 Schematic of a system with a PID controller implemented

Bayesian optimization

Algorithm

1. Choose some prior measure over the space of possible objectives f.
2. Combine prior and the likelihood to get a posterior measure over the objective given some observations.
3. Use the posterior to decide where to take the next evaluation according to some acquisition/loss function.
4. Augment the data. (Gonzalez, 2017)

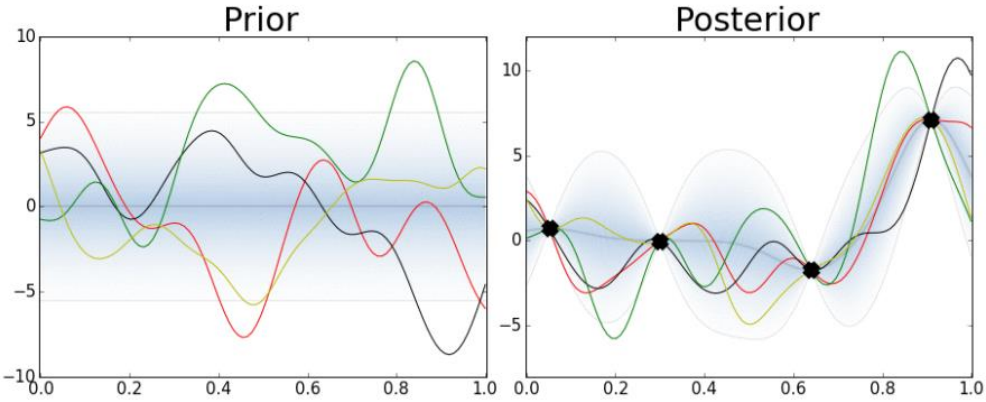


Figure 2 Sample problem explaining the Bayesian algorithem adopted from Gonzalez, 2017

Begin with an initial (possibly empty) dataset D that can grow incrementally through a sequence of observations of our design. In each iteration, an optimization policy inspects the available data and selects a point $x \in X$ where next observation is made. This action in turn reveals a corresponding value y provided by the system under study. Append the newly observed information to our dataset and finally decide whether to continue with another observation or terminate and return the current data. When finished, the returned data can be used by an external consumer as desired, for example to inform a subsequent decision (Garnett, 2023)

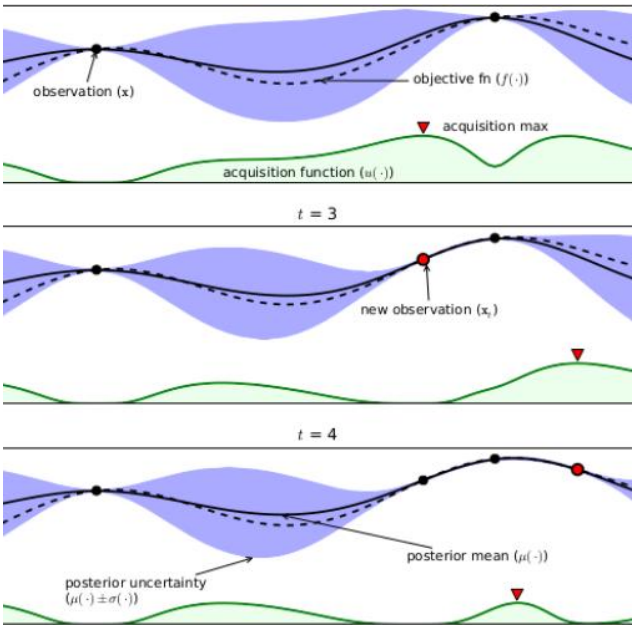


Figure 3 dashed like: real objective function. solid black line: posterior mean μ(x). green line: acquisition function. blue area: confidence area. Adapted from Jin 2018

Assume observations $(\boldsymbol{x_{1:t}}, \boldsymbol{f_{1:t}})$ and it follows the multivariate normal distribution

$$\mathcal{N}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$$

Now, a new point $\boldsymbol{x_{t+1}}$, by properties of GP, $\boldsymbol{f_{1:t}}$ and f_{t+1} are jointly Gaussian:

$$\begin{pmatrix} \mathbf{f}_{1:t} \\ \mathrm{f}_{t+1} \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{pmatrix} \boldsymbol{K} & \boldsymbol{k} \\ \boldsymbol{k}^T & k(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1}) \end{pmatrix}\right)$$

$$\boldsymbol{k} = \left(k(\boldsymbol{x}_{t+1}, \boldsymbol{x}_1), \cdots, k(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t)\right)$$

Then the predictive distribution:

$$p(f_{t+1}|D_{1:t}, \boldsymbol{x}_{t+1}) = \mathcal{N}\left(\mu_t(x_{t+1}), \sigma_t^2(\boldsymbol{x}_{t+1})\right)$$

$$\mu_t\left(\boldsymbol{x}_{(t+1)}\right) = \boldsymbol{k}^T \boldsymbol{K}^{-1} \boldsymbol{f}_{1:t}$$

$$\sigma_t^2(\boldsymbol{x}_{t+1}) = k(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1}) - \boldsymbol{k}^T \boldsymbol{K}^{-1} \boldsymbol{k}$$

Initialization

To initialize the problem, many approaches can be taken:

1. One point in the centre of the domain.
2. Uniformly selected random locations.
3. Latin design.
4. Halton sequences.
5. Determinantal point processes.

The idea is always to start at some locations trying to minimise the initial model uncertainty.

Gaussian processes

The Gaussian processes model is a probabilistic supervised machine learning framework that has been widely used for regression and classification tasks. A Gaussian processes regression (GPR) model can make predictions incorporating prior knowledge (kernels) and provide uncertainty measures over predictions (Rasmussen, 2006). Gaussian processes model is a supervised learning method developed by computer science and statistics communities.

In regression, for a given set of observed data points, there are infinite numbers of possible functions that fit these data points. The traditional nonlinear regression methods typically give one function that is considered to fit the dataset best. However, there may be more than one function that fit the observed data points equally well. When the dimension of MVN are infinite, predictions could be

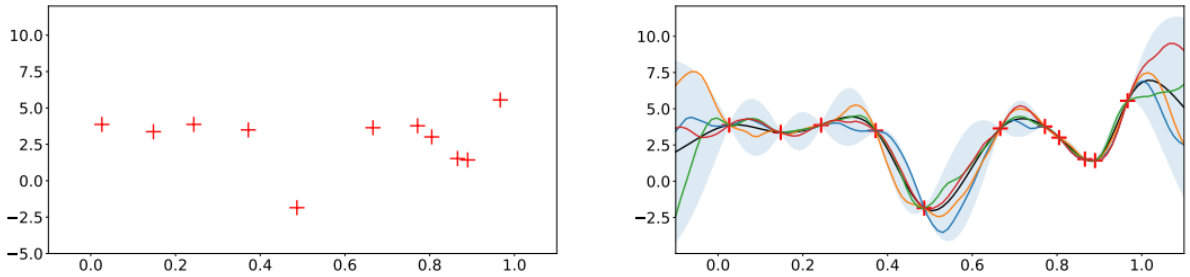


Figure 4 A regression example: (a) The observed data points, (b) Five sample functions

made at any point with these infinite numbers of functions. These functions 9 are MVN because it's our assumption (prior). More formally speaking, the prior distribution of these infinite functions is MVN. The prior distribution represents the expected outputs of f over inputs x without observing any data. When observations are obtained, instead of infinite numbers of functions, only the functions that fit the observed data points are kept.

Gaussian Distribution

A Gaussian processes model describes a probability distribution over possible functions that fit a set of points. Hence functions means can be calculated, as well as the variances to indicate how confident the predictions are. The key points are summarized as

1) the function (posteriors) updates with new observations,
2) a Gaussian process model is a probability distribution over possible functions, and any finite samples of functions are jointly Gaussian distributed,
3) the mean function calculated by the posterior distribution of possible functions is the function used for regression predictions.

A random variable X is Gaussian or normally distributed with mean μ and variance σ 2 if its probability density function (PDF) is

$$P_X(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Here, X represent random variables and x is the real argument. The normal distribution of X is usually represented by $P_X(x) \sim \mathcal{N}(\mu, \sigma^2)$.

Kernels

In regression, outputs of the function should be similar when two inputs are close to each other. One possible equation format is the dot product $A \cdot B = ||A||\,||B||\,cos\theta$, where θ is the angle between two input vectors. When two input vectors are similar, their dot product output value is high. If a function is defined solely in terms of inner products in the input space, then the function $k(x, x')$ is a kernel function (Rasmussen, 2006).

The most widely used kernel or covariance function is the squared exponential (SE) kernel function. The SE kernel is de-facto default kernel for Gaussian processes (Duvenaud, 2014). This is because it can be integrated against most functions that you need to due to its universal property. And every function in its prior has infinitely many derivatives. It's also known as the radial basis function (RBF) kernel.

- Squared exponential: $k_{SE(r)} = exp\left(-\frac{r^2}{2\ell^2}\right)$
- Matern class: $k_{Matern}(r) = \frac{2^{1-\mu}}{\Gamma(\nu)} * \left(\frac{\sqrt{2\nu r}}{\ell}\right)^{\nu} * K_{\nu}\left(\frac{\sqrt{2\nu r}}{\ell}\right)$
- Exponential: $k_{Exp(r)} = exp\left(-\frac{r}{\ell}\right)$
- γ-Exponential: $k_{\gamma-Exp(r)} = exp\left(\left(-\frac{r}{\ell}\right)^{\gamma}\right), for\ 0 < \gamma \leq 2$
- Neural Networks: $k_{NN}(x, x') = \frac{2}{\pi} sin^{-1}\left\{\frac{2x^T \Sigma x'}{\sqrt{1+2x^T \Sigma x} * \sqrt{1+2x'^T \Sigma x'}}\right\}$
- Periodic: $k_{periodic}(x, x') = exp\left\{-\frac{2sin^2\left(\frac{1}{2}(x-x')\right)}{\ell^2}\right\}$

A stationary covariance function is a function of $(x - x')$. A covariance function is called isotropic if it is a function only of $||x - x'||$ and is also called radial basis functions (RBFs).

$$r = ||x - x'||$$

A covariance function is called dot product covariance function if it is a function depends only on x and x' through $x \cdot x'$

A more general name of the covariance function of taking two inputs $x \in \mathcal{X}, x' \in \mathcal{X}$ into $\mathbb{R}$ is called kernel. A covariance matrix $\boldsymbol{K}$ is a Gram matrix of pairwise covariance functions of a given points $\{x_1, x_2, \cdots, x_n\}$, where $\boldsymbol{K}_{ij} = k(x_i, x_j)$.

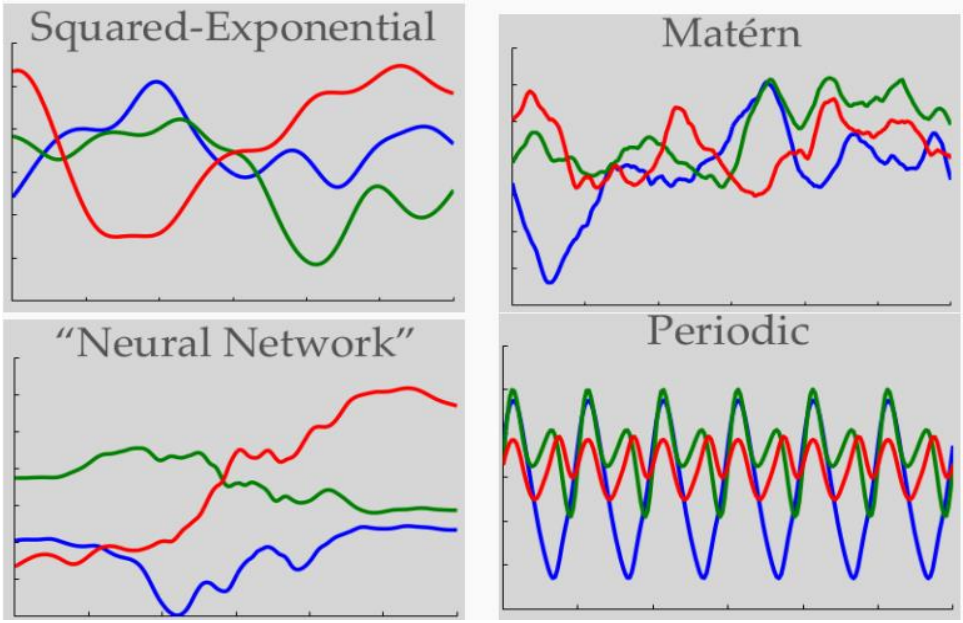


Figure 5 Example of covariance functions adapted from Jin 2018

Covariance function	**Expression**	**Stationary**	**Non degenerate**
Constant	σ_0^2	Y	
Linear	$\sum_{d=1}^{D} \sigma_d^2 \, x_d \, x_d^{'}$		
Polynomial	$\left(x \cdot x^{'} + \sigma_0^2\right)^p$		
Squared exponential	$exp\left(-\frac{r^2}{2\ell^2}\right)$	Y	Y
Matern	$\frac{2^{1-\mu}}{\Gamma(\nu)} * \left(\frac{\sqrt{2\nu r}}{\ell}\right)^{\nu} * K_{\nu}\left(\frac{\sqrt{2\nu r}}{\ell}\right)$	Y	Y
Exponential	$exp\left(-\frac{r}{\ell}\right)$	Y	Y
γ-exponential	$exp\left(\left(-\frac{r}{\ell}\right)^{\gamma}\right), for\ 0 < \gamma \leq 2$	Y	Y
Neural networks	$\frac{2}{\pi} sin^{-1}\left\{\frac{2x^T \Sigma x^{'}}{\sqrt{1+2x^T \Sigma x} * \sqrt{1+2x^{'T} \Sigma x^{'}}}\right\}$		Y

Here, dimension refers to the number of multi-variables of MVN. When the dimension of MVN gets larger, the region of interest will be filled up with more points. When the dimension becomes

infinity, there will be a point to represent any possible input point. All functions are randomly generated by the MVN model incorporating kernel functions as prior knowledge before having any observed data points.

Acquisition function

Acquisition function balances between exploration and exploitation.

Exploration

points with high variance. *Exploration* seeks to sample in locations where the uncertainty is high. This ensures that no major region of the space is left unexplored — the global minima may happen to lie there. (towards)

Exploitation

points with high mean (max problem). *Exploitation* seeks to sample where the surrogate model predicts a good objective. This is taking advantage of known promising spots. However, continually exploiting known information will yield little gain. (towards)

An acquisition function that encourages too much exploitation and too little exploration will lead to the model to reside only a minima it finds first (usually local — 'going only where there is light'). An acquisition function that encourages the opposite will not stay in a minima, local or global, in the first place. Yielding good results in a delicate balance.

Types

Probability improvement

$$a_{PI}(\boldsymbol{x}) = P\big(f(x) \geq f(x^+)\big) = \phi\left(\frac{\mu(\boldsymbol{x}) - f_n^*}{\sigma(\boldsymbol{x})}\right)$$

is pure exploitation and is sometimes modified as:

$$a_{PI}(\boldsymbol{x}) = P(f(x) \geq f(x^+) + \xi) = \phi\left(\frac{\mu(\boldsymbol{x}) - f_n^* + \xi}{\sigma(\boldsymbol{x})}\right)$$

Expected improvement

$$a_{EI}(\boldsymbol{x}) = \mathbb{E}[[f(\boldsymbol{x}) - f_n^*]^+]$$

Where

$$[v]^+ \triangleq \max\{0, v\}$$

And

$$f_n^* \triangleq \min_{x\,1:n} f(x) \text{ is the current best}$$

$$\begin{aligned} a_{EI}(\boldsymbol{x}) &= (\mu(\boldsymbol{x}) - f_n^*)\phi(\mathrm{Z}) + \sigma(x)\phi(\mathrm{Z}) && \sigma(x) > 0 \\ &= 0 && \sigma(x) = 0 \end{aligned}$$

$$Z = \frac{\mu(\boldsymbol{x}) - f_n^*}{\sigma(\boldsymbol{x})}$$

ϕ(Z) is the probability density function and Φ(Z) is the cumulative distribution

Upper confidence bound (UCB)

$$a_{UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \beta\sigma(\boldsymbol{x})$$

Where $\beta > 0$ is a tradeoff parameter

The goal is to find

$$Min \sum_{t}^{T} f^*(\boldsymbol{x_t}) - f(\boldsymbol{x_t}) = Max \sum_{t}^{T} f(\boldsymbol{x_t})$$

Lower confidence bound (LCB)

$$a_{LCB}(\boldsymbol{x}) = -\mu(\boldsymbol{x}) + \beta\sigma(\boldsymbol{x})$$

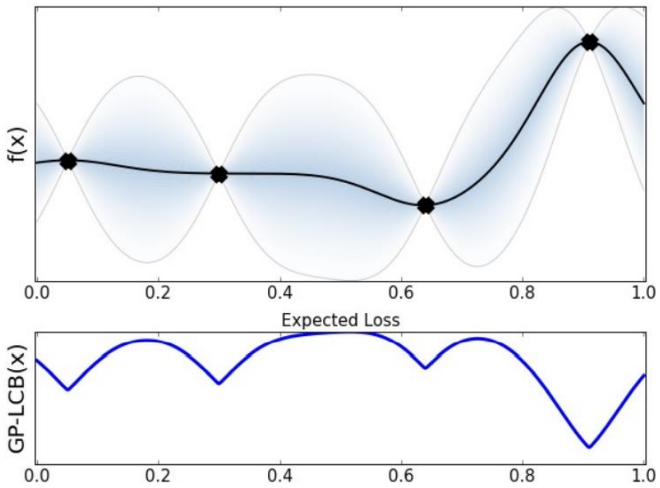


Figure 6 Lower confidence bound adopted from Gonzalez,2017

Entropy search/ Predicted entropy search

$$a_{ES}(\boldsymbol{x}) = H\big(P(\boldsymbol{x}^*)\big) - \mathbb{E}_{f(x)}\left[H\left(P\big(\boldsymbol{x}^*\big|f(\boldsymbol{x})\big)\right)\right]$$

$$a_{PES}(\boldsymbol{x}) = H\big(P(f(\boldsymbol{x}))\big) - \mathbb{E}_{x^*}\big[H\big(P(f(\boldsymbol{x})|\boldsymbol{x}^*)\big)\big]$$

Hyperparameters Optimization

In practice, GPR models are more complex. Kernel functions play significant roles in GPR. The choice of kernel functions determines almost all the generalization properties of a GP model (Duvenaud, n.d).

Optimizing Acquisition Function Methods

1. Gradient descent methods: Conjugate gradient, BFGS, etc.

2. Lipschitz based heuristics: DIRECT.
3. Evolutionary algorithms: CMA. (Gonzalez, 2017)

Methods

The airplane dynamics is divided into 2 modes: Longitudinal and lateral. These modes can serve as a good appreciation to the system dynamics under small disturbances. The Longitudinal mode approximation is used to avoid the nonlinearity in the airplane system: avoiding solver errors, slow solution, complicated analysis, and complicated PID controller system. The longitudinal mode includes the linear velocity in x direction u, the linear velocity in the z direction w, the angular velocity in the xz-plane q, and finally the pitch angle θ. The longitudinal mode is divided into 2 responses: the long-period response, and the-short period response. The short-term response includes u and θ and it is the one modeled. More information about the airplane in use in appendix.

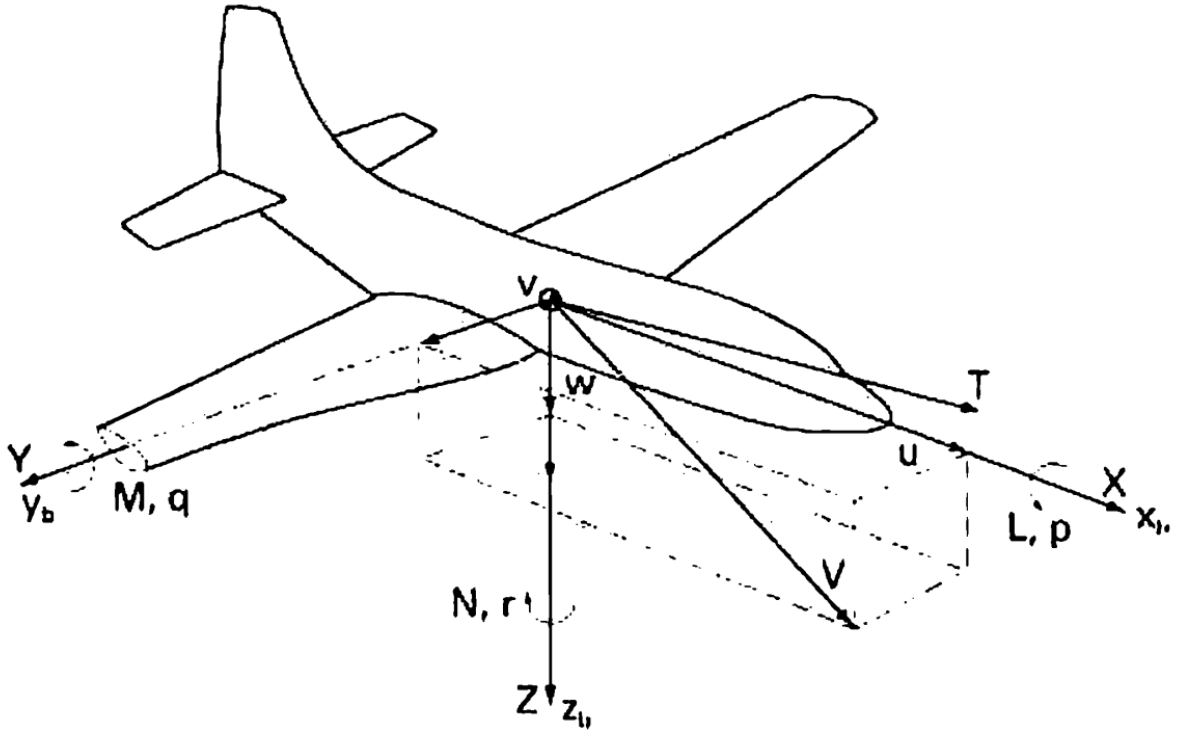


Figure 7 Schematic of the airplane body axis and their corresponding state variables.

By modeling the pitch angle response to a step input, the settling time is 39.0234 seconds.

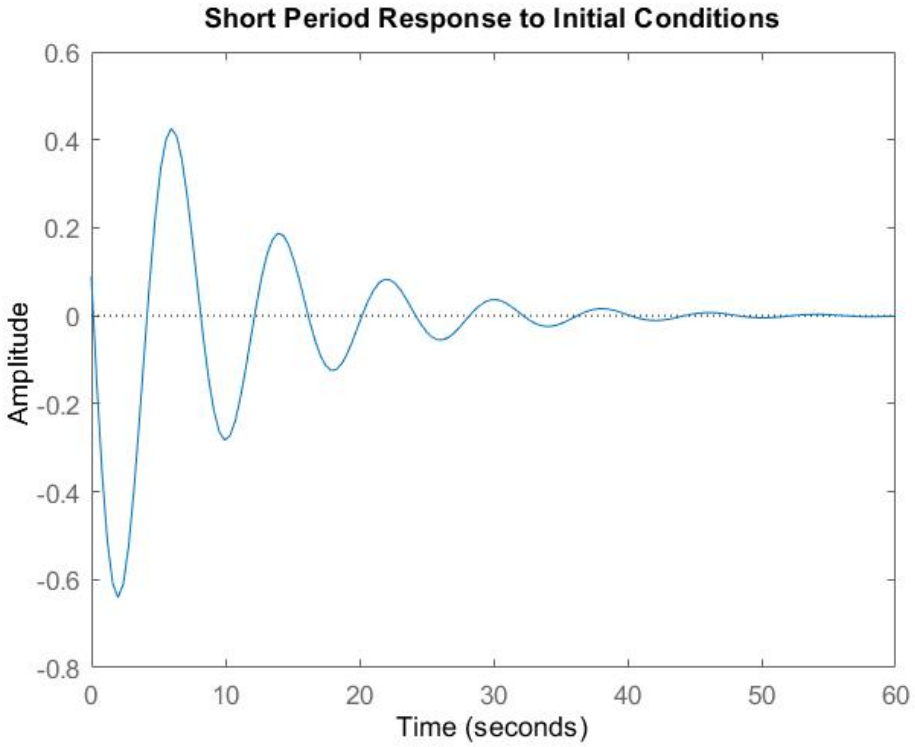


Figure 8 Short period response of the pitch angle θ

A PI controller was implemented on this system and its response was obtained. By changing the K_p and K_i values, different settling time values were calculated.

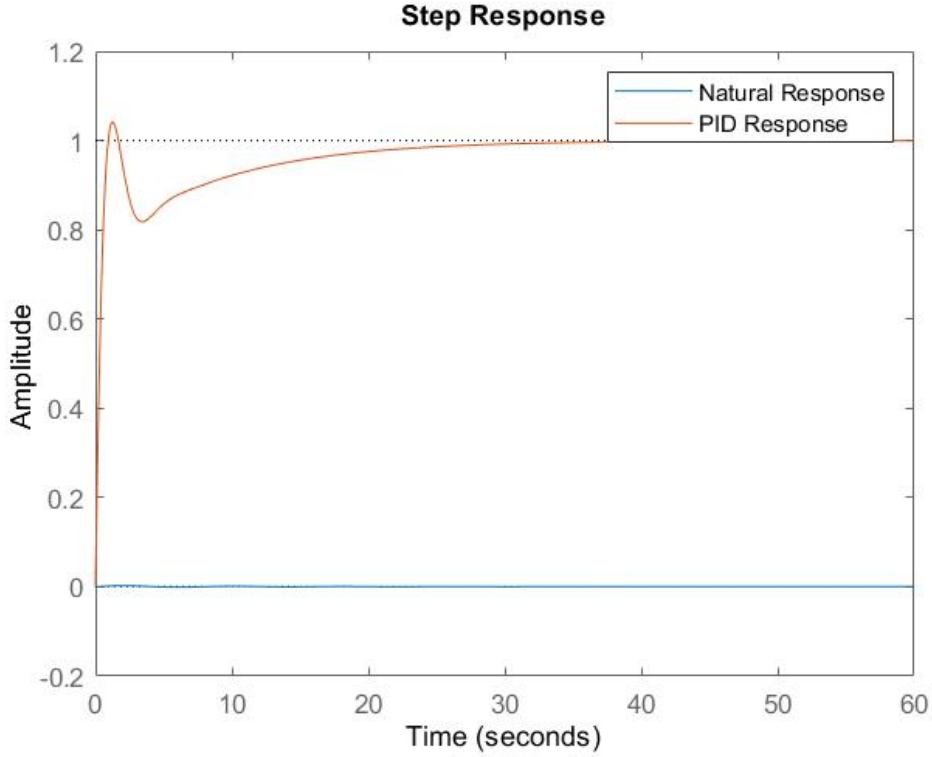


Figure 9 Sample PID response for Kp=1000, Ki=1000

One hundred responses were obtained by varying the values of both gains from 100 to 1000 by a step of 100. Then these values were manually excluded from MATLAB and sorted into an excel sheet. This data is used later in different optimizing algorithms.

First, a sample Gaussian processes were generated in the 100-1000 domain for visualization.

Second, the data was fitted using different kernel function by **fitgrip** function and visualized, and a minimum was obtained by **min** function.

Third, the data was run through MATLAB's **BayesOpt** function.

Fourth, the data was run through a 2D Bayesian optimization software by Pilario, 2023.

Results

Sample Gaussian Processes

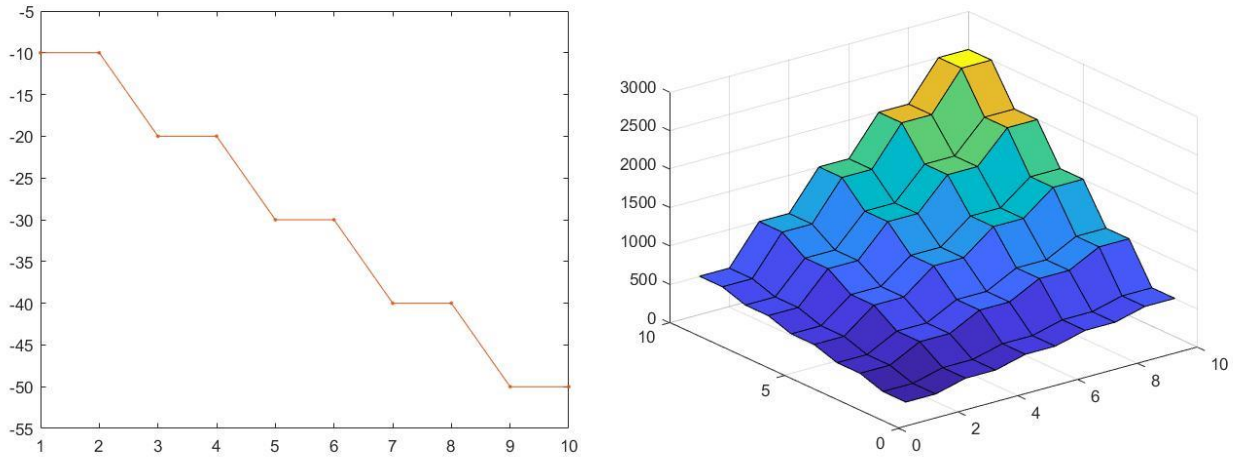
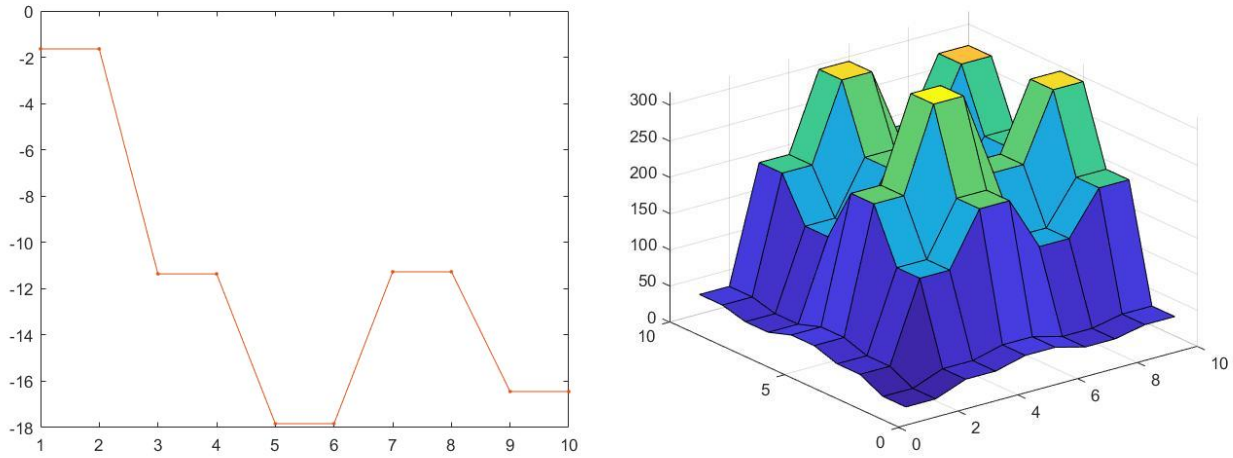


Figure 12 Linear Kernel



Fi

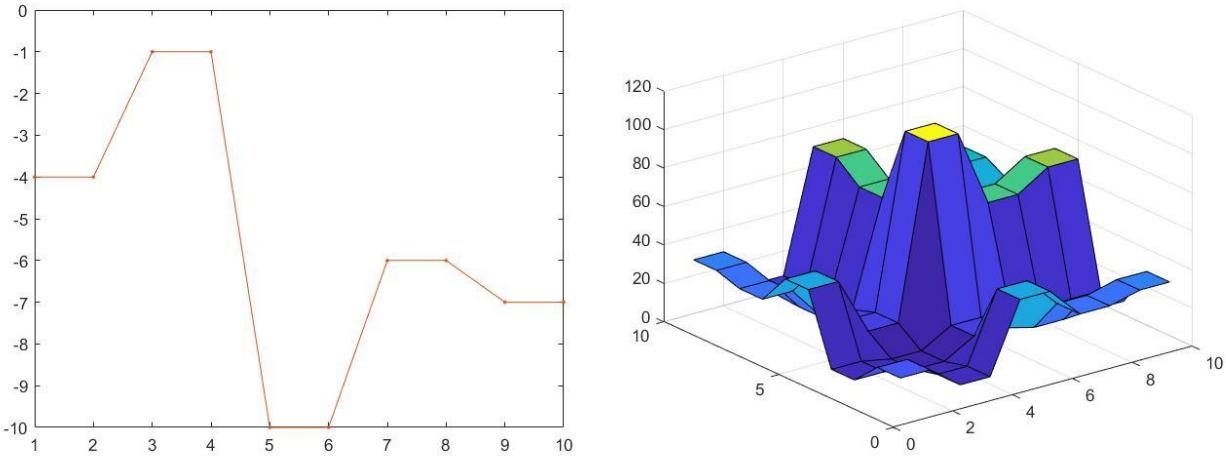


Figure 10 Squared Exponential Kernel

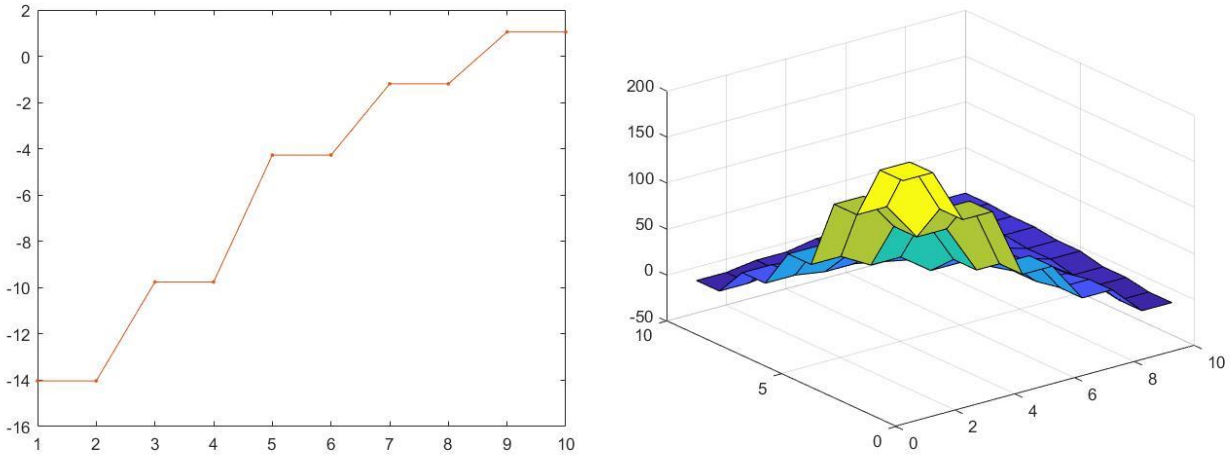


Figure 14 Periodic Kernel

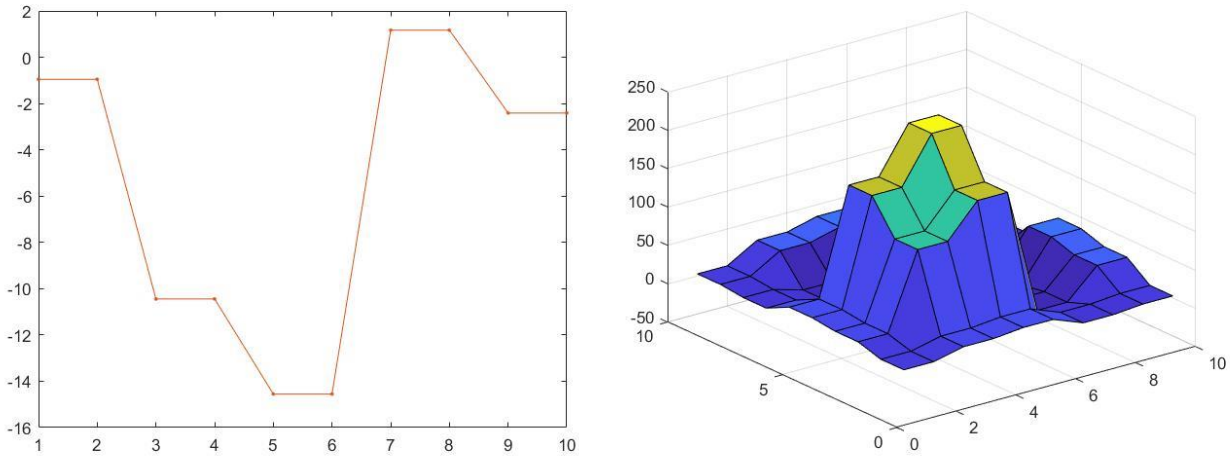


Figure 13 Exponential Kernel

These sample processes are random and changed every time. The axis values are multiplied by 100 to represent the actual range of values used.

Data Fitting

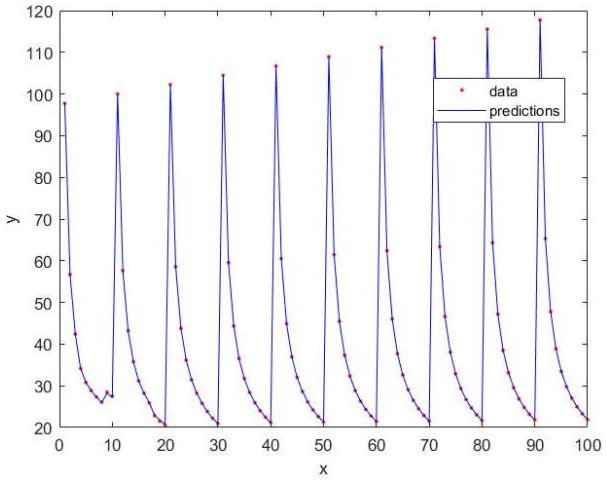


Figure 19 ArdExponential Kernel Function

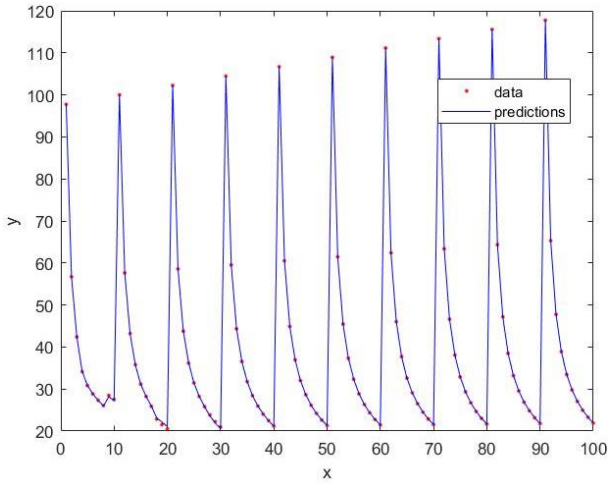


Figure 19 ArdMatern32 Kernel Function

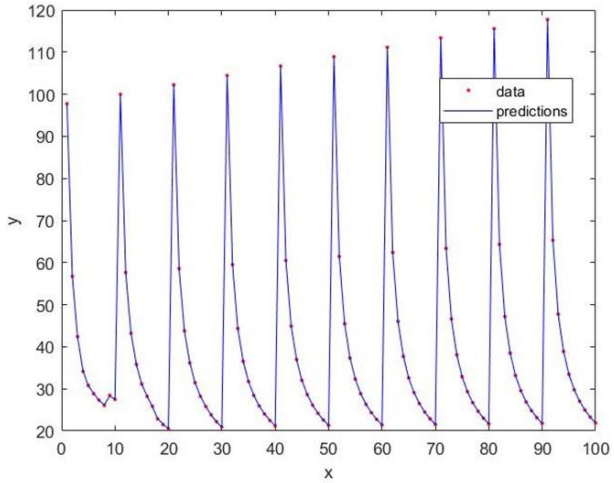


Figure 19 ArdMatern52

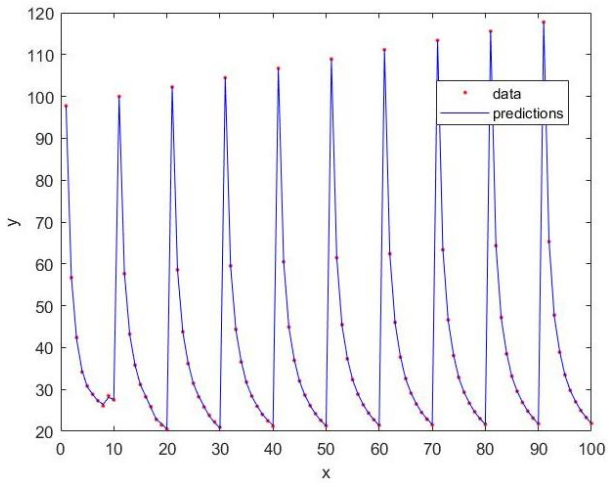


Figure 19 ArdRational quadratic

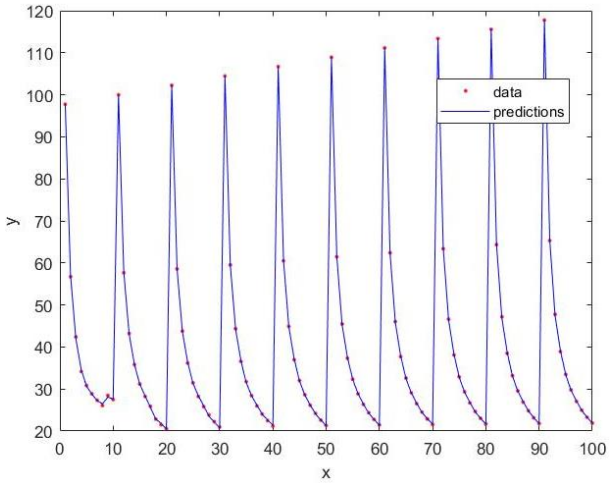


Figure 19 ArdSquared Exponential

The differences in fitting the data were not obvious in these graphs, so a 3d plot is created.

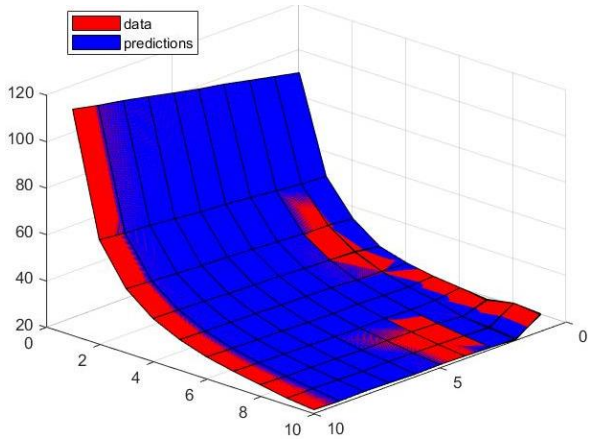


Figure 24 ArdExponential

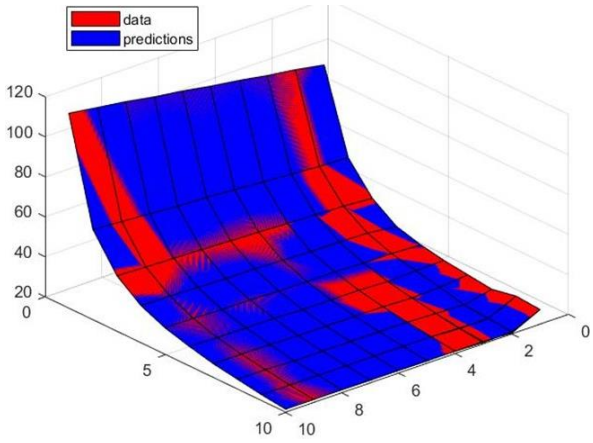


Figure 24 ArdMatern32

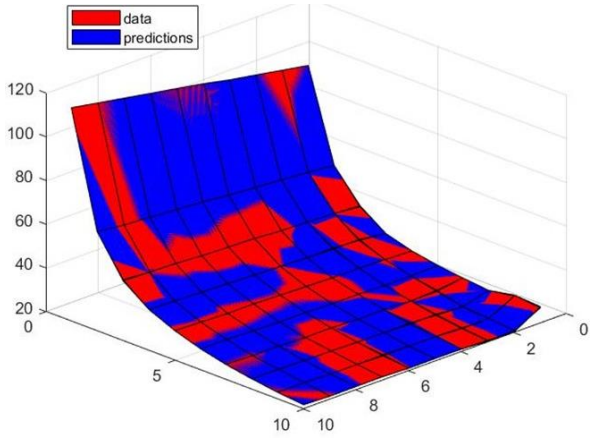


Figure 24 ArdMatern52

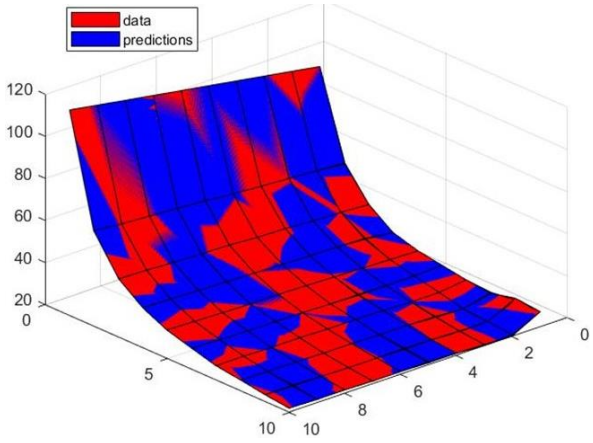


Figure 24 ArdRational Quadratic

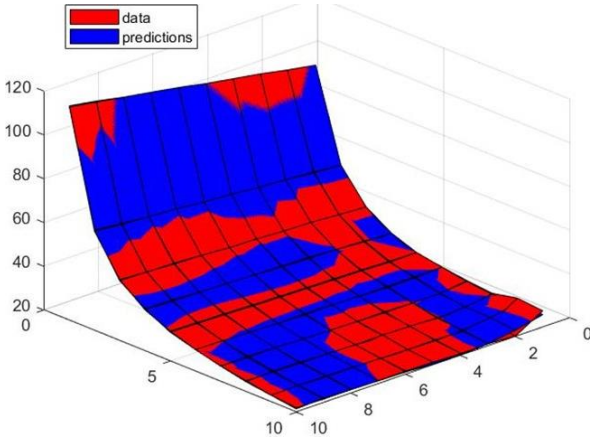
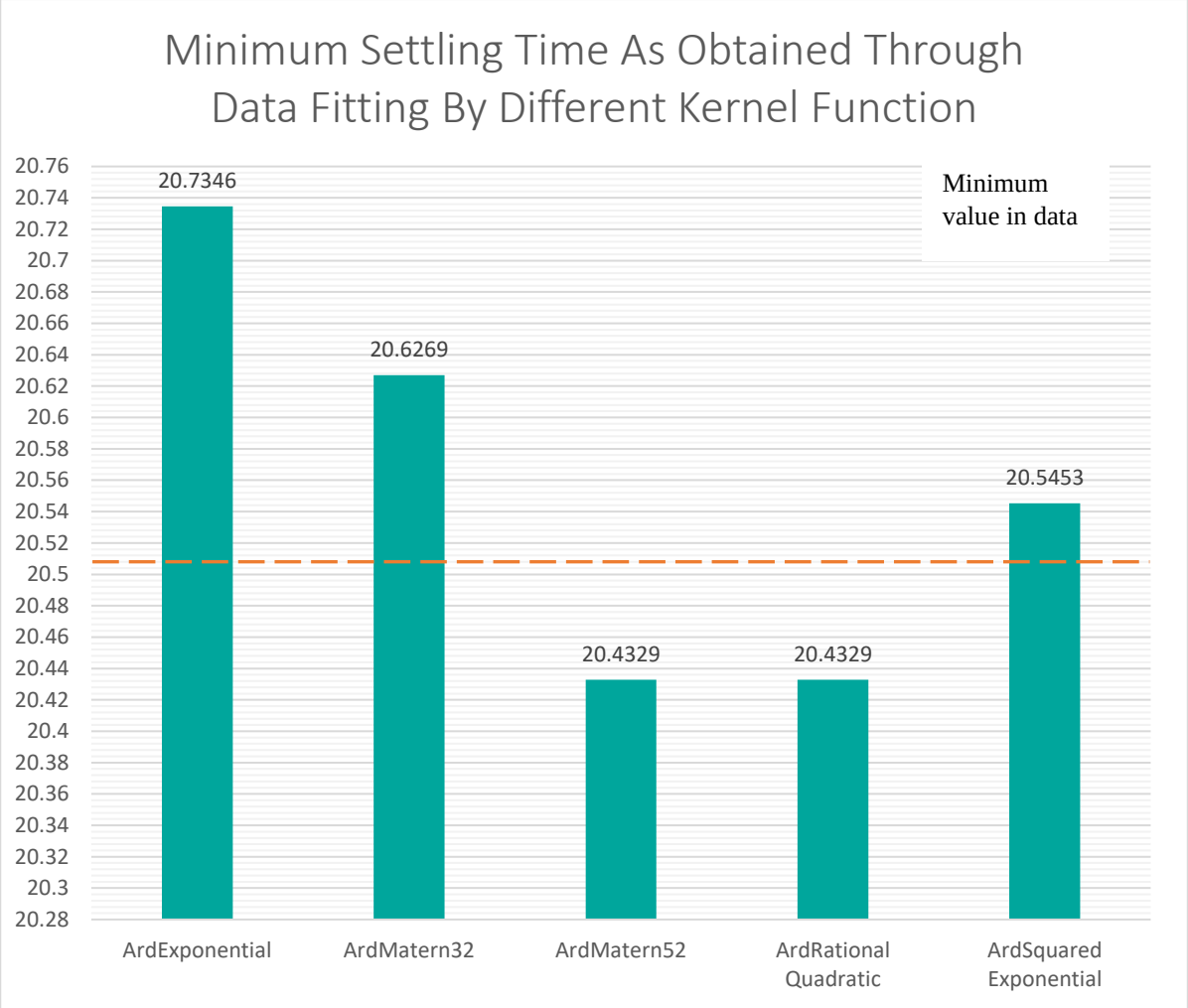


Figure 24 ArdSquared Exponential

Although now the differences between the actual data (red) and the predictions or the fitted data (blue) are obvious, there is still not a considerable difference between all kernel functions results.


Minimum Settling Time As Obtained Through Data Fitting By Different Kernel Function
20.76
20.74
20.72
20.7
20.68
20.66
20.64
20.62
20.6
20.58
20.56
20.54
20.52
20.5
20.48
20.46
20.44
20.42
20.4
20.38
20.36
20.34
20.32
20.3
20.28
20.7346
20.6269
20.4329
20.4329
20.5453
Minimum value in data
ArdExponential
ArdMatern32
ArdMatern52
ArdRational Quadratic
ArdSquared Exponential

BayesOpt

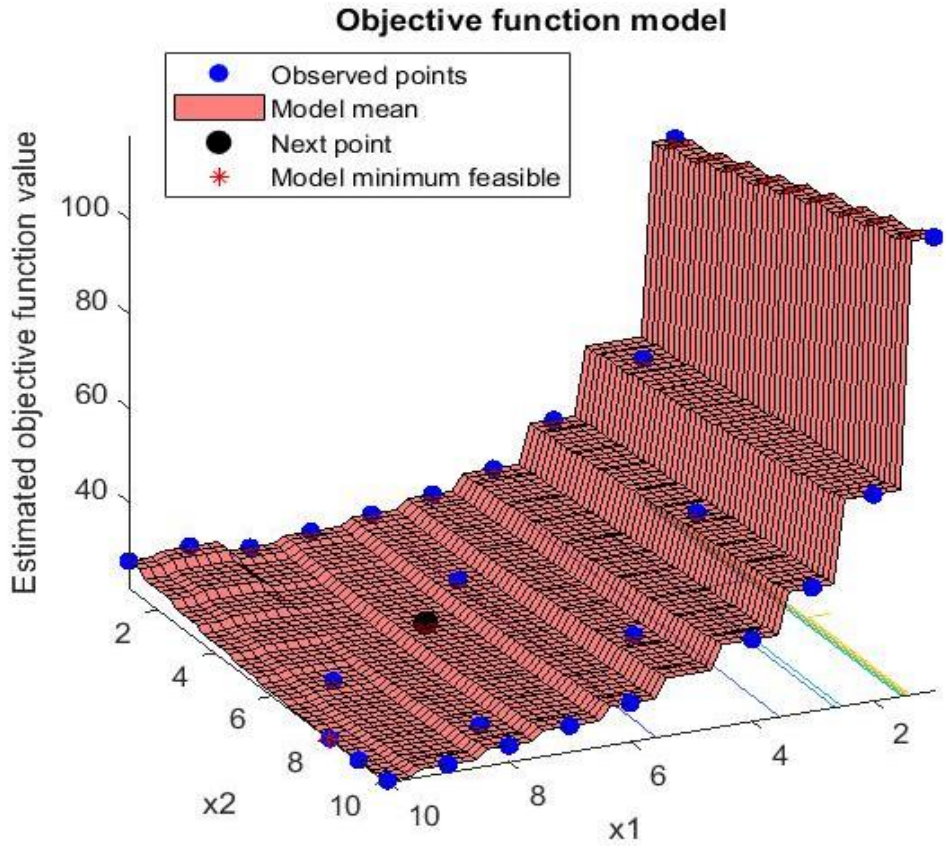
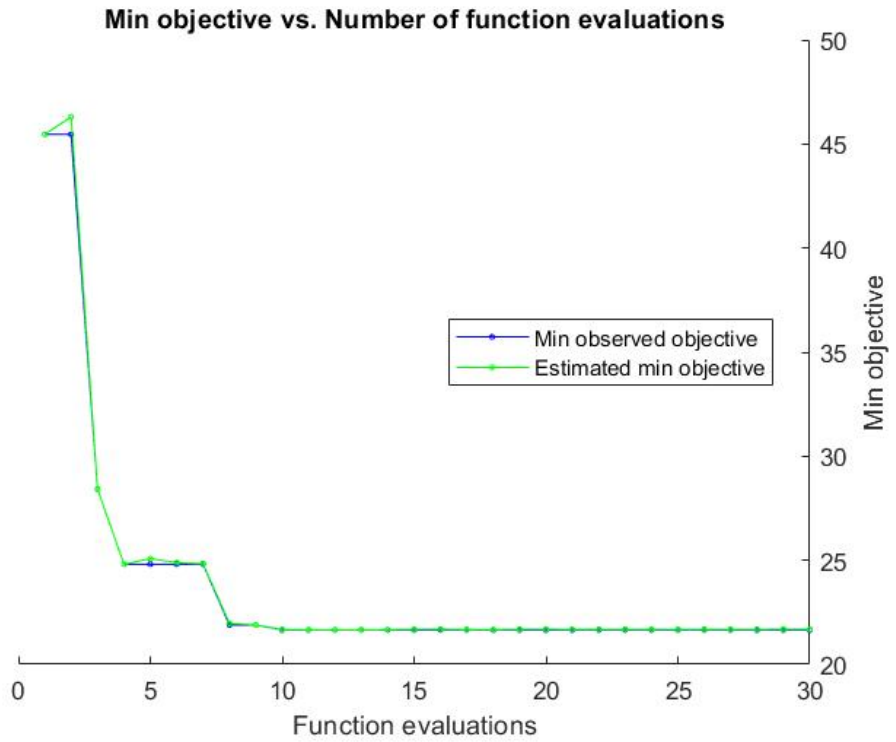


Figure 25 Function Model by MATLAB's BayesOpt



The output changed every run time. Minimum settling time is 21.7665 seconds

2D Bayesian Algorithm

According to the number of observations asked, the accuracy of the result changed.

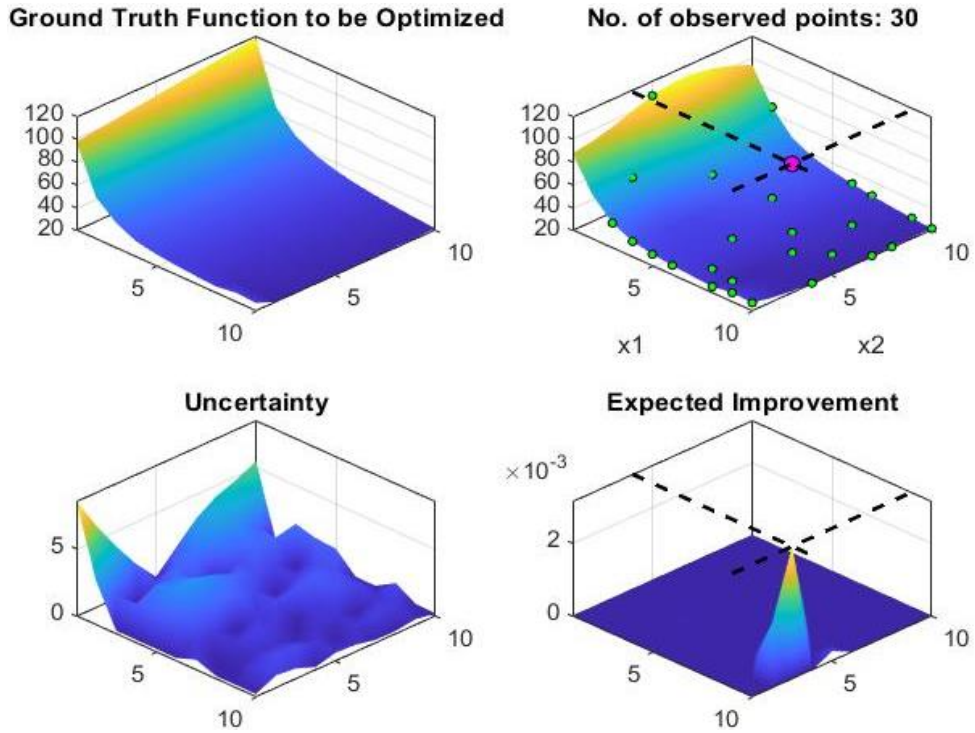


Figure 26 Final result of the 2D Simulation, using the expected improvement acquisition function for 30 observations

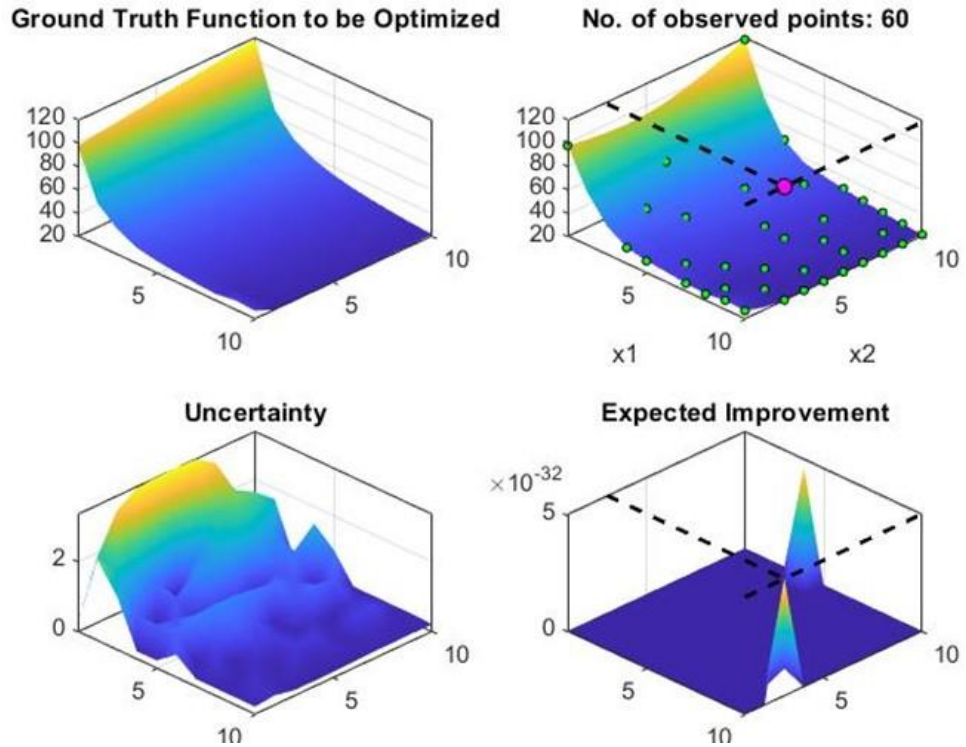


Figure 27 Final result of the 2D Simulation, using the expected improvement acquisition function for 30 observations

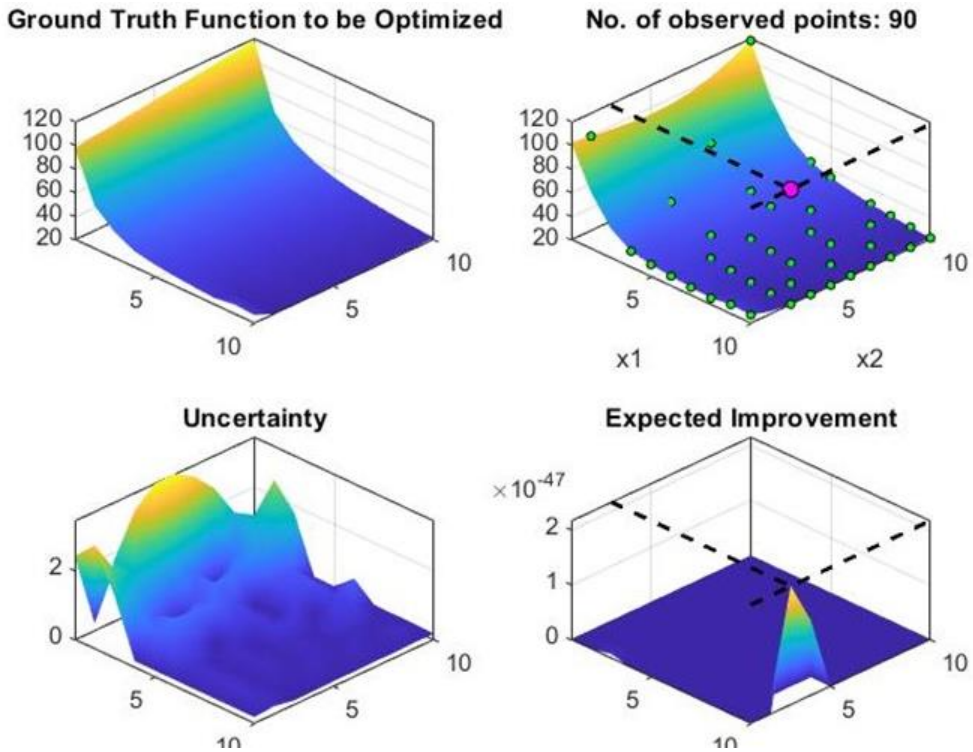


Figure 29 Final result of the 2D Simulation, using the expected improvement acquisition function for 30 observations

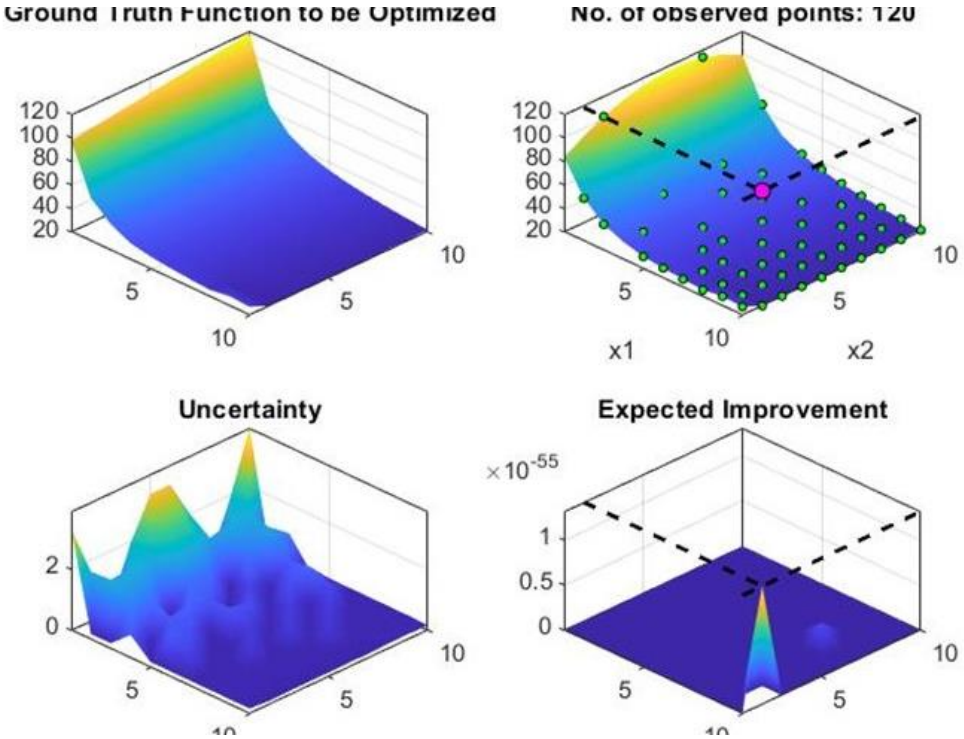
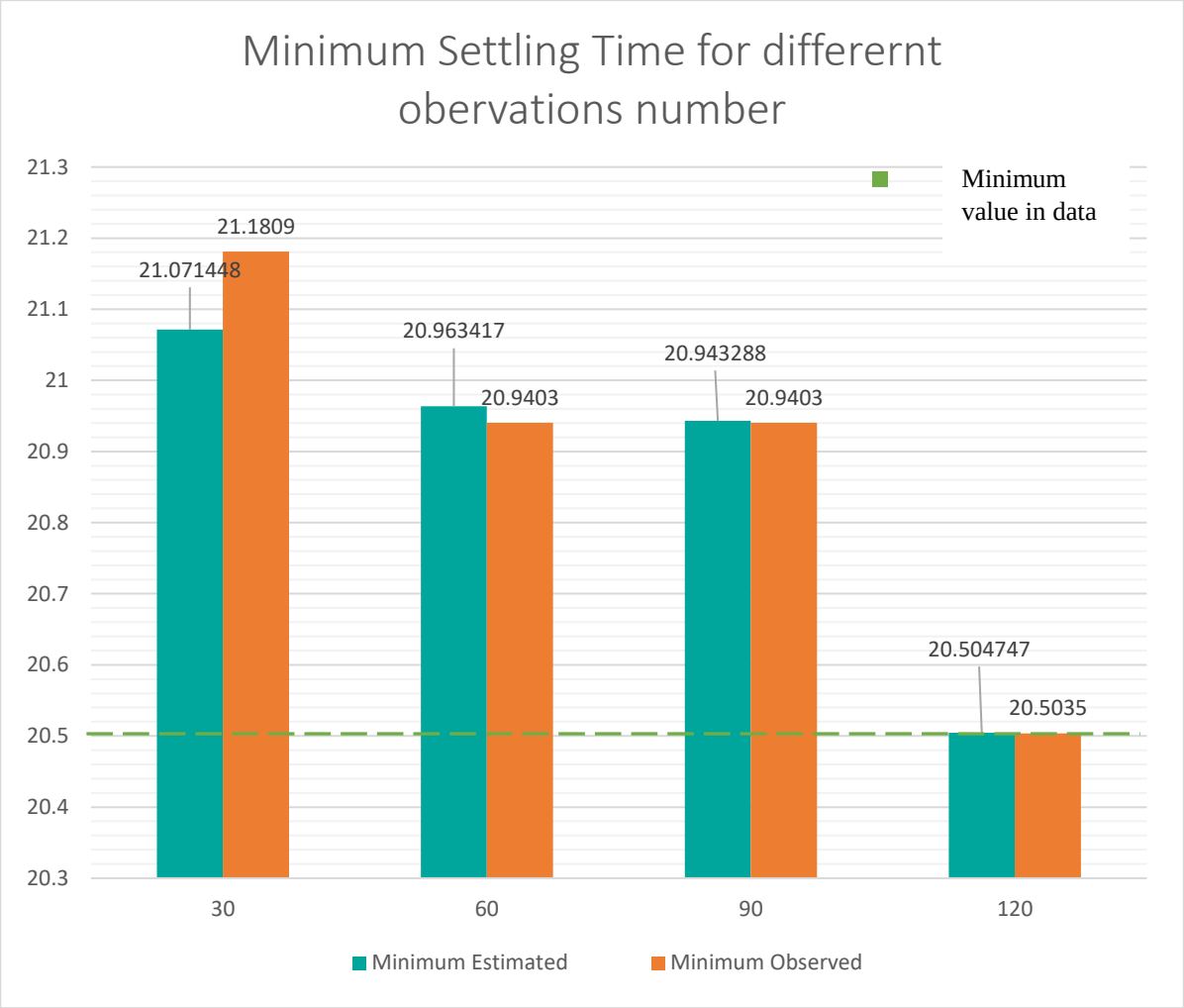
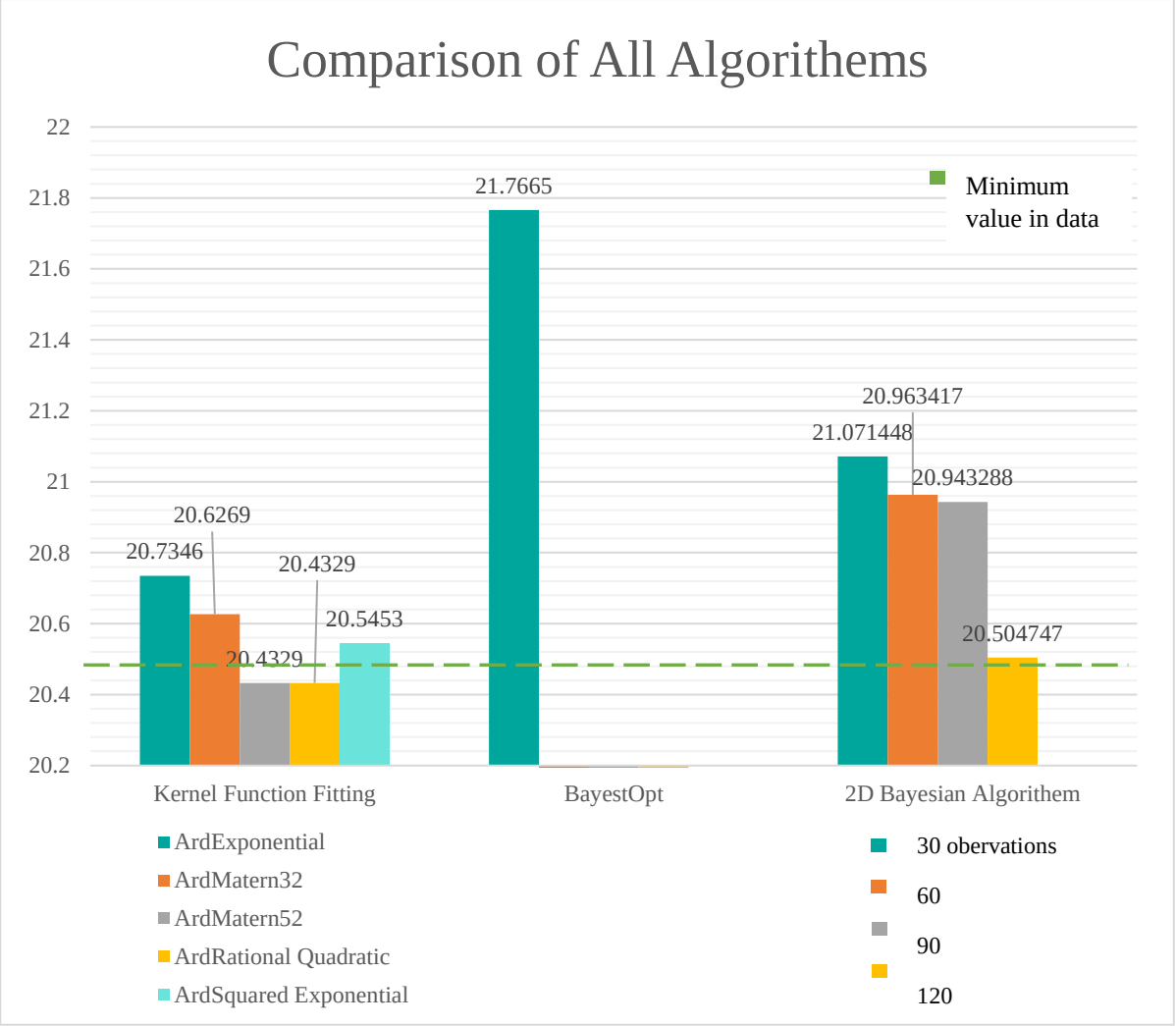


Figure 28 Final result of the 2D Simulation, using the expected improvement acquisition function for 30 observations


Minimum Settling Time for differernt obervations number
21.3
21.2
21.1
21
20.9
20.8
20.7
20.6
20.5
20.4
20.3
21.071448
21.1809
20.963417
20.9403
20.943288
20.9403
20.504747
20.5035
Minimum value in data
30
60
90
120
Minimum Estimated
Minimum Observed



The above chart represents the minimum settling time found by different algorithms with different settings. The dashed line represented the minimum settling time present in the original data, for reference.

Conclusions and Future Work

Bayesian optimization has proved to be efficient in many applications. Most notable of these applications are:

- Google's AutoML - use bayesian optimization
- AlphaGo Zero - self-play using Gaussian Process optimization (Bayesian optimization)
- Google Vizer - Internal use services to tune hyper-parameter
- Amazon SegaMaker
- Others including: robotics, automatic machine learning, hierarchical reinforcement learning.

Bayesian optimization is a big field of study with a great number of hyperparameters and combinations of methods that can be used. It is still a relatively new field and there is a lot of effort dedicated into exploring it. For further work, turning of a PID controller for a single output can be explored. Furthermore, to tune other types of controllers like the LQR, or H infinity controllers. And to explore complicated control systems that involve more than 1 controllable output. Focus needs to be directed towards understanding and comparing the different acquisitions functions that can be used and their generality ad suitability to types of problems.

References

A. Ye, “The beauty of Bayesian optimization, explained in simple terms,” *Medium*, 31-Oct-2020. [Online]. Available: https://towardsdatascience.com/the-beauty-of-bayesian-optimization-explained-in-simple-terms-81f3ee13b10f.

“Bayesopt,” *Bayesian Optimization Algorithm - MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/stats/bayesian-optimization-algorithm.html.

C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2006

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4):455–492, 1998.

D. Duvenaud, Automatic model construction with Gaussian processes, Ph.D. thesis, University of Cambridge (2014).

D. Duvenaud, The Kernel Cookbook, Available at https://www.cs. toronto.edu/~duvenaud/cookbook.

“Function,” *Select optimal machine learning hyperparameters using Bayesian optimization - MATLAB*. [Online]. Available: https://www.mathworks.com/help/stats/bayesopt.html#d124e152771.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. Towards Global Optimization, 2(117-129):2, 1978.

J. Görtler, R. Kehlbeck, and O. Deussen, “A visual exploration of Gaussian processes,” *Distill*, 17-Dec-2021. [Online]. Available: https://distill.pub/2019/visual-exploration-gaussian-processes/.

J. Wang, *An Intuitive Tutorial to Gaussian Processes Regression*, Ontorio, 2022.

J. Gonz´alez, “Introduction to Bayesian Optimization,” in *Masterclass*, 07-Feb-2017.

H. Jin, “Bayesian Optimization ,” 16-Nov-2018.

Karl Ezra Pilario (2023). Tutorial: Bayesian Optimization (https://www.mathworks.com/matlabcentral/fileexchange/114950-tutorial-bayesian-optimization), MATLAB Central File Exchange. Retrieved January 23, 2023.

O. Knagg, “An intuitive guide to Gaussian Processes,” *Medium*, 15-Jan-2019. [Online]. Available: https://towardsdatascience.com/an-intuitive-guide-to-gaussian-processes-ec2f0b45c71d.

R. turner et al. (2021). Bayesian Optimization Is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. Proceedings of the Neurips 2020 Competition and Demonstration Track.

T. Mortenson, “PID controller explained,” RealPars, 08-Oct-2022. [Online]. Available: https://realpars.com/pid-controller/. [Accessed: 04-Dec-2022].

Appendix

Mathematical Model

The equations that conform the 9-DOF non-linear dynamical model assume the following hypothesis:

1. The whole aircraft is assumed to be rigid body; it means that the distance between any two points in the airframe remains constant. This is a fundamental condition because it allows to understand the movement of the vehicle as a translation and a rotation around the centre of mass independently.
2. The rotational movement of the Earth is negligible with respect to the accelerations on the vehicle. i.e., the Earth frame is an inertial frame of reference.
3. The atmosphere is assumed to be calm (no wind or turbulence)

Geometry