Adding Flight Mechanics to Flight Loads Surrogate Model using Multi-Output Gaussian Processes

Ankit Chiplunkar* and Michele Colombo †

Airbus Operations S.A.S., Toulouse, 31060, France

Emmanuel Rachelson‡

ISAE-Supaero, Département d'Ingénierie des Systèmes Complexes (DISC), Toulouse, 31055, France

Joseph Morlier§

Université de Toulouse, Institut Clement Ader (ICA), CNRS, ISAE-SUPAERO, Toulouse, 31077, France

In this paper analytical methods to formally incorporate knowledge of physics-based equations between multiple outputs in a Gaussian Process (GP) model are presented. In Gaussian Processes a multi-output kernel is a covariance function over correlated outputs. Using a general framework for constructing auto- and cross-covariance functions that are consistent with the physical laws, physics-based relationships among several outputs can be imposed. Results of the proposed methodology for simulated data and measurement from flight tests are presented. The main contribution of this paper is the application and validation of our methodology on a dataset of flight tests, while imposing knowledge of flight mechanics into the model.

Nomenclature

y_i	Output Data	f_i	Latent Function
x_i	Input Data	$g(.)$	Relation between outputs
$m(x)$	Mean function for a Gaussian Process	$k(x, x', \theta)$	Covariance function for a Gaussian Process
K_{xx}	Auto-covariance matrix for points x	$k_{x_* x}$	Covariance Matrix at points x_* and x
θ	Hyperparameters for a Gaussian Process	L	Jacobian matrix of $g(.)$
x_*	Prediction points	y_*	Prediction at point x_*
X	Joint Input Data Matrix	Y	Joint Output Data Vector
η	Spanwise distance from wing root	C_z	Lift coefficient in z direction
k_z	Aerodynamic force in z direction	ϵ_i	Measurement error
σ_i	Variance of measurement error	i	Output number
d	Absolute distance between points	μ_{y_2}	Mean value of the process y_2
η_{root}	Spanwise position at fuselage and wing joint	η_{edge}	Spanwise position at wing edge

I. Introduction

Precise values of flight loads are highly useful information at several stages of an aircraft's life cycle. During certification, aircraft manufacturers need experimental flight loads to verify and improve their the-

*PhD Candidate, Flight Physics Airbus Operations, 316 route de Bayonne 31060

†Loads and Aeroelastics Engineer, Flight Physics, 316 route de Bayonne 31060

‡Associate Professor, Université de Toulouse, ISAE-SUPAERO, Département d'Ingénierie des Systèmes Complexes (DISC), 10 Avenue Edouard Belin, 31055 TOULOUSE Cedex 4, France

§Professor, Université de Toulouse, Institut Clement Ader (ICA), CNRS, ISAE-SUPAERO, 10 avenue Edouard Belin 31077 TOULOUSE Cedex 4

oretical models. Monitoring aircraft usage through loads can provide a better estimate of the remaining time-to-failure of components against their prescribed fatigue life. Aircraft flight loads are often difficult and expensive to measure experimentally. This calls for a need to develop a theoretical framework to merge different types of measurements and produce a robust flight loads surrogate model.

A. Background

Identification of an accurate flight loads surrogate model over the complete flight domain is an expensive exercise, notably because one cannot perform extensive flight testing on all the flight configurations. Since the last decade, techniques such as Neural Networks[1,2] have been applied while recently Gaussian Process Regression[3] (also known as Kriging) has shown great promise in developing the surrogate models that infer loads at untested flight configurations.

It has been proved extensively that introduction of prior information[4] improves the prediction capabilities of a GP surrogate model, diminishes chances of over-fitting and helps in identifying outliers. In recent times, techniques such as Gradient Enhanced Kriging (GEK)[5,6] have been used successfully in various engineering systems to impose prior information of gradients or local linear models, thereby improving the performance of surrogate models.

Building surrogate models on multi-output data is also known as co-kriging[7] or multi-output GP.[8] Modelling multi-outputs is particularly difficult because we need to construct auto- and cross-covariance functions between different outputs based on the relationships between them. Recently, it has been shown that relationships between multiple outputs can be imposed by encoding it in the covariance functions[9] to derive models consistent with relationship and data.

B. Contribution

In this paper we extend the framework of Gradient Enhanced Kriging to integral enhanced kriging, quadratic enhanced kriging or "any relationship between outputs" enhanced kriging. Using a general framework to encode relationships between outputs[9] , we will impose prior knowledge into the Gaussian Process Regression framework and apply it for predicting flight loads.

In this paper we assume one output to be independent while the remaining outputs are linked to the first output using physics-based relationships. For instance, lift coefficient (C_z) and spanwise aerodynamic force (k_z) have an integral relationship between them equation 21. As will be described in subsection II.B, the auto- and cross-covariance functions between these outputs are evaluated exactly if the physics-based relation between them is linear[10] or using approximate joint-covariance for non-linear relationship between the outputs.[9] The main contribution of this paper is the application and validation of our methodology on a dataset of flight tests, while imposing knowledge of flight mechanics into the model.

II. Theoretical Setup

Let us start by defining a P-dimensional input space and a D-dimensional output space. Such that $\{(x_i^j, y_i^j)\}$ for $j \in [1; n_i]$ is the training dataset for the i^{th} output. Here, n_i is the number of measurement points for the i^{th} output, while $x_i^j \in \mathbb{R}^P$ and $y_i^j \in \mathbb{R}$. We next define $x_i = \{x_i^1; x_i^2; \ldots; x_i^{n_i}\}$ and $y_i = \{y_i^1; y_i^2; \ldots; y_i^{n_i}\}$ as the full matrices containing all the training points for the i^{th} output such that $x_i \in \mathbb{R}^{n_i \times P}$ and $y_i \in \mathbb{R}^{n_i}$.

$$y_i^j = f(x_i^j) + \epsilon_i^j \tag{1}$$

The science of finding a mapping function f denoted by equation 1 between $\{(x_i^j, y_i^j)\}$ is called surrogate modeling in engineering design or regression in machine learning. We would use the technique of kriging (engineering design)[11] or Gaussian Process Regression (machine learning)[4] to learn this mapping function f. A Gaussian Process is a probability distribution over infinitely many points, such that any finite subset of the process is a multi-variate gaussian distribution. A GP can be fully parameterized by a mean and covariance function equation 2.

$$f(x) = GP(m(x, \theta), k(x, x', \theta)) \tag{2}$$

A random draw from a multi-variate gaussian distribution yields a random vector around its mean with values correlated through the covariance matrix. Similarly, a random draw from a GP provides a random function around the mean function $m(x, \theta)$ and of the shape as defined by the covariance function $k(x, x', \theta)$ (calculated between points x and x'). Hence, GPs provide us a method to define a family of functions whose shape is defined by its covariance function[12] also called kernel.

A. Gaussian Process Regression (GPR)

For the case of a single-output GP we assume that D equals to one, so our dataset will be $\{(x_1, y_1)\}$. We first start by defining a white noise measurement error between the measured outputs y and mapping function f.

$$\epsilon = \mathcal{N}(0, \sigma^2_{noise} I) \tag{3}$$

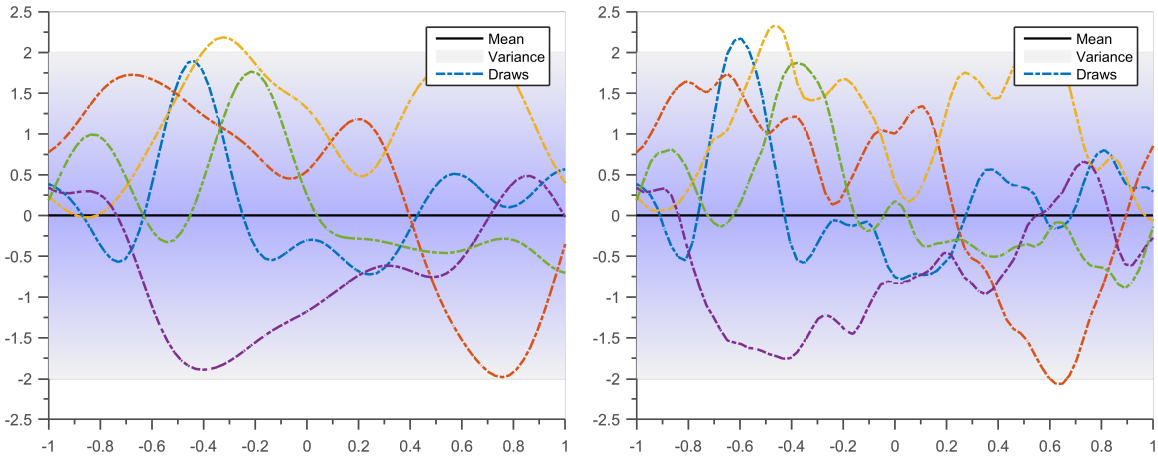
Here, σ_{noise} is the magnitude of measurement error. While performing regression in a GPR framework we first define a family of functions also called a prior described in subsection II.A.1. The next step involves taking the data and eliminating all the functions in our prior which do not obey the observations, this step gives us the posterior mean and covariance described in subsection II.A.2. Finally, we can further improve our predictions by improving our initial family of functions, this is achieved by maximizing the marginal likelihood subsection II.A.3.

1. *Prior*

Any regression algorithm starts with an initial family of functions or hypothesis space. By initially defining a family of functions we indirectly encode initial assumptions or bias (continuity, differentiability etc.) into our regression algorithm. If an algorithm is not able to represent the mapping function f in its hypothesis space we will find the closest according to a certain metric.

$$f(x) = GP(0, k(x, x', \theta)) \tag{4}$$

Here, the representation of mapping function f equation 4 defines a family of functions also called prior. Without loss of generality, we can assume that the mean function is zero.[4] The covariance function or kernel $k(x, x')$ defines the shape of functions defined by the prior equation 4. In this paper we will compare two types of kernels Standard Exponential kernels and Matérn kernels.



(a) Draws from a Standard Exponential (SE) Kernel ($\theta = [1, 0.2]$).

(b) Draws from a Mateŕn Kernel differentiable twice ($\theta = [1, 0.2]$)

Figure 1: Draws from Gaussian Process Priors. Solid black line defines the mean function, shaded blue region defines 95% confidence interval (2σ) distance away from mean. The dashed lines are functions randomly drawn from the priors

Standard Exponential Kernel

A popular choice of covariance function is a Squared Exponential (SE) function equation 5, because it defines a family of highly smooth (infinitely differentiable) non-linear functions as shown in figure 1a.

$$k_{SE}(x, x', \theta) = \theta_1^2 exp[-\frac{d^2}{2\theta_2^2}] \tag{5}$$

For the case of the SE kernel equation 5 the hyperparameters ($\theta = [\theta_1, \theta_2]$) are; amplitude θ_1 which defines average distance from mean and the length scale θ_2 which defines the wiggliness of functions. Here, d defines the absolute distance between points $|x - x'|$.

Matérn Kernel

The Mateŕn kernel is the second most popular kernel after the squared exponential kernel. It provides the flexibility to define a family of functions with varying degree of differentiability. We will take the case of twice differentiable Mateŕn kernel throughout this article.

$$k_{Mat2}(x, x', \theta) = \theta_1^2 \left(1 + \frac{\sqrt{5}d}{\theta_2} + \frac{5d^2}{3\theta_2^2}\right) exp\left[-\frac{\sqrt{5}d}{\theta_2}\right] \tag{6}$$

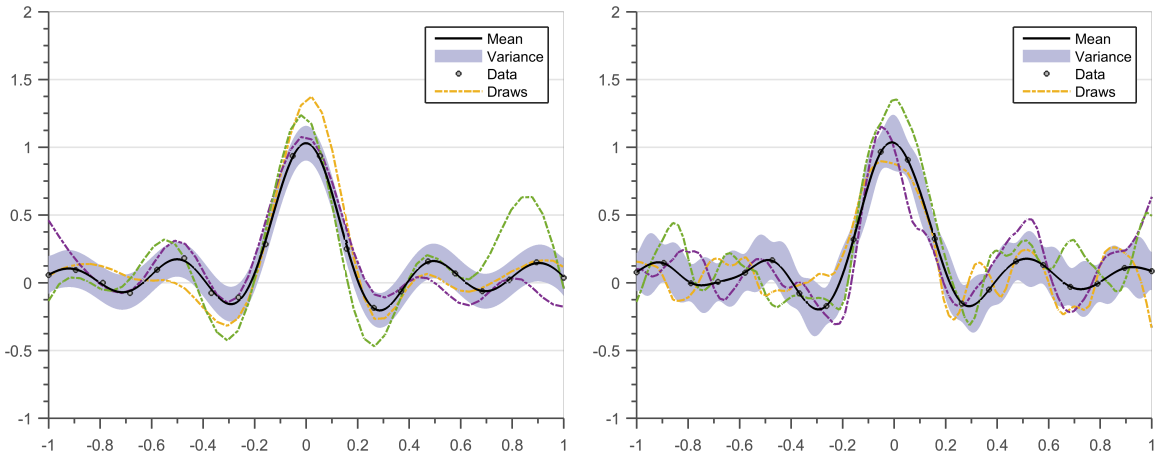
The hyperparameters ($\theta = [\theta_1, \theta_2]$) are; the amplitude θ_1 which defines average distance from mean and the length scale θ_2 which defines the wiggliness of functions. The Matérn kernel in equation 6 defines a family of functions differentiable only twice figure 1b. We can observe that draws from figure 1a are smoother when compared to draws from figure 1b for the same set of hyperparameters θ.

2. *Posterior*

Once we have defined an appropriate prior, we have a look at the data $\{(x_1, y_1)\}$. Using the Bayes rule we eliminate all the functions that are not passing through the observed data points. The predicted mean equation 7 and variance equation 8 can be analytically calculated[4] .

$$m(y_*) = k_{x_*x}(K_{xx} + \sigma_{noise}^2 I)^{-1} y \tag{7}$$

$$Cov(y_*) = k_{x_*x_*} - k_{x_*x}\ (K_{xx} + \sigma_{noise}^2 I)^{-1} k_{xx_*} \tag{8}$$



(a) Posterior from a Standard Exponential (SE) Kernel ($\theta = [1, 0.2]; \sigma_{noise} = 0.05$).

(b) Posterior from a Mateŕn Kernel differentiable twice ($\theta = [1, 0.2]; \sigma_{noise} = 0.05$).

Figure 2: Gaussian Process Posterior. Solid black line defines the mean function, shaded blue region defines 95% confidence interval (2σ) distance away from mean. The dashed lines are functions randomly drawn from the posterior

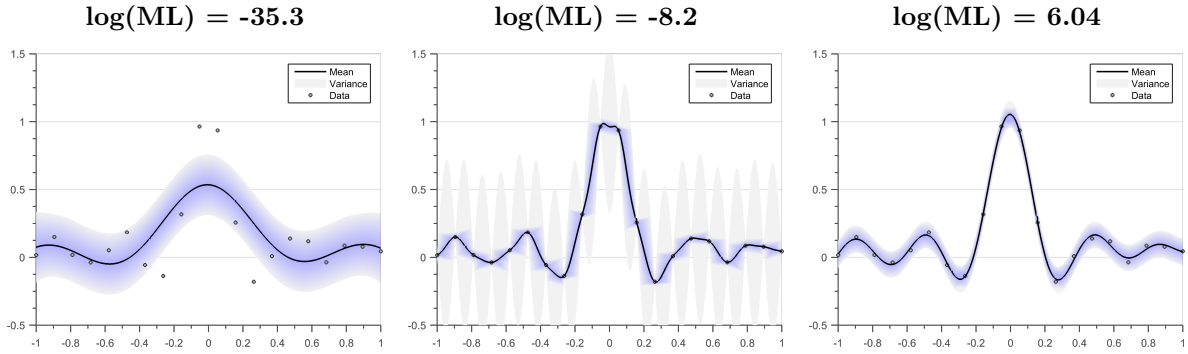
Figure 2a shows the posterior distribution upon introduction of data to the prior distribution figure 1a. The mean function does not pass through all the observation points because of the assumed measured noise term $\sigma_{noise} = 0.05$. Figure 2b shows the posterior distribution upon introduction of data to the prior distribution 1b. The mean function does not pass through all the observation points because of the initial noise term $\sigma_{noise} = 0.05$. We can observe that the mean of figure 2a is smoother when compared to the mean of figure 2b. Moreover the variance estimates in the case of the Matérn kernel are higher than that of the SE kernel. Since twice differentiability is a less strict initial assumption and has more functions in its family, more functions are possible through the data.

3. Maximizing Marginal Likelihood

To obtain good prediction capabilities, one needs to start from a prior that explains well our dataset. This is achieved by optimizing the Marginal Likelihood (ML) equation 9. The probability that our dataset comes from a family of functions defined by the prior is called the ML.[4] Hence, when we maximize the ML we are actually finding the optimal θ or family of functions that best describe our dataset.

$$\log(\mathbb{P}(y \mid x, \theta)) = -\frac{1}{2} y^T [K_{xx} + \sigma_{noise}^2 I]^{-1} y - \log \left| K_{xx} + \sigma_{noise}^2 I \right| - \frac{n}{2} \log(2\pi) \tag{9}$$

The ML is a trade-off between a data-fit term ($\frac{1}{2} y^T K_{xx}^{-1} y$) and a model complexity term ($\log |K_{xx}|$). The optimization of ML(θ) provides the best compromise in terms of explaining the existing dataset $\{(x_1, y_1)\}$ and the initial assumptions encoded in the prior. Hence to find the best set of hyperparameters we maximize the marginal likelihood equation 9 with respect to its hyperparameters.



(a) Posterior between SE prior with hyperparameters ($\theta = [0.05, 0.5]; \sigma_{noise} = 0.01$) and data.

(b) Posterior between SE prior with hyperparameters ($\theta = [0.05, 0.5]; \sigma_{noise} = 0.01$) and data.

(c) Posterior between SE prior with optimized hyperparameters ($\theta = [0.45, 0.17]; \sigma_{noise} = 0.0014$) and data.

Figure 3: Posteriors for 3 different sets of hyperparameters

Figure 3 compares the posterior distributions obtained for SE priors with 3 different hyperparameters. We observe that the mean of figure 3b passes through all the observed data points but is more complex (wiggly). The mean in figure 3a is a smooth function but does not fit the data properly. While figure 3c is a good compromise between the data-fit term and the model complexity term.

B. Multi-Output Gaussian Process Regression

Given a dataset for multiple outputs $\{(x_i, y_i)\}$ for $i \in [1; D]$ we define the joint output vector $Y = [y_1; y_2; y_3; \ldots; y_D]$ such that all the output values are stacked one after the other. Similarly, we define the joint input matrix as $X = [x_1; x_2; x_3; \ldots; x_D]$. For the sake of simplicity, suppose we measure two outputs y_1 and y_2 with some error, while the true physical process is defined by latent variables f_1 and f_2. Then the relation between the output function, measurement error and true physical process can be written as follows.

$$\begin{aligned} y_1 &= f_1 + \epsilon_{n1} \\ y_2 &= f_2 + \epsilon_{n2} \end{aligned} \tag{10}$$

Let us take the case of an explicit relationship between the two latent variables equation 11.

$$f_1 = g\left(f_2, x_1\right) \tag{11}$$

Here $g\left(.\right) \in \mathcal{C}^2$ is an operator defining the relation between f_1 and an independent latent variable f_2. Operator $g(.)$ can be a known physical equation or a computer code between the outputs. A GP prior in such a setting with 2 output variables is expressed in equation 12.

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sim GP\left[\begin{pmatrix} g(0) \\ 0 \end{pmatrix}, \quad \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}\right] \tag{12}$$

K_{12} and K_{21} are cross-covariances between the two inputs x_1 and x_2. K_{22} is the auto-covariance function of independent output, while K_{11} is the auto-covariance of the dependent output variable. The full covariance matrix K_{XX} is also called the joint-covariance. While, the joint error matrix will be denoted by Σ;

$$\Sigma = \begin{bmatrix} \sigma_{n1}^2 & 0 \\ 0 & \sigma_{n2}^2 \end{bmatrix} \tag{13}$$

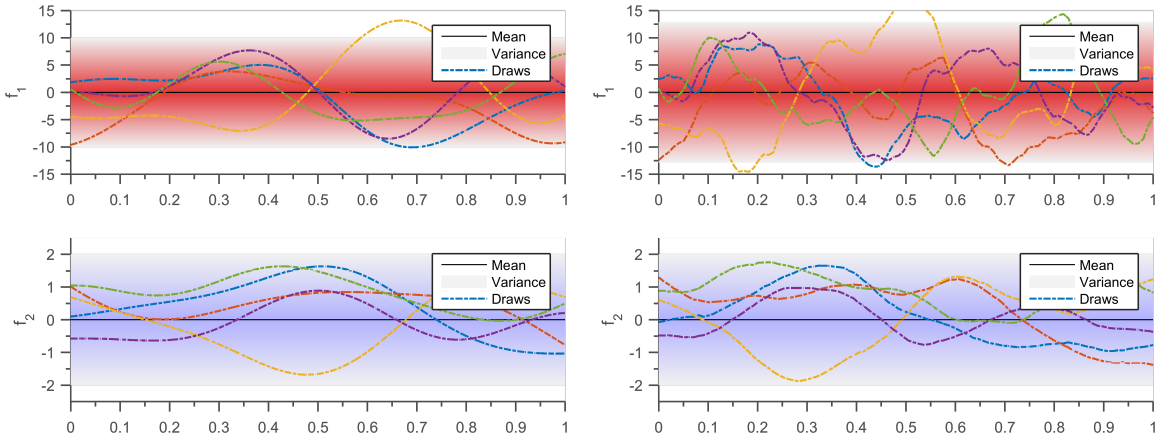
Where, ϵ_{n1} and ϵ_{n2} are measurement error sampled from a white gaussian noise $\mathcal{N}(0, \sigma_{n1})$ and $\mathcal{N}(0, \sigma_{n2})$.

1. *Multi-Output Prior*

For a linear operator $g(.)$ the joint-covariance matrix can be derived analytically,[13] due to the affine property of Gaussian's.

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sim GP\left[\begin{pmatrix} g(0) \\ 0 \end{pmatrix}, \quad \begin{pmatrix} g(g(K_{22}, x_2), x_1) & g(K_{22}, x_1) \\ g(K_{22}, x_2) & K_{22} \end{pmatrix}\right] \tag{14}$$

Using the known relation between outputs we have successfully correlated two GP priors from equation 11. This effectively means that when we randomly draw a function f_2 it will result in a correlated draw of f_1 such that the two draws satisfy the equation 11. We have effectively represented the covariance function K_{11} in terms of the hyperparameters of covariance function K_{22} using the known relation between outputs.



(a) Prior from a joint SE Kernel such that ($\theta = [1, 0.2]$) and $f_1 = \mathrm{d}f_2/\mathrm{d}x$.

(b) Prior from a joint Matern Kernel differentiable twice such that ($\theta = [1, 0.2]$) and $f_1 = \mathrm{d}f_2/\mathrm{d}x$.

Figure 4: Multi-Output Gaussian Process Prior. Solid black line defines the mean function, shaded regions define 95% confidence interval (2σ) distance away from mean. The dashed lines are functions randomly drawn from the prior. Pairs of similar colored dashed lines follow the given relation between f_1 and f_2.

However, a non-linear operation $g(.)$ on a Gaussian distributed variable does not result in a Gaussian distributed variable. Hence, for the case of non-linear $g\left(.\right)$ the above joint-covariance matrix as derived in Equation 14 is not positive semi-definite, which prevents us from taking its inverse in equation 9. Therefore we use an approximate joint-covariance[9] for imposing non-linear relations equation 15.

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sim GP\left[\begin{pmatrix} g(0) \\ 0 \end{pmatrix}, \quad \begin{pmatrix} LK_{22}L^T & LK_{22} \\ K_{22}L^T & K_{22} \end{pmatrix}\right] \tag{15}$$

Here, $L = \frac{\partial g}{\partial y}\Big|_{y_2=\bar{y}_2}$ is the Jacobian matrix of $g(.)$ evaluated at the mean of independent output y_2. Equation 15 is a Taylor series expansion for approximating related kernels. Since a Taylor series expansion is constructed from derivatives of a function which are linear operations, the resulting approximated joint kernel is a Gaussian kernel.[9]

Figure 4 shows the joint prior for two functions f_1 and f_2 given a relation $f_1 = \mathrm{d}f_2/\mathrm{d}x$ between them. The pairs of similarly colored random draws follow the given equation. We can verify this fact by observing that f_1 tends to zero when f_2 tends to a maxima or minima. The draws from figure 4a are smoother when compared to draws from figure 4b.

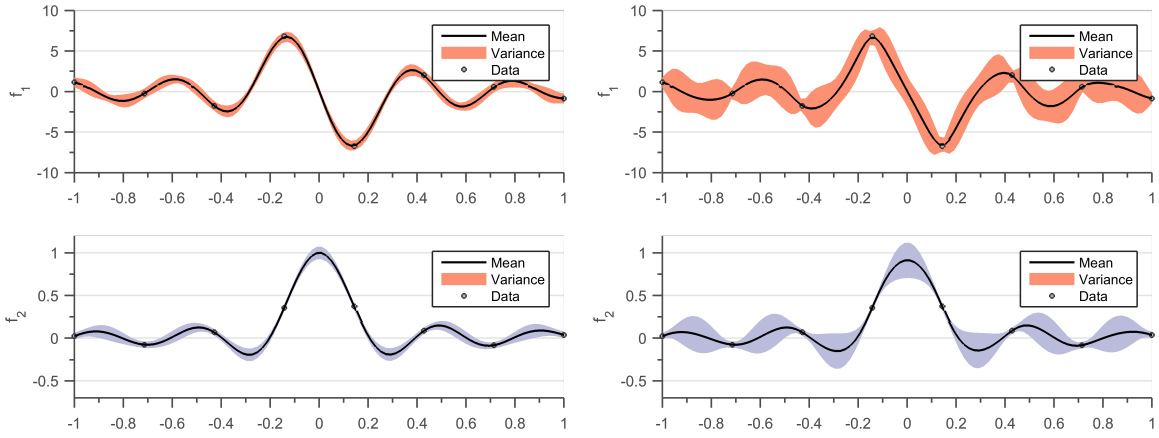
2. *Multi-Output Posterior*

To obtain predictions on non-measured points (X_*), we now proceed in a similar manner as prescribed in section A. We optimize the ML to find the optimal θ and then find the predicted mean and covariance as shown in equation 16 and equation 17.

$$m(Y_*) = K_{X_*X}(K_{XX} + \Sigma)^{-1}Y \tag{16}$$

$$Cov(Y_*) = K_{X_*X_*} - K_{X_*X}(K_{XX} + \Sigma)^{-1}K_{XX_*} \tag{17}$$

As mentioned earlier while calculating the posterior we apply the bayes rule. This effectively means that we eliminate all the functions that do not pass through measured points. Since we have encoded the relation between f_1 and f_2 in the joint-prior. When we eliminate not possible functions from the prior of f_1, we consequently eliminate functions from the prior of f_2 and vice-versa. In the process we come up with a posterior which is consistent with the observations and initial relation.



(a) Posterior from a joint SE Kernel such that ($\theta = [0.45, 0.17]$; $\sigma_{n2} = 0.02$; $\sigma_{n1} = 0.1$) and $f_1 = \mathrm{d}f_2/\mathrm{d}x$.

(b) Posterior from a joint Mateŕn Kernel differentiable twice such that ($\theta = [0.45, 0.17]$; $\sigma_{n2} = 0.02$; $\sigma_{n1} = 0.1$ and $f_1 = \mathrm{d}f_2/\mathrm{d}x$.

Figure 5: Multi-Output Gaussian Process Posterior. Solid black line defines the mean function, shaded region defines 95% confidence interval (2σ) distance away from mean. The dashed lines are functions randomly drawn from the posterior

Figure 5 shows the joint posterior for two functions f_1 and f_2 given a relation $f_1 = \mathrm{d}f_2/\mathrm{d}x$ and data between them. We have used same data as used in figure 2 but have reduced the number of measurement to 8 and derivative points to 8. Figure 5a shows the posterior for a SE kernel. At the positions where the value of derivative f_1 is given, we see a shrinking of variance in f_2. Draws from the posterior and mean of f_2 both have value of derivative as set by f_1. Figure 5b shows the posterior for SE kernel. At the positions where

value of derivative f_1 is given, we see a shrinking of variance in f_2. Draws from the posterior and mean of f_2 both have value of derivative as set by f_1.

Here the two outputs need not be measured at the same input points, this will be demonstrated in subsection III.B. Another interesting observation is that while optimizing the marginal likelihood for figure 5, the noise terms for two outputs are different ($\sigma_{n2} = 0.02$; $\sigma_{n1} = 0.1$). This means that the two outputs have different values of fidelity, this looks very similar to co-kriging. In fact co-kriging can be seen as a special case of joint multi-output GPR. Such that the two outputs are linked by the relation $f_1 = f_2$ and the two error measurements are linked by the relation $\sigma_{n1} >> \sigma_{n2}$, the output y_1 is a low fidelity measurement when compared to output y_2.

III. Results

In this section, we provide a numerical illustration to the theoretical derivations in the earlier sections. We start with a synthetic problem where we try to learn the model over quadratic relationships. We compare the cross-validation error values of independent GPR with that of multi-output joint GPR. Finally, we compare the performance of our methods on flight-loads estimation for a horizontal tail plane.

The basic toolbox used for this paper is GPML provided with "Gaussian Process for Machine Learning",[4] we generate covariance functions to handle relationships as described in equations 14 and 15 using the "Symbolic Math Toolbox" in MATLAB 2014b. All experiments were performed on an Intel quad-core processor with 4Gb RAM.

A. Quadratic relation on Synthetic Data

We take the case of a quadratic operator $g(.)$ equation 18.

$$f_1 = f_2^2 \tag{18}$$

To generate the data we randomly draw a single function of f_2 as described in equation 19 for 50 equally spaced inputs between [-1, 1]. The data for f_1 is the calculated using equation 18. The latent functions f are then corrupted according to equation 19 which gives us the outputs $y's$. We finally hide the observations for y_1 in the domain $x = [-0.2, 0.2]$. We thus have our dataset for testing the performance of quadratic relationship.

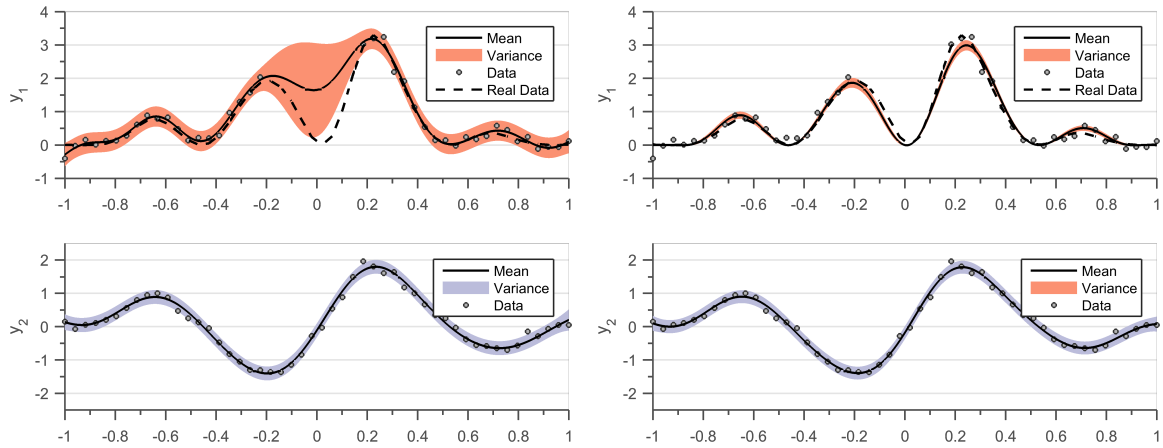
$$\begin{aligned} f_2 &\sim GP[0, K_{SE}(0.2, 1)] \\ \sigma_{n2} &\sim \mathcal{N}[0, 0.1] \\ \sigma_{n1} &\sim \mathcal{N}[0, 1] \end{aligned} \tag{19}$$

$K_{SE}(0.2, 1)$ means squared exponential kernel with length scale 0.2 and variance 1. σ_{n2} and σ_{n1} are the white noises added to the latent functions f_1 and f_2 respectively. Since the quadratic relationship $g(.)$ is non-linear in nature we use the equation 15 to calculate the auto- and cross-covariance functions as explained in equation 20.

$$\begin{aligned} K_{12} &= 2\mu_{y_2}(x_1)K_{22} \\ K_{11} &= 4\mu_{y_2}(x_1)^2 K_{22} \end{aligned} \tag{20}$$

Here, $\mu_{y_2}(x_1)$ is the mean value of function y_2 calculated at the input points x_1. For the case of quadratic relationship the jacobian L as described in equation 15 comes out to be $2\mu_{y_2}(x_1)$. For non-linear operators $g(.)$ the joint-covariance prior depends on the mean value of f_2. K_{22} and K_{11} are the auto-covariance functions calculated at the input points x_1 and x_2 as described in equation 15 and equation 12. K_{12} is the cross-covariance function between the input points x_1 and x_2. We can observe that the joint-covariance has been expressed as a function of covariance K_{22} using equation 18.

Figure 6a shows the independent fit of two GPR. For the case of y_1 there is a huge difference between the real data and predicted mean at points where data is unavailable. Figure 6b shows the joint-GP regression. Joint-GP model gives better prediction even in the absence of data for y_1 because transfer of information is happening from observations of y_2 present at those locations. For the case of y_2 there is no significant improvement in the prediction of two methods.



(a) Independent GP Regression for the two outputs y_1 and y_2. For y_1 the data is hidden from section $x = [-0.2, 0.2]$. We can observe the huge difference between the real data and the predicted mean values at zone with no data.

(b) Joint-GP Regression for the two outputs y_1 and y_2 related through equation 18. For y_1 the data is hidden from section $x = [-0.2, 0.2]$. We can observe the improved prediction between zone with no data because information is being shared between the two outputs.

Figure 6: GP Regression on Quadratic relationship. The solid black line represents the predicted mean while the shaded area denotes 95 % confidence interval (2σ) uncertainity region. The dashed black line represents the real value of f_1. For y_1 the data is hidden from section $x = [-0.2, 0.2]$.

Table 1: mean RMSE errors for quadratic relationship

	RMSE y_1	RMSE y_2
Independent Single-output GPR	1.74	0.14
Joint Multi-output GPR	0.26	0.12

Table 1 shows comparison of Root Mean Square Error (RMSE). 10 sets of experiments were run for 85% of data as training set and 15% of data as test set, the training and test sets were chosen randomly. We learn the optimal set of hyperparameters for on training data all 10 sets of experiments. Finally RMSE values are evaluated with respect to the test set. We see a significant improvement in performance for the case of joint multi-output GPR.

B. Flight Mechanics on Flight Test Data

Procuring data in engineering design is a costly exercise, a high-fidelity CFD simulation runs for days and a flight test campaign costs millions. In this context it is important to be data efficient while creating models with data. We leverage the knowledge of available relations between several types of data and build a more robust, physically consistent and data efficient model.

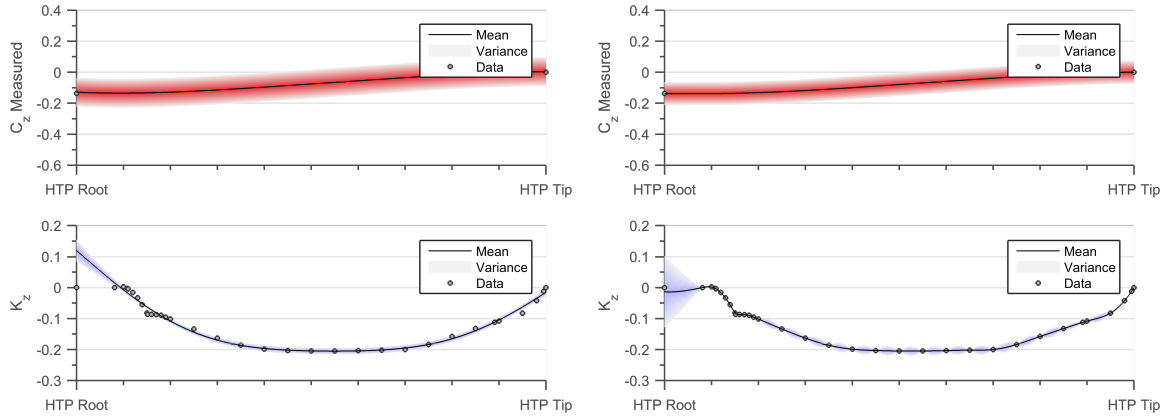
In this section we conduct experiments, applying our approach on the flight loads data. We look at normalized data of a simple longitudinal maneuver. The maneuver is quasi-static which means that airplane is in equilibrium at all times and there are no dynamic effects observed by the aircraft. The two outputs in our case are coefficient of lift C_z on the Horizontal Tail Plane (HTP) and spanwise lift k_z. We know that the integral of spanwise pressure will be equal to the coefficient of lift.

$$C_z(\eta) = \int_{\eta_{edge}}^{\eta_{root}} k_Z(\eta) d\eta \tag{21}$$

.

Here, η_{edge} denotes the edge of horizontal tail span while η_{root} represents root of tail. The above equation is linear in nature and hence we will use equation 14 to calculate the auto- and cross-covariance functions.

In practice, the coefficient of lift C_z is calculated using strain gauges on the HTP and the spanwise force is measured using pressure tappings. Chordwise pressures are measured at multiple η locations on the HTP, which are integrated chordwise to give spanwise lift distribution. We are trying to merge data coming from two different calculation processes of flight mechanics and aerodynamics using a basic physics based relation and improve the accuracy of our model.



(a) Joint multi-output GPR between K_z and C_z using a SE kernel and equation 21. Optimized hyperparameters ($\theta = [0.42, 0.08]$; $\sigma_{nKz} = 1.3 * 10^{-4}$; $\sigma_{nCz} = 0.0414$)

(b) Joint multi-output GPR between K_z and C_z using a twice differentiable Matérn kernel and equation 21. Optimized hyperparameters ($\theta = [0.53, 0.08]$; $\sigma_{nKz} = 9.3 * 10^{-7}$; $\sigma_{nCz} = 0.018$)

.

Figure 7: Joint multi-output GPR on K_z and C_z relationship. Solid black line defines the mean function, shaded region defines 95% confidence interval (2σ) distance away from mean.

Figure 7 shows a comparison between two joint multi-output GPR on K_z and C_z using initial assumptions regarding order of differentiability. Figure 7a shows the effect of SE kernel (infinite differentiability) on the joint model. The SE kernel is a strong assumption for type of lift distribution. To accomodate for the value of measured C_z the error band increases and the boundary condition at HTP root is not satisfied. Moreover, we identify points in the figure 7a which fall out of the 95% confidence band. We can claim with 95% probability that if our lift distributions come from a family of highly differentiable functions these outliers do not satisfy the physics of the problem and can be termed as measurement errors.

Figure 7b shows the effect of twice differentiable Matérn kernel on the joint model. The twice differentiable assumption gives our family of functions enough flexibility to follow the spanwise lift data. We observe that error estimates for C_z improves considerably while there is an increased uncertainity at the HTP root for K_z. This loss in order of differentiability can be explained due to fuselage HTP interaction near HTP root. Since there is a discontinuity in the surface due to presence of fuselage, the pressure distribution near the HTP root cannot remain infinitely smooth. We can conclude that even though we are working with very basic assumptions of differentiability the choice of kernel is important in performing the predictions.

This procured model can be used as a robust surrogate model which is consistent with the physics based relations. The relation $g(.)$ can be a known functional relationship or some transformation arising from a computer code. This surrogate model can be used in further use cases for example finding optimal placements of sensors so as to reduce the overall uncertainty in the system, or performing bayesian optimization or Efficient Global Optimization (EGO) using information coming from several outputs. Making the steps more data efficient and consistent with the physics of the system.

IV. Conclusion

This paper describes a general framework of joint multi-output GPR to find mapping function between a set of input and related outputs[9] . Using very basic assumptions such as order of differentiability and

knowledge of relation between the outputs a joint family of functions can be constructed. Later functions inconsistent with the observed data are eliminated using bayes rule. This gives us a prediction which is consistent with the initial assumptions, initial relationship and observed data.

We initially provide empirical demonstrations of the framework on synthetic data related through a quadratic relationship. The approach shows improvements in prediction capabilities and handling of missing data. The second demonstration was performed on flight loads data and demonstrated that imparting prior flight mechanical relationships into GPR allows for a better surrogate flight loads model. Thus, we are now capable of creating a more robust, physically consistent and interpretable surrogate model for these loads, thereby greatly improving the identification process.

As we keep on adding more physical information into GPR, the number of points in the joint-covariance matrix increase. However GPR does not scale well to very large number of input points. Future work will investigate recent advances in scaling up GPR[14] in order to handle several relationships and possibly cover the whole flight test dataset.

References

[1]Allen, M. J. and Dibley, R. P., "Modeling Aircraft Wing Loads from Flight Data Using Neural Networks," *SAE Technical Paper Series*, SAE International, sep 2003.

[2]Reed, S. and Cole, D., "Development of a parametric aircraft fatigue monitoring system using artificial neural networks," *22nd Symp. Int. Committee on Aeronautical Fatigue*, mar 2003.

[3]Ramon Fuentes, Elizabeth Cross, A. H. K. W. R. J. B., "Aircraft Parametric Structural Load Monitoring Using Gaussian Process Regression," *7th European Workshop on Structural Health Monitoring, Jul 2014*, EWSHM, jul 2014.

[4]Rasmussen, C. E. and Williams, C. K. I., *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.

[5]Liu, W., *Development of Gradient-Enhanced Kriging Approximations for Multidisciplinary Design Optimization*, Ph.D. thesis, University of Notre Dame, July 2003.

[6]Solak, E., Murray-Smith, R., Solak, E., Leithead, W., Rasmussen, C., and Leith, D., "Derivative observations in Gaussian Process models of dynamic systems," 2003.

[7]Stein, A. and Corsten, L., "Universal kriging and cokriging as a regression procedure," *Biometrics*, 1991, pp. 575–587.

[8]Boyle, P. and Frean, M., "Dependent Gaussian processes," *In Advances in Neural Information Processing Systems 17*, MIT Press, 2005, pp. 217–224.

[9]Constantinescu, E. M. and Anitescu, M., "Physics-Based Covariance Models for Gaussian Processes with Multiple Outputs," *International Journal for Uncertainty Quantification*, Vol. 3, 2013.

[10]Solak, E., Murray-smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E., "Derivative Observations in Gaussian Process Models of Dynamic Systems," *Advances in Neural Information Processing Systems 15*, edited by S. Becker, S. Thrun, and K. Obermayer, MIT Press, 2003, pp. 1057–1064.

[11]Forrester, A., Sobester, A., and Keane, A., *Engineering design via surrogate modelling: a practical guide*, John Wiley & Sons, 2008.

[12]Duvenaud, D., *Automatic Model Construction with Gaussian Processes*, Ph.D. thesis, Computational and Biological Learning Laboratory, University of Cambridge, 2014.

[13]Stein, M. L., *Interpolation of Spatial Data: Some Theory for Kriging*, Springer, New York, 1999.

[14]Deisenroth, M. P. and Ng, J. W., "Distributed Gaussian Processes," *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015, pp. 1481–1490.