Predicting On-axis Rotorcraft Dynamic Responses Using Machine Learning Techniques

Ryan D Jackson
School of Engineering,
Institute for Risk and Uncertainty,
University of Liverpool, Liverpool, UK, L69 3GQ
Michael Jump
School of Engineering,
University of Liverpool, Liverpool, UK, L69 3GQ

Peter L Green
School of Engineering,
Institute for Risk and Uncertainty,
University of Liverpool, Liverpool, UK, L69 3GQ

Abstract

Physical-law based models are widely utilized in the aerospace industry. One such use is to provide flight dynamics models for use in flight simulators. For human-in-the-loop use, such simulators must run in real-time. Due to the complex physics of rotorcraft flight, to meet this real-time requirement, simplifications to the underlying physics sometimes have to be applied to the model, leading to errors in the model's predictions of the real vehicle's response. This study investigated whether a machine-learning technique could be employed to provide rotorcraft dynamic response predictions. Machine learning was facilitated using a Gaussian Process (GP) non-linear autoregressive model, which predicted the on-axis pitch rate, roll rate, yaw rate and heave responses of a Bo105 rotorcraft. A variational sparse GP model was then developed to reduce the computational cost of implementing the approach on large data sets. It was found that both of the GP models were able to provide accurate on-axis response predictions, particularly when the model input contained all four control inceptors and one lagged on-axis response term. The predictions made showed improvement compared to a corresponding physics-based model. The reduction of training data to one-third (rotational axes) or one-half (heave axis) resulted in only minor degradation of the sparse GP model predictions.

Introduction

Flight simulators form a vital part of any aircraft life cycle. They are used in design and development phases, testing and qualification activities as well as in training and research (Refs. 1 and 2). Due to their availability and relative low cost compared to the corresponding in-service aircraft, the use of simulators has continued to both increase and evolve. The role of flight training simulators, in particular, has moved to zero flight-time qualification in the civilian arena, Ref. 3. This may help to address the increased demand for new pilots, who are needed to replace the current aging population, by providing both initial and recurrent training opportunities. Moreover, the military is increasing the use of simulators for both mission rehearsal and system procurement, Ref. 3.

The fidelity of any flight simulation experience is dependent upon the accuracy of the flight model and the realism of the other integrated components of the simulator, such as the motion actuation and image generation systems. This paper focuses on the first of these elements, the (non-linear) flight dynamics model. Techniques to design and develop such models are well-known and documented Refs. 4 and 5.

To be of any utility, the entire simulation system must run in real-time. This requirement can necessitate simplifications to the model's underlying physics, particularly when complex aircraft such as rotorcraft are the subject of interest. These simplifications mean that the flight model cannot necessarily capture all of the complex dynamics that would be present during the equivalent real scenario, potentially leading to significant differences between the response of the model and of the real aircraft. These differences can, in the worst case, have a negative impact on training for the crew using the simulator Ref. 6.

The quality of the flight dynamics model will clearly influence the training fidelity of the simulation device. The ‘engineering fidelity’ of such a device is typically measured against a series of quantitative requirements contained within simulator qualification documents such as Refs. 7 and 8. It is recognized that examining the response of the simulator in this way only partially serves to characterize its utility. While efforts are underway to develop methods that can better meet this need (Refs. 9, 10, 11, 12 and 13), the current paper seeks to explore techniques whereby the accuracy of the flight dynamics element of the simulation device can be improved, even when the modelled physics can no longer provide an accurate representation of reality in real-time.

Machine learning, the development of models based on patterns and/ or correlations found in data, provides a potential means to enhance flight dynamics models for use in a flight simulator when the physics-based models are unable to predict the appropriate aircraft response. The field of machine learning has expanded rapidly in recent years, as a result of the increased abundance of data in many disciplines. Of course it should be noted that, whilst these powerful methods allow the exploitation of these data, the models derived using such approaches are generally only valid in, or close to, the regions where data is available.

Previous work on the development of rotorcraft flight dynamics models have focused on the use of non-linear autoregressive models (Refs. 14, 15 and 16) but where uncertainties were not taken into account. In the current study the authors develop and validate an autoregressive data-based rotorcraft dynamics model for on-axis control inputs, using machine learning methods that are probabilistic and are therefore able to capture uncertainties in the predicted model outputs. Specifically, the current paper uses Gaussian Processes (GPs) to facilitate machine learning. GPs are probabilistic non-parametric regression methods that have been widely studied in the machine learning community. A useful property of GPs is that, once trained, they can produce very fast emulators of complex models - this is beneficial when dealing with non-linear behaviour such as rotorcraft dynamics. In this paper, GP non-linear autoregressive models with exogenous inputs (so-called ‘NARX models’) are developed. The models can predict the on-axis pitch, roll and yaw rates as well as the heave response of a Bo105 rotorcraft to control inputs, having been trained using corresponding flight-test measurements of the on-axis responses.

One of the known disadvantages of GPs is that they can be computationally expensive if a large training dataset is used. It is, therefore, beneficial to be able to reduce the number of data points used when training a GP algorithm. In this paper, firstly a GP model is created using 32 manually selected training points out of the 800 available; this GP is referred to as the 'full' GP. However, it is unclear if the chosen training points are optimal. The variational sparse Gaussian process approach (Refs. 17 and 18) is then used to automatically create a 'sparse GP' (a GP trained on a subset of the available data) that closely approximates a GP that has been trained on all of the available data. The sparse GP model is compared to the full GP to illustrate that it can produce an accurate model, despite being trained on only a fraction of the available data. Importantly, through this result, it is demonstrated that the proposed method is *scalable* to larger data sets, and therefore should be practical for real-world data available from, for example, flight test campaigns.

In the paper, GP models are developed to predict pitch, roll, yaw and heave rates for a Bo105 rotorcraft. These predictions are compared with both unseen flight test data (i.e., data not used in training) and to physical-law based model predictions from a rotorcraft flight dynamics model implemented using Advanced Rotorcraft Technology's (ART) FLIGHTLAB software. The research presented in this paper builds on the author's previous work (Ref. 19) by developing simulation methods that can accurately capture the complex dynamics of a rotorcraft, while still being able to run in real time.

The paper proceeds as follows. The Numerical Methods Section introduces GPs and Variational sparse Gaussian Processes are introduced next. The Method Section provides a brief background to the test data whilst the Results Section shows the comparison of model response predictions using different approaches described in the paper. The Discussion Section provides commentary on the results achieved. The Conclusions and Future Work Section draw out the conclusions from the work, as well as providing detail on the the proposed next steps for the research.

Numerical Methods

Gaussian Processes

Gaussian Processes (GPs) (Ref. 20, 21) have been widely used in recent years for many different applications, such as engine modelling (Refs. 22, 23, 24). In the current work, GPs are used to perform regression (they can also perform other tasks such as classification (Ref. 20)). An advantageous property of GPs is that they can be used to quantify the uncertainties in their predictions. Specifically, the predictive uncertainties will grow if the GP is applied further away from the training data, thus allowing an assessment to be made for the suitability of the GP model. GP models are sometimes referred to as 'non-parametric' meaning that they do not belong to a certain family of parametric functions*.

In the remainder of this Section, a brief overview of linear regression is given, followed by an introduction to GP regression, focusing on the case where the available training data is corrupted by measurement noise, such as might be present in real sensor data collected during a flight test campaign.

From linear regression to Gaussian Process Regression

Here, a regression model, defined as a linear combination of fixed basis functions, is considered to aid the understanding of a Gaussian process approach. Such a model typically takes the form

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{w}$ is a vector of parameters to be identified and $\boldsymbol{\phi}$ is a vector of 'basis functions'. Adopting a Bayesian approach (Ref. 20) to the identification of $\boldsymbol{w}$, the prior distribution over $\boldsymbol{w}$ can, for example, be chosen to be Gaussian:

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, \gamma^{-1}\boldsymbol{I}) \tag{2}$$

*Linear regression, performed using a second-order polynomial, for example, restricts the resulting model to be one of a family of quadratic equations. Non-parametric approaches do not suffer from such limitations where γ is the precision of the distribution, and $\boldsymbol{I}$ is an identity matrix. Defining N as the number of training points and writing the function values at the points where training data are available as $\boldsymbol{f} = f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_N))^T$ then, from Eq. (1), it can be shown that

$$\boldsymbol{f} = \boldsymbol{\Phi w} \tag{3}$$

where $\boldsymbol{\Phi}$ is referred to as the 'design matrix' (for more information see (Ref. 20)). Given the prior in Eq. (2) it can then be shown that the prior distribution over $\boldsymbol{f}$ is Gaussian, with mean

$$\mathbb{E}[\boldsymbol{f}] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}] = \boldsymbol{0} \tag{4}$$

and covariance matrix

$$\text{cov}[\boldsymbol{f}] = \mathbb{E}[\boldsymbol{f}\boldsymbol{f}^T] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T]\boldsymbol{\Phi}^T = \frac{1}{\gamma}\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{K} \tag{5}$$

where $\boldsymbol{K}$ is a matrix with elements:

$$K_{nm} = \frac{1}{\gamma}\boldsymbol{\phi}(\boldsymbol{x}_n)^T\boldsymbol{\phi}(\boldsymbol{x}_m); \qquad n, m = 1, \ldots, N. \tag{6}$$

At this stage, defining y_n as the nth observation of the modelled system's response, the following 'noise model' is often assumed:

$$y_n = f_n + \epsilon_n, \qquad \epsilon_n \sim \mathcal{N}(\epsilon_n \mid 0, \beta^{-1}) \tag{7}$$

where $f_n \equiv f(x_n)$ and β is the precision of the noise that corrupted each observation. A typical Bayesian approach would then involve inferring the parameters, $\boldsymbol{w}$, implied by the available data. With a GP approach, however, instead of defining basis functions shown in Eq. (1), one can simply define a prior over $\boldsymbol{f}$ directly, such that

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{0}, \boldsymbol{K_{NN}}) \tag{8}$$

where the covariance matrix, $\boldsymbol{K_{NN}}$, is given by

$$\boldsymbol{K_{NN}} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_N, \boldsymbol{x}_1) \\ k(\boldsymbol{x}_1, \boldsymbol{x}_2) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \ldots & k(\boldsymbol{x}_N, \boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ k(\boldsymbol{x}_1, \boldsymbol{x}_N) & k(\boldsymbol{x}_2, \boldsymbol{x}_N) & \ldots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}$$

and where $k(\cdot, \cdot)$ is a user-defined 'kernel function'. The kernel function is chosen so that $\boldsymbol{K_{NN}}$ is a valid covariance matrix (i.e., symmetric and positive-semidefinite). An example of such a kernel function is the squared exponential (Ref. 20):

$$\boldsymbol{K_{NN}}(n, m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \exp\left(-\tfrac{\alpha}{2}(\boldsymbol{x}_n - \boldsymbol{x}_m)^T(\boldsymbol{x}_n - \boldsymbol{x}_m)\right) \tag{9}$$

where the 'hyperparameter', α, induces correlations that depend on the closeness of $\boldsymbol{x}_n$ to $\boldsymbol{x}_m$. The accuracy of the resulting GP model is affected by the choice of hyperparameters. The optimal hyperparameters are most commonly identified by maximising the likelihood of witnessing the observed data:

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y} \mid \boldsymbol{f})p(\boldsymbol{f})d\boldsymbol{f} = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{0}, \boldsymbol{C}) \tag{10}$$

where the elements of $\boldsymbol{C}$ are given by

$$C(n, m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \beta^{-1}I_{nm} \tag{11}$$

and I_{nm} is the (n, m)th element of the identity matrix $\boldsymbol{I}$.

Gaussian Process Predictions

The main aim of GP regression is to make predictions for new model inputs that are not part of the training data. Given a new input vector, $\boldsymbol{x}^*$, the probability of observing a new point, y^*, given previous observations, $\boldsymbol{y}$, can be obtained. Defining $\boldsymbol{y}^* = [y_1, y_2, \ldots, y_N, y^*]^T$, the joint distribution over $\boldsymbol{y}^*$ is

$$p(\boldsymbol{y}^*) = \mathcal{N}(\boldsymbol{y}^* \mid \boldsymbol{0}, \boldsymbol{C_{N+1}}) \tag{12}$$

where $\boldsymbol{C_{N+1}}$ is a $(N+1) \times (N+1)$ covariance matrix:

$$\begin{bmatrix} \boldsymbol{C} & \boldsymbol{k} \\ \boldsymbol{k}^T & c \end{bmatrix} \tag{13}$$

where $\boldsymbol{k}_n = k(\boldsymbol{x}_n, \boldsymbol{x}^*)$ and $c = \beta^{-1} + k(\boldsymbol{x}^*, \boldsymbol{x}^*)$. The mean and variance of y^* given $\boldsymbol{y}$ can be shown to be (Ref. 20):

$$\mu(y^* \mid \boldsymbol{y}) = \boldsymbol{k}^T\boldsymbol{C}^{-1}\boldsymbol{y} \quad \text{and} \quad \sigma^2(y^* \mid \boldsymbol{y}) = c - \boldsymbol{k}^T\boldsymbol{C}^{-1}\boldsymbol{k}$$

such that:

$$p(y^* \mid \boldsymbol{y}) = \mathcal{N}(y^* \mid \mu, \sigma^2) \tag{14}$$

where μ is the mean prediction and σ^2 is the variance (which is used to measure the uncertainty in the predictions of y^*). This can be written more compactly using the notation $y^* \sim \mathbf{GP}(\boldsymbol{x}^*)$.

Choice of Kernel

The kernel function chosen for the current study was used, for example, in Ref. 25. The kernel takes the form:

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \prod_{i=1}^{N_D} \alpha^{(4(x_n^i - x_m^i)^2)} \tag{15}$$

where x^i is used to represent the ith element of the vector $\boldsymbol{x}$ and N_D is the number of dimensions of the input vector. The kernel, Eq. (15), allows the hyperparameter, α, to always be between 0 and 1, providing bounds on the possible values that α can take.

Gaussian Process NARX models

Depending on their input structure, GPs can be used to emulate static or dynamic relationships. The non-linear autoregressive with exogenous inputs (NARX) structure has the format:

$$y_n = f\left(y_{n-1}, y_{n-2}, y_{n-3}, \ldots, v_n, v_{n-1}, v_{n-2}, v_{n-3}, \ldots\right) + \epsilon_n \tag{16}$$

where v is a generic system input and n is a discrete time step. As before, y represents system observations and f is the function that is required to be modelled. The NARX structure uses information from 'lagged' terms (previous observations and inputs) and the current input to help predict y_n.

The model investigated in the current paper has four inputs; longitudinal cyclic stick position (δ^x), lateral cyclic stick position (δ^y), tail rotor pedal position (δ^p) and collective lever position (δ^o). The outputs investigated here correspond to the rotorcraft's four response rate states; pitch (q), roll (p), yaw (r) and heave ($\dot{h}$). Generally speaking, the NARX input structure for the GPs in the following analyses was chosen to be of the form:

$$\boldsymbol{x}_n = \begin{pmatrix} \delta^x_n \\ \delta^y_n \\ \delta^p_n \\ \delta^o_n \\ y_{n-1} \end{pmatrix} \tag{17}$$

where y is the relevant observation (q, p, r or $\dot{h}$). In other words, predictions were made based on the current longitudinal stick position, lateral stick position, pedal position and collective lever as well as one 'lagged term' of the quantity being predicted. This is referred to as the 'all inputs and one lagged term' input structure. A second input structure that was investigated as part of this study used only one input and the relevant 'lagged' output. The input used in this case is the on-axis response such that, for example, when attempting to predict pitch rate, the input structure would take the form:

$$\boldsymbol{x}_n = \begin{pmatrix} \delta^x_n \\ y_{n-1} \end{pmatrix} \tag{18}$$

In the following, the inputs and outputs were normalized to ensure that all values were between 0 and 1, thus guaranteeing that each were on the same scale. The models generated were used to produce what is known as ‘one step ahead predictions’ and ‘full model predictions’. These two types of predictions are discussed, in detail, in the following two Sections.

One Step ahead predictions

One step ahead predictions (OSAP) use the previously observed data to predict a single step into the future. In this specific case, a OSAP is defined as:

$$y_n^* = \mathbf{GP}\left(\delta_n^x, \delta_n^y, \delta_n^p, \delta_n^o, y_{n-1}\right) \tag{19}$$

where y_{n-1} is the previous observation of the relevant quantity (i.e., pitch, roll, yaw or heave). During the training of the GP, OSAP are used to quantify the fidelity of the emulator. The obvious disadvantage of this approach is that the resulting model can only predict a single step into the future. Predicting further into the future requires ‘full model predictions’.

Full model predictions

To make predictions beyond a single step, the GP-NARX framework must utilize previous predictions as part of the model input. To illustrate this, the situation where a single prediction was used, at time n, has already been made according to Eq. (19), is considered. This prediction is written as y_n^*. Following this, predictions of y_{n+1}^* would be realized according to

$$y_{n+1}^* = \mathbf{GP}\left(\delta_{n+1}^x, \delta_{n+1}^y, \delta_{n+1}^p, \delta_{n+1}^o, y_n^*\right) \tag{20}$$

(where, for this example, the input structure is shown in Eq. (18) was used).

The key aspect to note with regard to Eq. (20) is that y_n^* - the uncertain prediction made by the GP at time n - is now part of the model input. A Monte Carlo analysis can be used to address the additional uncertainty that is introduced by including y_n^* as a model input. By definition, y_n^* is a Gaussian random variable and, as such, samples of y_n^* can be generated easily. A simple algorithm for generating an ensemble of predictions for full model predictions (FMP) is shown in Algorithm 1 where R is the number of Monte Carlo samples and Y denotes a random sample. For a more detailed algorithm for FMP, Ref. 26 is recommended.

Algorithm 1 Full Model Predictions algorithm

```
for $r = 1 : R$ do
    $Y^{*}_{(n),r} \sim \mathbf{GP}(y_{(n-1)}, \delta_n)$
    $Y^{*}_{(n+1),r} \sim \mathbf{GP}(Y^{*}_{(n),r}, \delta_{n+1})$
end for
```

Optimization of hyperparameters for Gaussian Process training

GP training requires hyperparameter optimisation. Defining $\boldsymbol{\theta}$ as a vector of GP hyperparameters ($\boldsymbol{\theta} = [\alpha, \beta]^T$ is this case), the posterior parameter distribution is given by Bayes' rule:

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}). \tag{21}$$

Markov Chain Monte Carlo (MCMC) can be used to generate samples from the posterior hyperparameter distribution, Eq. (21), thus allowing the incorporation of hyperparameter uncertainty into the GP predictions. It should be noted that, in Ref. 19, the authors applied the the MCMC algorithm described in (Ref. 27) to the current problem and concluded that hyperparameter uncertainty had little effect on the outcome of the model. As a result, in the following, standard gradient-based optimisation techniques were used to estimate the optimal GP hyperparameters.

Variational Sparse Gaussian Process

Motivation for Sparse Gaussian Processes

The selection of the 'correct' amount of training data is crucial for the creation of an efficient and accurate model. The use of too much data will produce a model that will take longer than necessary to optimize and make predictions. Too little data will have the opposite effect; the optimization process will be quicker, but the accuracy of the model will be compromised.

The application of GP models is intractable for large data sets, as the complexity associated with training scales with $O(N^3)$, where N is the number of training points (Ref. 17). A reduction in the required number of training points is therefore beneficial. In the current work, the authors investigated the application of variational sparse GPs (a class of GP that is better suited to larger data sets) as, to be implemented practically in a flight simulator, it is likely that the method would be applied to much larger data sets than those described in the current paper. Generally speaking, a GP model that has been created with a subset of the original training data is referred to as a sparse GP. The aim is to make the sparse GP model create predictions that are as close as possible to the predictions made by the GP model trained using all of the available training data (hereafter referred to as the 'full' GP).

The training inputs used for a sparse GP are often referred to as 'inducing points' in the literature. There have been methods proposed to select such inducing points automatically. One approach, by Snelson and Ghahramani (Ref. 28), uses inducing points that do not have to come from the initial training set (referred to as 'pseudo-inputs'). For the work reported in this paper, the inducing points are simply chosen to be a subset of the available training inputs. In the following, for the sake of readability, the inputs used to train a GP model are referred to simply as the 'training points'.

Variational Lower bound

This Section provides a brief overview of the variational sparse GP approach, proposed by Titsias in Refs. 17 and 18, that is employed in the current work. For the interested reader, the authors have provided tutorial material and example Python code for variational sparse GPs at the following Github repository: `https://github.com/plgreenLIRU/Gaussian-Process-SparseGP-Tutorial`.

In the current paper, the variational sparse GP selects a subset of m training points, $\boldsymbol{X}_M$, from the available input data such that:

$$\boldsymbol{X}_M \subset \boldsymbol{X}, \qquad \text{where} \qquad \boldsymbol{X}_M = \{\boldsymbol{x}_{m,1}, \ldots, \boldsymbol{x}_{m,M}\}. \tag{22}$$

The 'optimal choice' of training points must first be established. Defining:

$$\boldsymbol{f}_M = \{f(\boldsymbol{x}_{m,1}), ..., f(\boldsymbol{x}_{m,M})\} \tag{23}$$

then, through Bayes' theorem, it can be shown that:

$$p(\boldsymbol{f}, \boldsymbol{f}_M \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_M)p(\boldsymbol{f}, \boldsymbol{f}_M)}{p(\boldsymbol{y})}. \tag{24}$$

In (Refs. 17 and 18) it is shown that the optimal choice of $\boldsymbol{f}_M$ leads to:

$$p(\boldsymbol{y} \mid \boldsymbol{f}_M) = p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_M) \tag{25}$$

such that, if $\boldsymbol{f}_M$ is already known, then knowing $\boldsymbol{f}$ does not add additional information with regards to the probability of witnessing $\boldsymbol{y}$. When the condition shown in Eq. (25) is satisfied it can be shown that the posterior distribution factorizes as:

$$p(\boldsymbol{f}, \boldsymbol{f}_M \mid \boldsymbol{y}) = p(\boldsymbol{f}_M \mid \boldsymbol{y})p(\boldsymbol{f} \mid \boldsymbol{f}_M). \tag{26}$$

A Variational Bayesian approach (Ref. 20) can then be used to approximate $p(\boldsymbol{f}, \boldsymbol{f}_M \mid \boldsymbol{y})$ with the probability density function (PDF):

$$q(\boldsymbol{f}, \boldsymbol{f}_M) = \psi(\boldsymbol{f}_M)p(\boldsymbol{f} \mid \boldsymbol{f}_M) \tag{27}$$

where $\psi(\boldsymbol{f}_M)$ is a PDF that is known as the 'variational distribution'. The aim is to minimize the Kullback Lieber (KL) divergence between $p(\boldsymbol{f}, \boldsymbol{f_M} \mid \boldsymbol{y})$ and $q(\boldsymbol{f}, \boldsymbol{f}_M)$:

$$\text{KL}\left(q(\boldsymbol{f}, \boldsymbol{f}_m) \mid\mid p(\boldsymbol{f}, \boldsymbol{f}_m \mid \boldsymbol{y})\right). \tag{28}$$

Note that minimization of the KL divergence implies that the posterior can be factorized as shown in Eq. (26). The chosen subset, $\boldsymbol{X}_M$, that minimizes Eq. (28) will, therefore, be the optimal choice of training inputs for the sparse GP.

Following a standard variational Bayesian approach, it is known that minimizing the KL divergence is equivalent to maximizing $\mathcal{L}$:

$$\mathcal{L} = \int \int q(\boldsymbol{f}, \boldsymbol{f}_m) \log \left[\frac{p(\boldsymbol{f}, \boldsymbol{f}_m, \boldsymbol{y})}{q(\boldsymbol{f}, \boldsymbol{f}_m)} \right] d\boldsymbol{f} d\boldsymbol{f}_m. \tag{29}$$

A lower bound on $\mathcal{L}$, denoted F_v, can be found through Jensen's inequality (Refs. 17 and 18):

$$F_v(\boldsymbol{X}_M) = -\frac{1}{2} \left(N \log(2\pi) + \log \left(\left| \boldsymbol{I}\beta^{-1} + \boldsymbol{Q} \right| \right) + \boldsymbol{X}^T \boldsymbol{X} \left(\boldsymbol{I}\beta^{-1} + \boldsymbol{Q} \right)^{-1} - \beta \mathrm{Tr} \left(\boldsymbol{K}_{NN} - \boldsymbol{Q} \right) \right) \tag{30}$$

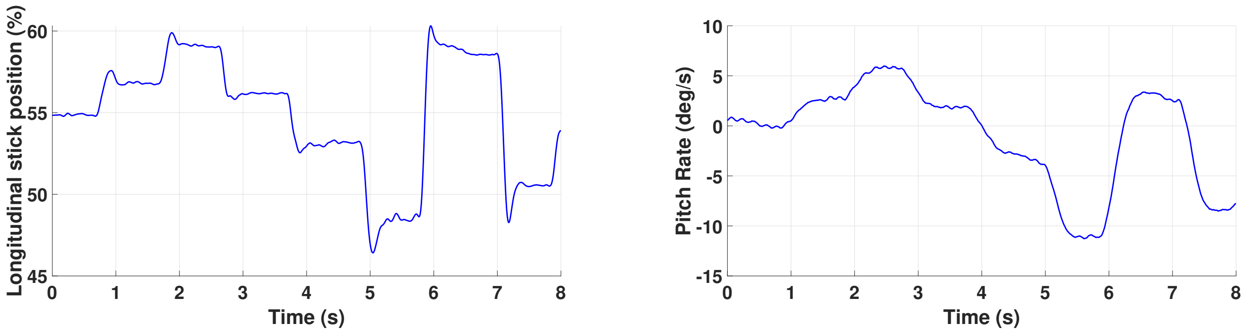
where:

$$\boldsymbol{Q} = \boldsymbol{K}_{NM} \boldsymbol{K}_{MM}^{-1} \boldsymbol{K}_{MN}$$

$$\boldsymbol{K}_{NM}(i,j) = k(\boldsymbol{x}_i, \boldsymbol{x}_{m,j}), \quad \boldsymbol{K}_{MM}(i,j) = k(\boldsymbol{x}_{m,i}, \boldsymbol{x}_{m,j})$$

This approach is effective due to the reduced computational cost associated with evaluating F_v. As equation (30) involves just the trace of $\boldsymbol{K}_{NN}$, only the diagonal elements of $\boldsymbol{K}_{NN}$ are required. The matrices in Eq. (30) that require inversion are of size $M \times M$, thus reducing the computational cost to $O(M^3)$ compared to $O(N^3)$ for a standard GP (as $M < N$). The selection of the training points and the optimization of the hyperparameters is conducted in an Expectation-Maximization 'like' algorithm, applied to $F_v(\boldsymbol{X}_M)$, as described in (Refs. 17 and 18).

To begin training a variational sparse GP model, a small number of initial training data points must be chosen. The K-means clustering algorithm (Ref. 20) was used for this purpose in the present work. Such an approach ensures that the initial points are chosen from clusters that already exist in the data, providing a sensible initial estimate for $\boldsymbol{X}_M$. For the sparse results presented in this paper, the K-means clustering algorithm identified five clusters, allowing the variational sparse GP to be initiated using five points (one from each cluster).



(a) Longitudinal stick position input of the Bo105 rotorcraft.

(b) Pitch rate response from the Bo105 rotorcraft.

Fig. 1: Example of the Bo105 flight test data

Method

GP rotorcraft dynamics models were trained using Bo105 flight test datasets. Two different approaches to select the training points were used. The 'full GP' approach involved choosing 32 training points at equal time intervals. The second approach used the variational sparse GP method, where the training points were automatically selected. The response of these models was then obtained and compared with the corresponding flight test response, as well as the output from a physics-based multi-body dynamics model.

Flight Test Data

The flight test database used to train the GP models for this study is the so-called 'AGARD database', which was used by the AGARD Working Group FMP WG 18, on 'Rotorcraft System Identification'. The dataset was delivered to the University of Liverpool as part of the GARTEUR HC(AG-16) Rotorcraft Pilot Couplings (Ref. 36). An example of the flight test data from the Bo105 rotorcraft is shown in Fig. 1, where Fig. 1a is the longitudinal stick position and Fig. 1b is the corresponding pitch rate response.

Table 1 shows the 4 datasets used to train the GP models, one for each primary axis of interest. In each case, the 3-2-1-1 maneuver was used. This was the most complex maneuver available and hence, provided the most information-rich set of dynamic responses for GP training.

All datasets used for results				
Data Set	**Response**	**Inputs structure**	**Prediction**	**Case**
3-2-1-1 Longitudinal Input	Pitch Rate	Longitudinal Input and one lagged Pitch Rate	OSAP	1
			FMP	2
		All Inputs and one lagged Pitch Rate	OSAP	3
			FMP	4
3-2-1-1 Lateral Input	Roll Rate	Lateral Input and one lagged Roll Rate	OSAP	5
			FMP	6
		All Inputs and one lagged Roll Rate	OSAP	7
			FMP	8
3-2-1-1 Pedal Input	Yaw Rate	Pedal Input and one lagged Yaw Rate	OSAP	9
			FMP	10
		All Inputs and one lagged Yaw Rate	OSAP	11
			FMP	12
3-2-1-1 Collective Input	Heave	Collective Input and one lagged Heave	OSAP	13
			FMP	14
		All Inputs and one lagged Heave	OSAP	15
			FMP	16

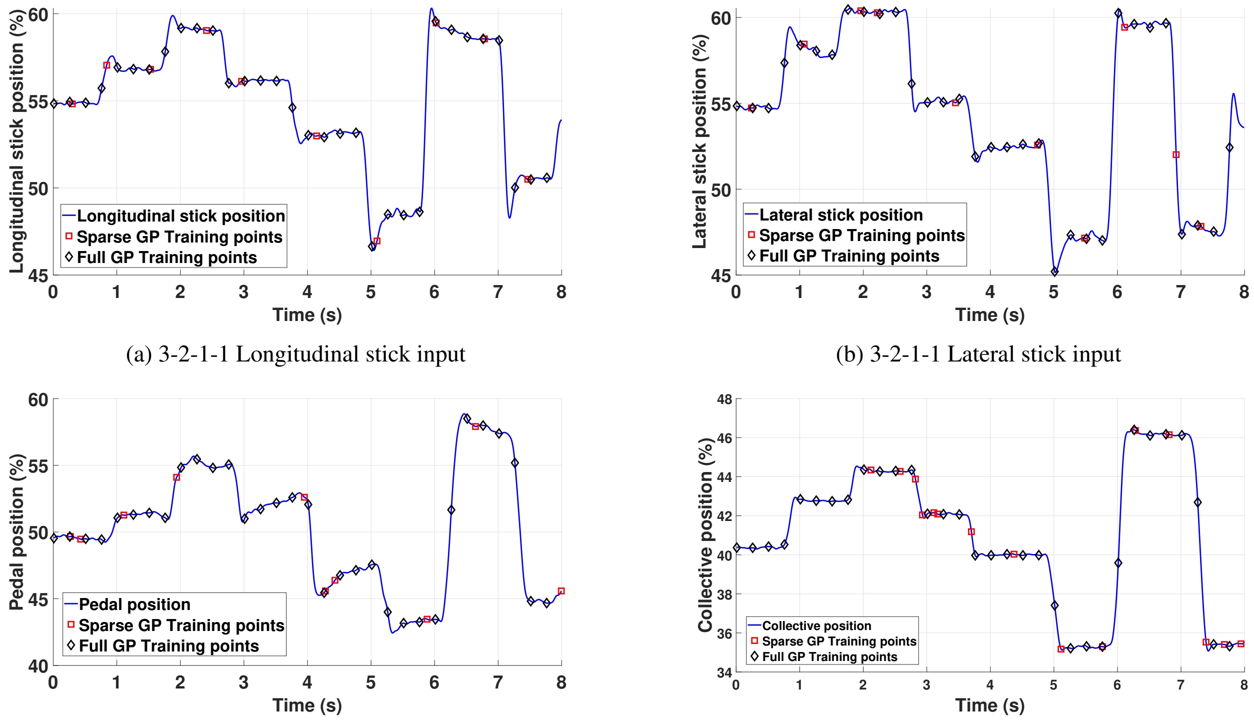
Table 1: Data sets and input structures used in the Section entitled Results

Fig. 2 shows the primary inceptor time-history for each maneuver in Table 1. Also shown are the training data points used in each case to train the full GP model and the sparse GP model. The full GP model training data points are displayed as black diamonds. The sparse GP training data points are shown as squares; these data points are selected automatically by the variational sparse GP approach.

For each axis, rotorcraft response predictions were made using 2 different forms of the GP model input structure. The first input structure contained the primary input for the corresponding response, as well as a lagged response. For example, to predict the nth pitch rate, the input structure ‘Longitudinal Input and one lagged Pitch Rate’ refers specifically to inputs of the form shown in equation (18), where y_{n-1} is the lagged response (i.e., q at time $n-1$). The second input structure also contained the non-primary inputs. For example, to predict the nth pitch rate, the input structure ‘All Inputs and one lagged Pitch Rate’ refers specifically to inputs of the form shown in equation (17).

Bo105 rotorcraft

The Bo105 is a twin-engined rotorcraft, in the 2.5 ton-class, which has fulfilled a number of roles in transport, offshore, police and military missions. The Bo105 database used to generate the model in this



(a) 3-2-1-1 Longitudinal stick input

(b) 3-2-1-1 Lateral stick input

(c) 3-2-1-1 Pedal stick input

(d) 3-2-1-1 Collective input

Fig. 2: Time history of each 3-2-1-1 maneuver

study was first extracted from Ref. 29 which itself incorporates open-source data collected from GARTEUR HC(AG06) activities in Ref. 30. The flight control system data were first taken from Ref. 31. The more complete database that was provided to the GARTEUR Action Group HC(AG16) (Ref. 32) was then utilized.

FLIGHTLAB

The physics-based rotorcraft model was developed using FLIGHTLAB (Ref. **?**). A summary of FLIGHTLAB's capabilities are provided in Ref. 33. Note, the FLIGHTLAB model shown in the current work was previously configured by an 'expert' user.

Model Response Prediction Comparison Method

To objectively compare the performance of the FLIGHTLAB and GP rotorcraft model response predictions, the root-mean-square (RMS) error of the model response at a given time compared to the corresponding flight test data response value was used. This metric is calculated using

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i^* - y_i)} \tag{31}$$

where y^* is the mean prediction and y are the observed values.

Results

This Section reports upon the comparison between the FLIGHTLAB and GP model response predictions and the corresponding flight test datasets. Response predictions are shown for the 3-2-1-1 maneuver for each axis of interest both for the full GP model and then for the variational sparse GP model for each of the cases shown in Table 1. Note that the RMSE errors associated with the following analyses are reported in Table 2.

Gaussian Process (Full GP)

In this Section, the 4-axis rotational on-axis responses of the FLIGHTLAB Bo105 model to a longitudinal, lateral, pedal and collective 3-2-1-1 cyclic input is compared with the GP model's output as well as measurements from the equivalent Bo105 flight test data. The flight test control input from the Bo105 rotorcraft was also applied to the physical-law based FLIGHTLAB model.

Longitudinal Stick Input to Pitch Rate Prediction

Fig. 3 shows the OSAP prediction made by the full GP predicting the pitch rate, using the current longitudinal stick position and one lagged pitch rate input structure (Case 1, Table 1). This and subsequent OSAP-related figures show:

1) The original flight test data;

2) The GP OSAP scheme rotorcraft model response predictions and

3) The GP model-calculated confidence bounds based upon 3 standard deviations from the mean.

As discussed previously, for the OSAP scheme, it would be expected that the predicted response is very close to the Bo105 flight test data. This is indeed apparent in these results: first, the RMS error between

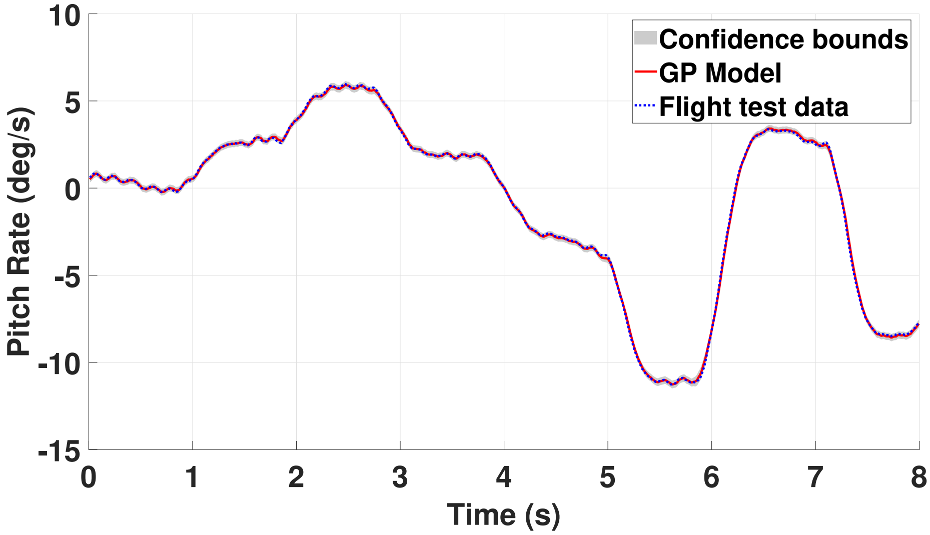


Fig. 3: One step ahead predictions of the pitch rate 3-2-1-1 response for the input structure containing the longitudinal input and one lagged pitch rate (Case 1 in Table 1).

the GP model and the flight test data is 0.074; second, the confidence bounds are difficult to see, indicating the model's confidence in the response predictions made. For the sake of brevity, in the remainder of the paper, OSAP predictions are not shown as they were always found to lie close to the real flight data (and, furthermore, are of little value for flight simulation purposes, where full model predictions are required).

Fig. 4 shows the pitch rate response for: the original flight test data (blue line), the GP FMP response predictions using the current longitudinal stick position and one lagged pitch rate input structure (Case 2, Table 1, red lines), the GP FMP responses using all of the current inceptor positions and one lagged pitch rate input structure (Case 4, Table 1, green lines) and the FLIGHTLAB model response, again, using all of the corresponding inceptor inputs for that flight test point (dashed black line).

The FMP should be a more challenging test for the GP model, as it tests its ability to make predictions more than one step into the future. As shown in Algorithm 1, a Monte Carlo analysis is required to enable FMP predictions. Fig. 4 shows the ensemble of predictions made by a Monte Carlo analysis that used 10,000 runs. The Monte Carlo results are plotted using semi-transparent lines, such that darker regions represent areas where more model response realizations were observed.

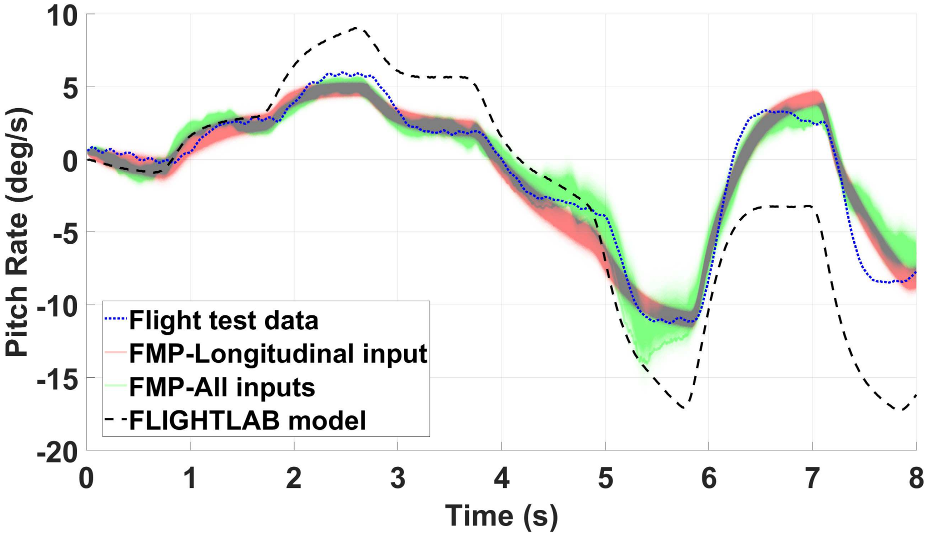


Fig. 4: Realization of the ‘full’ GP pitch rate prediction for input structure containing the Longitudinal input with one lagged pitch rate (Case 2 in Table 1) and ALL inputs with one lagged pitch rate (Case 4 in Table 1) with the comparison of the FLIGHTLAB model.

For ease of comparison, Fig. 4 shows pitch rate realizations from the FMP both for when the GP used a single inceptor input (in this case, longitudinal) and a lagged pitch rate prediction as the input structure (Case 2, Table 1) and for when all of the inputs (longitudinal, lateral, collective and pedal) and a lagged pitch rate prediction as the input structure (Case 4, Table 1) were used. It can be seen that the Case 2 response captures the essence of the rotorcraft response but that Case 4 provides a slightly improved prediction over the middle portion of the response. Both Cases become less well able to predict the rotorcraft response towards the very latter stages of the maneuver. That said, it is apparent that the GP models provide improved response predictions when compared to the FLIGHTLAB model for much of the response time-history. In general, the GP predictions remain much closer to the flight test truth data throughout the maneuver, but particularly in its latter stages. It is interesting to note that the ensemble GP response predictions in Fig. 4 do not always encompass the flight test data. Potential reasons for this are discussed later in the paper.

Lateral Stick Input to Roll Rate Prediction

Fig. 5 shows the lateral response comparison between the flight test data, the GP full model predictions and FLIGHTLAB. It is immediately apparent that the FMP using only a single inceptor as input deviates significantly from the flight test data between approximately 3 to 6 seconds of the maneuver. The

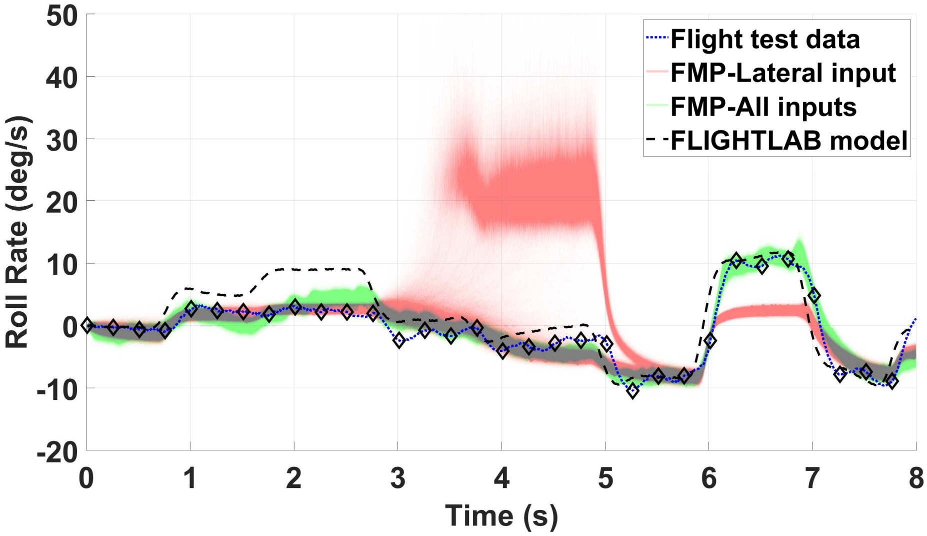


Fig. 5: Realization of the ‘full’ GP roll rate prediction for input structure containing the Lateral input with one lagged roll rate (Case 6 in Table 1) and ALL inputs with one lagged roll rate (Case 8 in Table 1) with the comparison of the FLIGHTLAB model.

response predictions appear to bifurcate into two relatively distinct prediction regions. This type of result can arise when an incomplete input structure is utilized as, in this relatively low-dimensional input space, the ‘closeness’ of new inputs to the training data may be erroneously estimated compared to if a higher dimensional input space had been used. In Fig. 5 the GP model predictions using all of the inceptor inputs are much improved compared to the single inceptor input case. This indicates that, in the case of the roll rate response, all of the inputs are required to closely capture the system dynamics. This was not seen in the longitudinal results, shown above.

As with the pitch rate prediction, all of the models capture the essence of the real aircraft response. Once again, the FLIGHTLAB model provides less accurate response predictions compared to the GP models and, specifically, the all inceptor input and lagged roll rate input structure model.

Rudder Pedal Input to Yaw Rate Prediction

The GP full model predictions for the yaw rate, using the current pedal input and one lagged yaw rate input structure (Case 9, Table 1), are shown in Fig. 6. Again, results are compared with the FLIGHTLAB model and the flight test data. All of the model predictions appear less accurate for this axis than for the

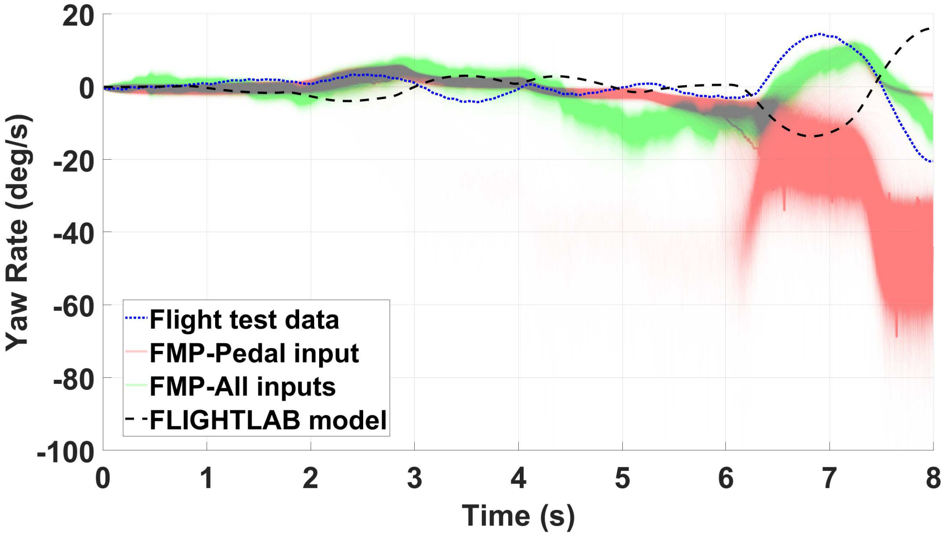


Fig. 6: Realisation of the ‘full’ GP yaw rate prediction for input structure containing the Pedal input with one lagged yaw rate (Case 10 in Table 1) and ALL inputs with one lagged yaw rate (Case 12 in Table 1) with the comparison of the FLIGHTLAB model.

previous two axes. The FLIGHTLAB model has predictions that are of a similar magnitude but in the opposite sense to the flight test ‘truth’ data, whilst the GP models appear to make closer predictions to reality for the early part of the maneuver. The FLIGHTLAB and single inceptor input structure GP model provide response predictions closest to the flight test data over the mid-part of the maneuver, but then the multiple inceptor input structure GP model predicts closest to the flight test data, in both magnitude and direction towards the latter end of the maneuver.

Collective Input to Heave Response Prediction

For the input structure containing the current collective lever and one lagged heave term (Case 13, Table 1), Fig. 7 shows the heave response comparison between the FLIGHTLAB and GP models and the flight test data. It is clear that the single inceptor input GP model and the FLIGHTLAB model struggle to accurately capture the dynamics. The GP model, again in this case, shows a bifurcation in its predictions. For the GP model that uses an input structure that contains all of the control inceptors, the response prediction becomes a much closer representation of the dynamics of the system.

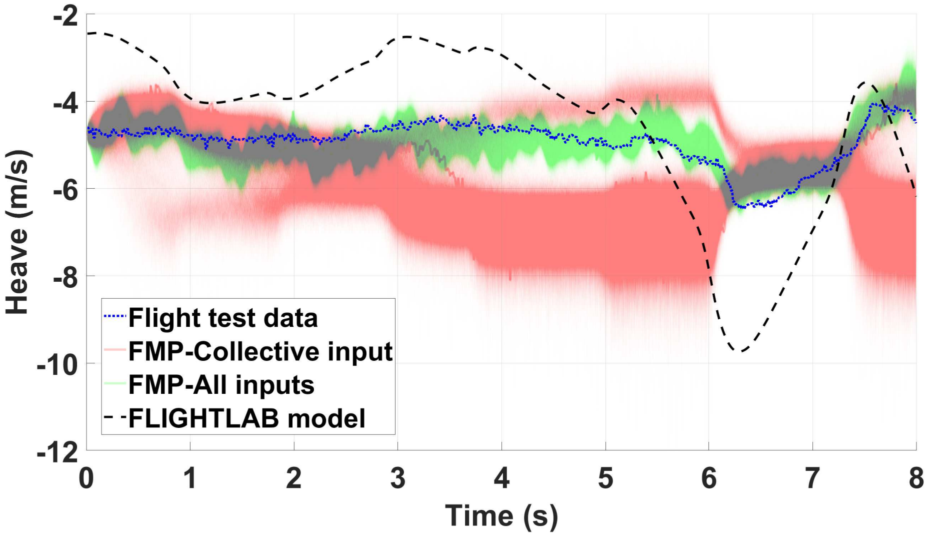


Fig. 7: Realization of the ‘full’ GP heave prediction for input structure containing the Collective input with one lagged heave (Case 14 in Table 1) and ALL inputs with one lagged heave (Case 16 in Table 1) with the comparison of the FLIGHTLAB model.

Variational Sparse Gaussian Process

In the previous Section, Full GP model predictions were used based upon training data using 32 samples points, as shown in Fig. 2. However, to investigate any effect on the accuracy of the achievable results via the use of fewer data points, the variational sparse method described in Section (Variational Lower bound) was used to select training points from the initial training set. In the current Section, for the sake of brevity, the OSAP for the variational sparse GPs will not be shown. Furthermore, only the results for the all inputs and a relevant lagged output input structure, Eq. (17) will be shown. The comparisons of Fig. 5, Fig. 6 and Fig. 7 indicate that all of the inputs are required to closely predict the roll, yaw and heave responses (see Table 2 for RMSE values).

Longitudinal Stick Input to Pitch Rate Prediction

For the primary longitudinal response axis, a variational sparse GP approach was applied to Case 4 in Table 1. Fig. 8 shows the convergence of the lower bound for 32 training points after the initial five training points were chosen. The selection of the number of training points for the variational sparse GP was based on the judgment of convergence of the lower bound. By inspection, Fig. 8 indicates that the

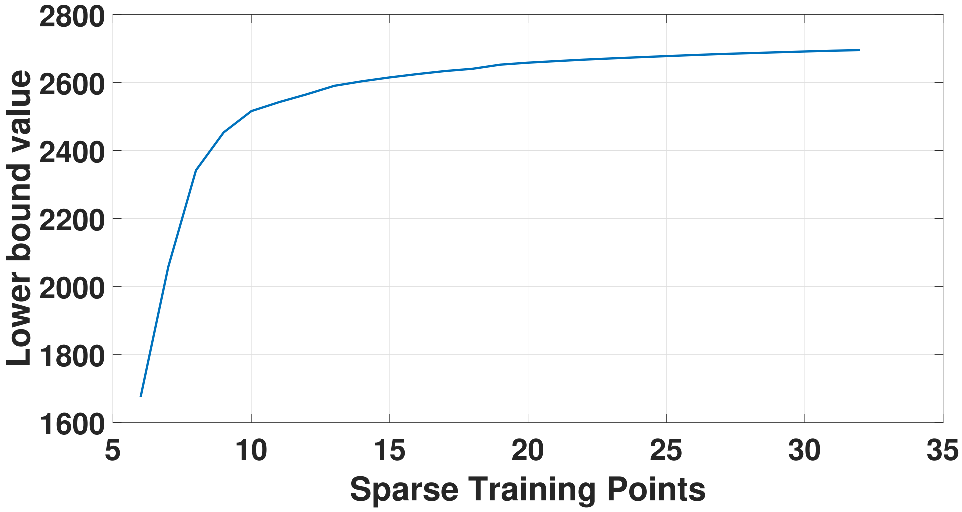


Fig. 8: Convergence of the lower bound for the sparse GP pitch rate prediction using 32 training points.

variational sparse GP could potentially use only 10 training points to create a model that predicts similar results to the 'full' GP. To avoid repetition, in the Sections (lateral, directional and heave axes), the convergence plots of the lower bound are not shown (all cases demonstrated behavior similar to that shown in Fig. 8).

The sparse GP FMP predictions for pitch rate during the 3-2-1-1 maneuver, using only 10 training points, are shown in Fig. 9. Also shown on this plot are: the equivalent predictions made by the 'full' GP model (using 32 training points), the equivalent FLIGHTLAB model predictions and the corresponding flight test results. All of the models capture the essence of the response, with the FLIGHTLAB model diverging from reality earliest in the maneuver. It is of interest that, for a reduction of just over 1/3 of the utilized training data points, the sparse GP model predictions are very close to those of the full GP model for the majority of the maneuver, but with slightly larger uncertainty, as might be expected.

Lateral Stick to Roll Rate and Rudder Pedal to Yaw Rate Predictions

Lateral and directional axis response predictions were made for Cases 4 and 6 in Table 1, again, using only 10 training data points in both cases. The results of these predictions are shown in Figs. 10 and 11. As with the longitudinal axis, the sparse GP models provide response predictions that are close to their full GP model counterparts. Again, the responses are an improvement over the FLIGHTLAB model used

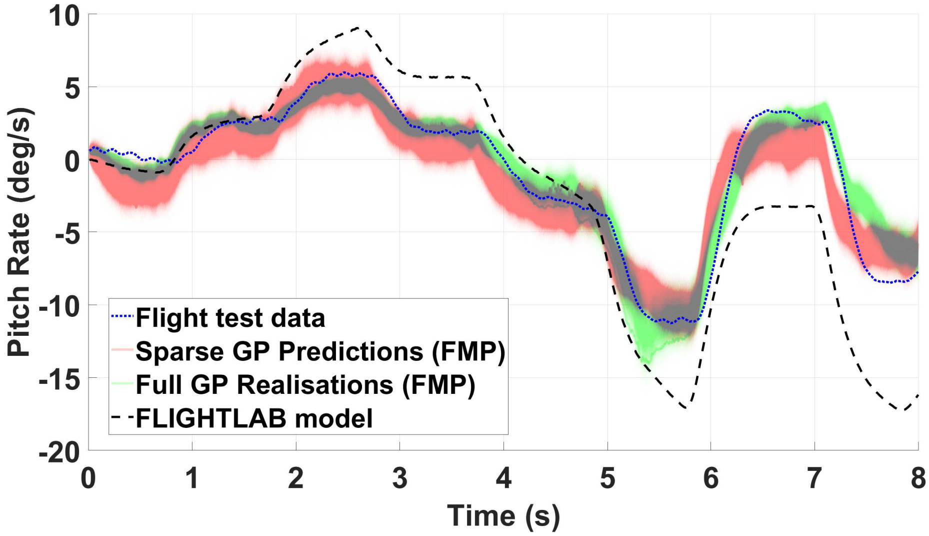


Fig. 9: Comparison of the variational sparse GP pitch rate prediction for the input structure containing ALL inputs and one lagged pitch rate using 10 training points with the Full GP using the same input structure (Case 4 in Table 1) and the FLIGHTLAB model.

for this study. For the lateral case, by inspection, the sparse model representation provides results that match the flight test data less closely and this is reinforced by the RMS data in Table 2. The same is true for the directional results but, for localised regions within the maneuver time history, it is apparent that the sparse GP model provides response values closer to the flight test data than does the full GP model. This could be due to the training points selected by the sparse GP approach containing more information than the training points selected at evenly-spaced time intervals.

Collective Lever to Heave Response Predictions

For the sparse GP model response prediction case, 15 training points had to be used; this was necessary as the convergence of the lower bound took longer than the previous cases. With this slight modification to the analysis procedure, Fig. 12 shows the model prediction comparison for Case 16 in Table 1. As before, the sparse GP model creates predictions that are close to, but not quite as accurate as those made by the full GP model. Both GP models provide a response prediction that is closer to the flight test data than the FLIGHTLAB model used for this study.

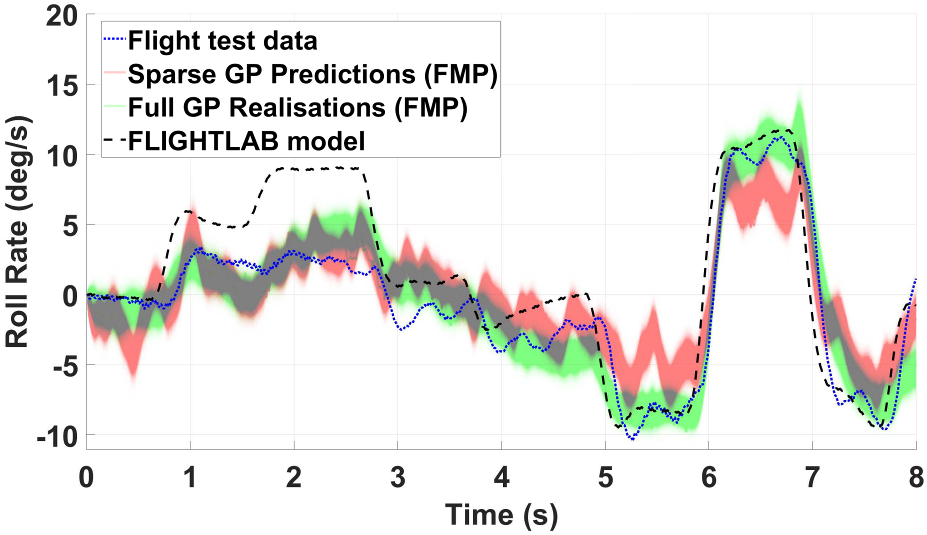


Fig. 10: Comparison of the variational sparse GP roll rate prediction for the input structure containing ALL inputs and one lagged roll rate using 10 training points with the Full GP using the same input structure (Case 8 in Table 1) and the FLIGHTLAB model.

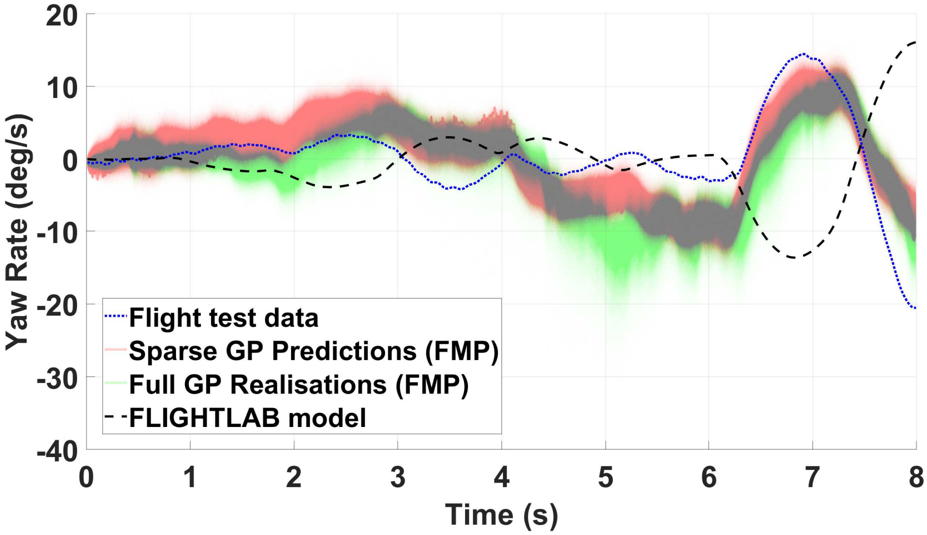


Fig. 11: Comparison of the variational sparse GP yaw rate prediction for the input structure containing ALL inputs and one lagged yaw rate using 10 training points with the Full GP using the same input structure (Case 12 in Table 1) and the FLIGHTLAB model.

Root Mean Squared Error values for all models					
Response	**Model**	**GP Input**	**Prediction**	**Case**	**RMSE Value**
Pitch Rate	‘Full’ Gaussian Process (32 Points)	Long	OSAP	1	0.07
			FMP	2	1.12
		ALL	OSAP	3	0.07
			FMP	4	1.16
	Variational Sparse Gaussian Process (10 Points)	ALL	OSAP	3	0.13
			FMP	4	1.57
	FLIGHTLAB	—	—	—	4.24
Roll Rate	‘Full’ Gaussian Process (32 Points)	Lat	OSAP	5	0.20
			FMP	6	3.97
		ALL	OSAP	7	0.19
			FMP	8	1.66
	Variational Sparse Gaussian Process (10 Points)	ALL	OSAP	7	0.27
			FMP	8	2.38
	FLIGHTLAB	—	—	—	3.37
Yaw Rate	‘Full’ Gaussian Process (32 Points)	Ped	OSAP	9	0.16
			FMP	10	12.08
		ALL	OSAP	11	0.13
			FMP	12	5.69
	Variational Sparse Gaussian Process (10 Points)	ALL	OSAP	11	0.20
			FMP	12	4.55
	FLIGHTLAB	—	—	—	10.73
Heave	‘Full’ Gaussian Process (32 Points)	Col	OSAP	13	0.06
			FMP	14	1.16
		ALL	OSAP	15	0.07
			FMP	16	0.30
	Variational Sparse Gaussian Process (10 Points)	ALL	OSAP	15	0.06
			FMP	16	0.29
	FLIGHTLAB	—	—	—	1.59

Table 2: Root Mean Squared Error for the four models Gaussian Process, FLIGHTLAB and Variational Sparse Gaussian Process approach for the predictions of the pitch rate, roll rate, yaw rate and heave.

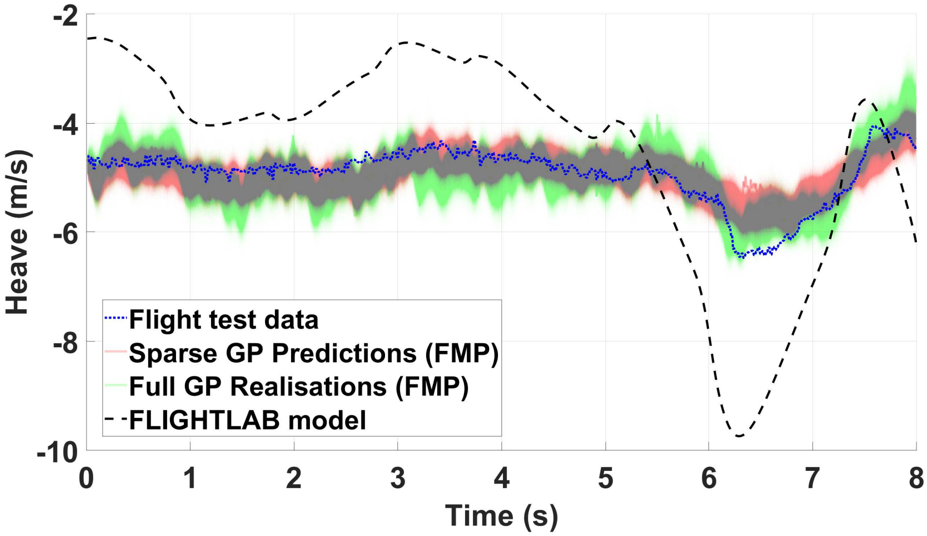


Fig. 12: Comparison of the variational sparse GP heave prediction for the input structure containing ALL inputs and one lagged heave using 15 training points with the Full GP using the same input structure (Case 16 in Table 1) and the FLIGHTLAB model.

Discussion

To be of any utility for simulation purposes, any machine-learnt flight dynamics model must be able to make predictions in real time. For the sparse GP models that used 10 training points, a full set of realizations (i.e., 8 seconds of predictions) were made, using a standard desktop PC, in between 0.03-0.045 seconds. Likewise, for the sparse GP models that used 15 training points, response predictions were made in between 0.035 and 0.05 seconds. This strongly suggests the models described in the current work can be used for real-time simulation.

One important feature of rotorcraft dynamic behaviour is that individual axis inputs give rise to coupled axis output responses. In order to obtain further improvements in the GP model predictions, an obvious next step would be to create a multi-input-multi-output (MIMO) GP model, which would then include the cross-coupling response effects. The missing coupling dynamics could well be at least part of the reason as to why the Full GP model predictions do not encompass all of the real flight test data and may also facilitate predictions of off-axis responses. The authors aim to investigate this aspect as part of the project's future work. Another avenue of investigation to further improve prediction accuracy is to use a mixture of kernels to capture response predictions at different frequencies.

In all cases, the GP model responses were compared not only with the flight test data to assess the prediction accuracy but also with a FLIGHTLAB physics-based model available to the authors. Whilst the FLIGHTLAB model was generally able to give good prediction accuracy over the initial part of the maneuvers tested; the GP models were able to provide more accurate response predictions over the majority of the maneuver period. In that sense, the GP models outperformed the FLIGHTLAB model in this study. The authors do not claim that this is a general result. The FLIGHTLAB model used was not intended to be a high-fidelity representation of a Bo105 aircraft but was meant to be 'representative' of it. It is recognized that higher fidelity models can be created given sufficient time and resource, such as those reported in Refs. 34 and 35.

Even when applied successfully, machine learning techniques should never be used to make extrapolations beyond regions where training data is available. Clearly then, such models must be trained on datasets that cover the normal operational environment before they can applied in a flight simulation environment. Gathering data to validate model responses outside of the normal operating envelope is an obvious problem here but this is equally true for existing techniques. The advantage of the proposed technique is that it is able to compute and inform as to when the model response is uncertain. It is also worth noting that machine learning algorithms are very much 'black-box' techniques that can be difficult or impossible to interpret in a physical sense. However, when physical-law-based models are considered, it is often the case that measurement data for all of the internal physical variables are not be available (thus forcing a reduction in the accuracy and applicability of the model). This presents a conundrum as to whether to use a reduced accuracy physics-based model, where all of the internal states are understood or a potentially more accurate machine-learnt model where they are not. As stated previously, the methods presented are not intended to replace physics-based modelling, but to complement and enhance accuracy when necessary - the so-called grey-box model. Based upon the results presented thus far, it would seem that this may be a feasible way to usefully utilize flight test data gathered in the future.

Conclusion and Future Work

Gaussian process machine learning techniques have been used to develop rotorcraft response prediction models. The performance of these models have been assessed by comparing them with the original flight test data as well as to a corresponding physics-based model. From this investigation, the following conclusions can be drawn.

First, using the GP machine-learning approach it is possible to predict the on-axis rotational and heave translational rate responses of a rotorcraft. The level of accuracy achieved was observed both subjectively and objectively to be at least comparable and generally higher when compared to the physics-based model used for the study.

Second, using a model input structure that contained only the primary axis inceptor and a single lagged response term for that axis was sometimes found to give good response predictions. The most appropriate input structure for the proposed machine learning approach, i.e., the one that gave the most accurate results, was found to be one where all control inputs and a single lagged term for the axis of interest were used.

Finally, the variational sparse GP approach used between one-half and one-third of the training points used to create a 'full' model. These sparse models generally performed almost as well as the full models and still generally gave more accurate results than the physics-based model used. This technique, therefore, shows promise for use in real-world settings where the volume of available training data will, potentially, be significantly larger.

Whilst this investigation provides a useful initial insight into these techniques and their possible application to flight dynamics/simulation; there are further studies that need to be conducted. First, although the accuracy and form of the results achieved are impressive, there is room for improvement, both in terms of the raw accuracy of a given prediction and the level of confidence in that prediction. One crucial feature of rotorcraft dynamics is, of course, that the output responses are coupled. The most accurate models created in this investigation used multiple inceptors, but only the primary lagged response terms as inputs. One potential means to improve on the accuracy of the models would be to use a lagged response term from each response axis as input. The model outputs would then also feature response predictions for each axis. This multi-input multi-output approach has the potential to increase the accuracy of and/or reduce the uncertainty in the GP model predictions by including the cross-coupling aspects of the helicopter response.

It was noted in the Results Section that the full GP model response predictions do not encompass all of the real flight test data of the Bo105 rotorcraft. The authors aim to investigate this anomaly. One solution to this issue may be to use a mixture of kernels to capture the different response frequencies within the data. One kernel would be used for the high-frequency element of the response and another for the low-frequency element of the response.

The final key element of future work, as mentioned previously, was to generate a ‘grey-box’ helicopter model. GP models, once trained, can predict model error that is present in the current simulation response. The ‘grey box’ model will combine physical-law based models (i.e., ‘white box’) with data-based models (i.e., ‘black box’). When the GP model detects that the white-box model is not providing sufficiently accurate response predictions, then the black-box model will take-over, until such time as the white-box model can recover to an accurate position.

Acknowledgements

The authors gratefully acknowledge Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) Institut für Flugsystemtechnik for the provision of the flight test data and the permission to use it for this study.

References

[1] Shy, K. S., Hageman, J. J. and Le, J. H., “The Role of Aircraft Simulation in Improving Flight Safety Through Control Training,” NASA TM 2002-210731, 2002.

[2]Bell, H. H. and Waag, W. L., “Evaluating the Effectiveness of Flight Simulators for Training Combat Skills: A Review,” *The International Journal of Aviation Psychology*, Vol. 8, (3), 1998, pp. 223–242, DOI: 10.1207/s15327108ijap0803_4.

[3]Allerton, D. J., “The Impact of Flight Simulation in Aerospace,” *The Aeronautical Journal*, Vol. 114, (1162), 2010, pp. 747–756, DOI: 10.1017/S0001924000004231.

[4]Padfield, G. D., *Helicopter Flight Dynamics: the Theory and Application of Flying Qualities and Simulation Modelling*, John Wiley & Sons, Hoboken, NJ, 2008, Chapter 3.

[5]Punjani, A. and Abbeel, P., “Deep learning helicopter dynamics models,” Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, May 26–30, 2015.

[6]Myers III, P. L., Starr, A. W., and Mullins, K., “Flight Simulator Fidelity, Training Transfer, and the Role of Instructors in Optimizing Learning,” *International Journal of Aviation, Aeronautics, and Aerospace*, Vol. 5, (1), 2018, pp. 6, DOI: 10.15394/ijaaa.2018.1203.

[7]Anon, “JAR-FSTD(H), Helicopter Flight Simulation Training Devices,” Joint Aviation Authorities, May 2008.

[8]Anon, “FAA Advisory Circular AC120-63 Helicopter Simulator Qualification,” Federal Aviation Authority, November 1994.

[9]Perfect, P., Timson, E., White, M. D., Padfield, G. D., Erdos, R., and Gubbels, A. W., “A Rating Scale for the Subjective Assessment of Simulation Fidelity,” *The Aeronautical Journal*, Vol. 188, (1206), 2014, pp.953–974, DOI: 10.1017/S0001924000009635.

[10]Perfect, P., White, M. D., Padfield, G. D. and Gubbels, A. W., “Rotorcraft Simulation Fidelity: New Methods for Quantification and Assessment,” *The Aeronautical Journal*, Vol. 117, (1189), 2013, pp. 235–282, DOI: 10.1017/S0001924000007983.

[11]White, M. D., Perfect, P., Padfield, G. D., Gubbels, A. W., and Berryman, A. C., “Acceptance Testing and Commissioning of a Flight Simulator for Rotorcraft Simulation Fidelity Research,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 227, (4), 2013, pp. 663–686, DOI: 10.1177/0954410012439816.

[12]Tobias, E., Tischler, M., Berger, T., and Hagerott, S. G., “Full Flight-envelope Simulation and Piloted Fidelity Assessment of a Business Jet using a Model Stitching Architecture,” Paper AIAA 2015-1594,

Proceedings of the AIAA modeling and Simulation Technologies Conference, Kissimmee, FL, January 05–09, 2015.

[13]Lu, L., Padfield, G. D., White, M., and Perfect, P., "Fidelity Enhancement of a Rotorcraft Simulation Model through System Identification," *The Aeronautical Journal*, Vol. 115, (1170), 2011, pp. 453–470, DOI: 10.1017/S0001924000006102.

[14]Manso, S., "Simulation and System Identification of Helicopter Dynamics using Support Vector Regression," *The Aeronautical Journal*, Vol. 119, (1222), 2015, pp. 1541–1560, DOI: 10.1017/aer.2016.69.

[15]Kumar, M. V., Omkar, S.N., Ganguli, R., Sampath, P., and Suresh, S., "Identification of Helicopter Dynamics using Recurrent Neural Networks and Flight Data," *Journal of the American Helicopter Society*, Vol. 51, (2), 2006, pp. 164–174, DOI: 10.4050/JAHS.51.164.

[16]Omkar, S. N., Mudigere, D., Senthilnath, J., and Kumar, M. V., "Identification of Helicopter Dynamics based on Flight Data using Nature Inspired Techniques," *International Journal of Applied Metaheuristic Computing (IJAMC)*, Vol. 6, (3), 2015, pp. 38–52, DOI: 10.4018/ijamc.2015070102.

[17]Titsias, M. K., "Variational Model Selection for Sparse Gaussian Process Regression," University of Manchester, 2009.

[18]Titsias, M., "Variational Learning of Inducing Variables in Sparse Gaussian Processes," Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, FL, May 09–12, 2009.

[19]Jackson, R. D. and Jump, M. and Green, P. L., "Towards Gaussian Process Models of Complex Rotorcraft Dynamics, Proceedings of the 74th Annual Forum of the Vertical Flight Society, Phoenix, AZ, May 14–17, 2018.

[20]Bishop, C., *Pattern Recognition and Machine Learning*, Springer, New York, NY, 2006, Chapters 2, 3, 6 and 11.

[21]Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning*, MIT, Cambridge, MA, 2006, Chapters 2, 5 and 8.

[22]Berger, B., Rauscher, F., and Lohmann, B., "Analysing Gaussian Processes for Stationary Black-box Combustion Engine Modelling," Proceedings of the 18th IFAC World Congress, Milano, Italy, August 28–September 02, 2011.

[23]Gutjahr, T., Kleinegräber, H., Ulmer, H., Kruse, T., and Eckstein, C., "New Approaches for Modeling Dynamic Engine Behavior with Gaussian Processes," Proceedings of the 7th Conference of Design of Experiments (DoE) in Engine Development, Berlin, Germany, June 18–19, 2013.

[24]Berger, B., and Rauscher, F., "Robust Gaussian Process Modelling for Engine Calibration," Proceedings of the 7th Vienna International Conference on Mathematical Modelling (MATHMOD), Vienna, Austria, Feburary 14–17, 2012.

[25]Higdon, D., Gattiker, J., Williams, B., and Rightley, M., "Computer Model Calibration using High-dimensional Output," *Journal of the American Statistical Association*, Vol. 103, (482), 2008, pp. 570–583, DOI: 10.1198/016214507000000888.

[26]Girard, A., "Approximate Methods for Propagation of Uncertainty with Gaussian Process Models," University of Glasgow, 2004.

[27]Green, P. L., "Bayesian System Identification of a Nonlinear Dynamical System using a Novel Variant of Simulated Annealing," *Mechanical Systems and Signal Processing*, Vol. 52, 2015, pp. 133–146, DOI: 10.1016/j.ymssp.2014.07.010.

[28]Snelson, E., and Ghahramani, Z., "Sparse Gaussian Processes using Pseudo-inputs," Proceedings of the 20th Annual Conference on Advances in Neural Information Processing Systems, Vancouver, B.C., Canada, December 04–09, 2006.

[29]Padfield, G. D., *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*, Blackwell Science, Oxford, UK, 1996, Chapter 3.

[30]Anon, "GARTEUR HC(AG06) : Mathematical Modelling for the Prediction of Helicopter Flying Qualities," GARTEUR TP-075, 1996.

[31]Heffley, R. K., Lehman, J. M., and van Winkle, R. A., "A Compilation and Analysis of Helicopter Handling Qualities Data, Volume One: Data Compilation," NASA CR-3144, 1979.

[32]Dieterich, O., Götz, J., Pavel, M., Dangvu, B., Masarati, P., Quaranta, G., Gennaretti, M., Serafini, J., Jump, M., and Liu, L., "Rigid Body and Aeroelastic Rotorcraft-Pilot Coupling (RPC) – Prediction Tools and Means for Prevention," GARTEUR LIMITED, 2008.

[33]Lawrence, D. B., "The Flying Qualities of the Wright Flyers," PhD Thesis, Department of Engineering, University of Liverpool, September 2004.

[34] Zang, C., Xin, H., and Driscoll, J., "Development and Validation of an Engineering Simulation Model in FLIGHTLAB with Customized Modeling Enhancements," Proceedings of the 73rd Annual Forum of the American Helicopter Society, Fort Worth, TX, May 09–11, 2017.

[35] Xin, H., Zhang, C., and Driscoll, J. T., "Enhancement of an Engineering Simulation Model to Improve the Correlation with Flight Test Data in Climb/Descent and Autorotation," Proceedings of the 75th Annual Forum of the American Helicopter Society, Philadelphia, PA, May 13–16, 2019.

[36] Götz, J., "Garteur HC AG16/WP-7 - Bo105: Flight Test Data," Institute of Flight Systems, Braunschweig, Germany, April 2009.