

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 model response errors in the predictions compared to the real vehicle. This study investigated whether a machine-learning technique could be employed to provide rotorcraft dynamic response predictions, with the ultimate aim of this model taking over when the physics-based model's accuracy degrades. In the current work, a machine-learning technique was employed to train a model to predict the dynamic response of a rotorcraft. 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 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 GP model predictions.

Keywords: Rotorcraft, Machine Learning, Gaussian Process, Flight Simulation

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 of pilots, 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 having to be made, 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 model and 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 available 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. It should be noted, of course 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. Here, the methods used are probabilistic and are therefore able to capture uncertainties in the predicted model outputs based upon the data used to create it.

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 rate, roll rate, yaw rate and 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 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 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 volumes of data available from, for example, flight test campaigns.

In the paper, GP models are developed to predict pitch rate, roll rate, yaw rate and heave 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 (Ref. [19]). The research presented in this paper builds on the author’s previous work (Ref. [20]) by developing simulation methods that can accurately capture the complex dynamics of a rotorcraft, while still being able to run in real time.

In summary, the paper investigates:

1) Whether, using a machine-learning approach, it is possible to predict the pitch rate, roll rate, yaw rate and heave of a rotorcraft with a level of fidelity that is comparable to that which can be achieved using physical-law based models (FLIGHTLAB, in this case).

2) The most appropriate input structure for the proposed machine learning approach, given that longitudinal, lateral, pedal and collective lever input data is available.

3) The *scalability* of the proposed approach. Specifically, envisaging future applications where much larger sets of training data will likely be utilized, it is investigated whether the use of variational sparse GPs can significantly reduce the cost of applying the proposed method to larger data sets.

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 draws 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) have been widely used in recent years for many different applications, such as in engine modelling (Refs. [21], [22], [23]) and robotics (Ref. [24]). In the current work, GPs are used to perform regression (they can also perform other tasks such as classification (Ref. [25])). 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*.

*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

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 (Ref. [25], [26]), is considered, to aid the understanding of a Gaussian process approach. Such a model typically takes the form

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

where \mathbf{x} is the input vector, \mathbf{w} is a vector of parameters to be identified and $\boldsymbol{\phi}$ is a vector of ‘basis functions’. Adopting a Bayesian approach (Ref. [25]) to the identification of \mathbf{w} , the prior distribution over \mathbf{w} can, for example, be Gaussian:

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

where γ is the precision of the distribution, and \mathbf{I} is an identity matrix. Here, the function values at the points where training data are available is written as the vector \mathbf{f} :

$$\mathbf{f} = \begin{pmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_N) \end{pmatrix} \quad (3)$$

where N is the number of training points. Using Eq. (1), the vector \mathbf{f} can be written as:

$$\mathbf{f} = \boldsymbol{\Phi} \mathbf{w} \quad (4)$$

where Φ is referred to as the ‘design matrix’ (for more information see (Ref. [25])). Given the prior in Eq. (2), it can then be shown that the prior distribution over \mathbf{f} is Gaussian, with mean

$$\mathbb{E}[\mathbf{f}] = \Phi \mathbb{E}[\mathbf{w}] = \mathbf{0} \quad (5)$$

and covariance matrix

$$\text{cov}[\mathbf{f}] = \mathbb{E}[\mathbf{f}\mathbf{f}^T] = \Phi \mathbb{E}[\mathbf{w}\mathbf{w}^T] \Phi^T = \frac{1}{\gamma} \Phi \Phi^T = \mathbf{K} \quad (6)$$

where \mathbf{K} is a matrix with elements:

$$K_{nm} = \frac{1}{\gamma} \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m); \quad n, m = 1, \dots, N. \quad (7)$$

At this stage, y_n is defined as the n th observation of the modelled system’s response. The following ‘noise model’ is often assumed:

$$y_n = f_n + \epsilon_n, \quad \epsilon_n \sim \mathcal{N}(\epsilon_n | 0, \beta^{-1}) \quad (8)$$

where $f_n \equiv f(x_n)$ and β is the precision of the noise that corrupted the observation. A typical Bayesian approach would then involve inferring the parameters, \mathbf{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 \mathbf{f} directly, such that

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f} | \mathbf{0}, \mathbf{K}_{NN}) \quad (9)$$

where the covariance matrix, \mathbf{K}_{NN} is given by

$$\mathbf{K}_{NN} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_1) & \dots & k(\mathbf{x}_N, \mathbf{x}_1) \\ k(\mathbf{x}_1, \mathbf{x}_2) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_N, \mathbf{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ k(\mathbf{x}_1, \mathbf{x}_N) & k(\mathbf{x}_2, \mathbf{x}_N) & \dots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

and where $k(\cdot, \cdot)$ is a user-defined ‘kernel function’. The kernel function is chosen so that \mathbf{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. [25]):

$$\mathbf{K}_{NN}(n, m) = k(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-\frac{\alpha}{2}(\mathbf{x}_n - \mathbf{x}_m)^T(\mathbf{x}_n - \mathbf{x}_m)\right) \quad (10)$$

where the ‘hyperparameter’, α , induces correlations that depend on the closeness of \mathbf{x}_n to \mathbf{x}_m . Choosing different hyperparameters can affect the accuracy of the resultant GP model. The optimum hyperparameters are most commonly identified by maximising the likelihood of witnessing the observed data, given by

$$p(\mathbf{y} | \mathbf{f}) = \mathcal{N}(\mathbf{y} | \mathbf{f}, \beta^{-1}\mathbf{I}) \quad (11)$$

(optimization techniques are discussed further in Section (Optimization of hyperparameter for Gaussian Process training)). The marginal distribution over \mathbf{y} is:

$$p(\mathbf{y}) = \int p(\mathbf{y} | \mathbf{f})p(\mathbf{f})d\mathbf{f} \quad (12)$$

which, given equations (9) and (11), allows the marginal distribution over \mathbf{y} to be written as:

$$p(\mathbf{y}) = p(\mathbf{y} | \mathbf{0}, \mathbf{C}) \quad (13)$$

where the elements of \mathbf{C} are given by

$$C(n, m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1}I_{nm} \quad (14)$$

and I_{nm} is the (n, m) th element of the identity matrix 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, \mathbf{x}^* , the probability of observing a new point, y^* , given previous observations, \mathbf{y} , can be obtained. Defining $\mathbf{y}^* = [y_1, y_2, \dots, y_N, y^*]^T$, the joint distribution over \mathbf{y}^* is written as:

$$p(\mathbf{y}^*) = \mathcal{N}(\mathbf{y}^* | \mathbf{0}, \mathbf{C}_{N+1}) \quad (15)$$

where \mathbf{C}_{N+1} is a $(N + 1) \times (N + 1)$ covariance matrix:

$$\begin{bmatrix} \mathbf{C} & \mathbf{k} \\ \mathbf{k}^T & c \end{bmatrix} \quad (16)$$

where

$$\mathbf{k} = k(\mathbf{x}_n, \mathbf{x}^*), \quad n = 1, \dots, N \quad (17)$$

and

$$c = \beta^{-1} + k(\mathbf{x}^*, \mathbf{x}^*). \quad (18)$$

The mean and variance of y^* given \mathbf{y} can be shown to be (Ref. [25]):

$$\mu(y^* | \mathbf{y}) = \mathbf{k}^T \mathbf{C}^{-1} \mathbf{y}$$

$$\sigma^2(y^* | \mathbf{y}) = c - \mathbf{k}^T \mathbf{C}^{-1} \mathbf{k}$$

such that:

$$p(y^* | \mathbf{y}) = \mathcal{N}(y^* | \mu, \sigma^2) \quad (19)$$

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:

$$y^* \sim \mathbf{GP}(\mathbf{x}^*). \quad (20)$$

Choice of Kernel

The kernel function chosen for the current study was used, for example, in (Ref. [27]). The kernel takes the form

$$k(\mathbf{x}_n, \mathbf{x}_m) = \prod_{i=1}^{N_D} \alpha^{4(x_n^i - x_m^i)^2} \quad (21)$$

where x^i is used to represent the i th element of the vector \mathbf{x} and N_D is the number of dimensions of the input vector. The kernel, Eq. (21), 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:

$$y_n = f(y_{n-1}, y_{n-2}, y_{n-3}, \dots, v_n, v_{n-1}, v_{n-2}, v_{n-3}, \dots) + \epsilon_n \quad (22)$$

where v is a generic system input, n is a discrete time step and, 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 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:

$$\mathbf{x}_n = \begin{pmatrix} \delta_n^x \\ \delta_n^y \\ \delta_n^p \\ \delta_n^o \\ y_{n-1} \end{pmatrix} \quad (23)$$

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:

$$\mathbf{x}_n = \begin{pmatrix} \delta_n^x \\ y_{n-1} \end{pmatrix}. \quad (24)$$

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:

$$y_n^* = \mathbf{GP}(\delta_n^x, \delta_n^y, \delta_n^p, \delta_n^o, y_{n-1}) \quad (25)$$

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. (25), is considered. This prediction is written as y_n^* . Following this, predictions of y_{n+1}^* would be realized according to

$$y_{n+1}^* = \mathbf{GP}(\delta_{n+1}^x, \delta_{n+1}^y, \delta_{n+1}^p, \delta_{n+1}^o, y_n^*) \quad (26)$$

(where, for this example, the input structure is shown in Eq. (24) was used).

The key aspect to note with regard to Eq. (26) 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.

Algorithm 1 Full Model Predictions algorithm

```

1: for  $r = 1 : R$  do
2:    $Y_{(n),r}^* \sim \mathbf{GP}(y_{(n-1)}, \delta_n)$ 
3:    $Y_{(n+1),r}^* \sim \mathbf{GP}(Y_{(n),r}^*, \delta_{n+1})$ 
4: end for

```

Where R is the number of Monte Carlo samples and Y denotes a random sample. For a more detailed algorithm for FMP, Ref. [28] is recommended.

Optimization of hyperparameters for Gaussian Process training

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

$$p(\theta | \mathbf{y}) \propto p(\mathbf{y} | \theta)p(\theta). \quad (27)$$

The techniques used to train the standard GPs used in the current paper are discussed in the next Section.

Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) can be used to generate samples from the posterior hyperparameter distribution given in Eq. (27), thus quantifying uncertainties in the hyperparameter selection. It also provides a more global search, unlike the local search provided by methods such as gradient-based ascent (Ref. [25]). Utilizing MCMC to generate samples from the posterior hyperparameter distribution enables the incorporation of hyperparameter uncertainty into the GP prediction. The authors investigated hyperparameter uncertainty in Ref. [20], where it was concluded that hyperparameter uncertainty had little effect on the outcome of the model. Therefore, it was deemed not necessary and will not be used in the current work. It is important to note that, even if hyperparameter uncertainty is not being considered, GPs still provide probabilistic predictions and, as such, uncertainty is still considered in the following analysis.

The best known MCMC method is the Metropolis algorithm (MA) (Ref. [29]). In the following, the target distribution, $\pi(\boldsymbol{\theta})$ (i.e. the distribution from which samples are generated), is equal to $p(\boldsymbol{\theta} | \mathbf{y})$.

The first step in each iteration of MA is to propose a new state $\boldsymbol{\theta}'$, where the current state of the Markov chain is $\boldsymbol{\theta}^{(r)}$ (Ref. [30]). The proposed state is generated from a probability density function (PDF), $q(\boldsymbol{\theta}' | \boldsymbol{\theta}^{(r)})$, which is conditional on the current state. The proposal is accepted as the new state of the Markov chain with probability:

$$\min \left\{ 1, \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(r)})} \right\} = \min \left\{ 1, \frac{\pi^*(\boldsymbol{\theta}')}{\pi^*(\boldsymbol{\theta}^{(r)})} \right\} \quad (28)$$

where $\pi^*(\boldsymbol{\theta})$ is the un-normalized target distribution. If accepted, the new state of the Markov chain is $\boldsymbol{\theta}^{(r+1)} = \boldsymbol{\theta}'$, otherwise $\boldsymbol{\theta}^{(r+1)} = \boldsymbol{\theta}^{(r)}$. This process is run for a user-defined number of samples. The Markov chain will then reach a stationary distribution, producing posterior samples of the hyperparameters.

Simulated Annealing

Simulated annealing is a form of MCMC which uses MA. The difference between simulated annealing and MA, is that, while running, the method slowly increases the influence of the likelihood via a variable ζ (Ref. [31]). Increasing the influence of the likelihood ($p(\mathbf{y} | \boldsymbol{\theta})$), as shown in Eq. (27), increases the influence of the data. This assists algorithm convergence. As with the MA, a prior probability distribution is chosen to generate an initial candidate vector of hyperparameters while a small initial value of ζ is also chosen. ζ is then increased where, for each ζ value, a full run of the MA is conducted using a user-defined number of samples. The rate at which ζ is increased is called the annealing schedule. In the current paper, an adaptive annealing schedule is used to ensure a constant change in Shannon entropy of the target distribution (Ref. [31]). The advantage of using this method over the standard MA is that it is easier to tune and often demonstrates better convergence, as the proposal distribution's covariance matrix can be updated after each run of the MA (Ref. [31]). The optimization stops when $\zeta = 1$, when the complete likelihood ($p(\mathbf{y} | \boldsymbol{\theta})$) has been fully introduced.

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. However, 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 (herein 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 selected such inducing points automatically. One approach by Snelson and Ghahramani (Ref.[32]) uses inducing points that do not have to come from the initial training set, called 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 (for more details see (Refs.[17] and [18])) that is employed in the current work.

As previously mentioned, the variational sparse approach aims to make the sparse GP closely approximate the ‘full’ GP. In the current paper, the variational sparse GP selected a subset of m training points, \mathbf{X}_M , from the available input data such that:

$$\mathbf{X}_M \subset \mathbf{X}, \quad \text{where} \quad \mathbf{X}_M = \{\mathbf{x}_{m,1}, \dots, \mathbf{x}_{m,M}\}. \quad (29)$$

The ‘optimum choice’ of training points must first be established. Defining:

$$\mathbf{f}_M = \{f(\mathbf{x}_{m,1}), \dots, f(\mathbf{x}_{m,M})\} \quad (30)$$

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

$$p(\mathbf{f}, \mathbf{f}_M | \mathbf{y}) = \frac{p(\mathbf{y} | \mathbf{f}, \mathbf{f}_M)p(\mathbf{f}, \mathbf{f}_M)}{p(\mathbf{y})}. \quad (31)$$

In (Refs. [17] and [18]) it is shown that the optimum choice of \mathbf{f}_M leads to:

$$p(\mathbf{y} | \mathbf{f}_M) = p(\mathbf{y} | \mathbf{f}, \mathbf{f}_M) \quad (32)$$

such that, if \mathbf{f}_M is already known, then knowing \mathbf{f} does not add additional information with regards to the probability of witnessing \mathbf{y} . When the condition Eq. (32) is satisfied then it can be shown that the posterior distribution factorizes as:

$$p(\mathbf{f}, \mathbf{f}_M | \mathbf{y}) = p(\mathbf{f}_M | \mathbf{y})p(\mathbf{f} | \mathbf{f}_M). \quad (33)$$

The Variational Bayesian approach (Ref. [25]) can then be used to approximate $p(\mathbf{f}, \mathbf{f}_M | \mathbf{y})$ with the

probability density function (PDF):

$$q(\mathbf{f}, \mathbf{f}_M) = \psi(\mathbf{f}_M)p(\mathbf{f} | \mathbf{f}_M) \quad (34)$$

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

$$\text{KL}(q(\mathbf{f}, \mathbf{f}_M) || p(\mathbf{f}, \mathbf{f}_M | \mathbf{y})). \quad (35)$$

Note that minimization of the KL divergence implies that the posterior can be factorized as shown in Eq. (32). The chosen subset, \mathbf{X}_M , that minimizes Eq. (35) will, therefore, be the optimum 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} (Ref. [25]), where

$$\mathcal{L} = \int \int q(\mathbf{f}, \mathbf{f}_M) \log \left[\frac{p(\mathbf{f}, \mathbf{f}_M, \mathbf{y})}{q(\mathbf{f}, \mathbf{f}_M)} \right] d\mathbf{f} d\mathbf{f}_M. \quad (36)$$

A lower bound on \mathcal{L} , denoted F_v , can be found through Jensen’s inequality (Refs. [17] and [18]) and is a functional of \mathbf{X}_M :

$$F_v(\mathbf{X}_M) = -\frac{1}{2} \left(N \log(2\pi) + \log \left(\left| \mathbf{I}\beta^{-1} + \mathbf{Q} \right| \right) + \mathbf{X}^T \mathbf{X} (\mathbf{I}\beta^{-1} + \mathbf{Q})^{-1} - \beta \text{Tr}(\mathbf{K}_{NN} - \mathbf{Q}) \right) \quad (37)$$

where:

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

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

This approach is effective due to the reduced computational cost associated with evaluating F_v . As equa-

tion (37) involves just the trace of \mathbf{K}_{NN} , only the diagonal elements of \mathbf{K}_{NN} are required. The matrices in Eq. (37) 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 is significantly smaller than N). For example, in the current study, the datasets have 800 points, which if the GP model was trained on all available data, N would be 800. The subset M could be any value that is less than 800 and for the GP model to make accurate predictions. The same cost function is used to optimize the hyperparameters. The selection of the training points and the optimization of the hyperparameters is conducted in an Expectation-Maximization ‘like’ algorithm, as described in (Refs. [17] and [18]).

To begin training a variational sparse GP model, one must choose a small number of initial training data points. The K-means clustering algorithm (Ref. [25]) was used for this 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 \mathbf{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).

Method

Gaussian Process rotorcraft dynamics models were trained using Bo105 flight test datasets. Two different approaches to select the training points were used. One referred to as the full GP where 32 training points were chosen at equal time intervals. The other approach uses the variational sparse GP method, in which the training points are automatically selected. The response of these models was then obtained and compared with the corresponding flight test response as well as the output of 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

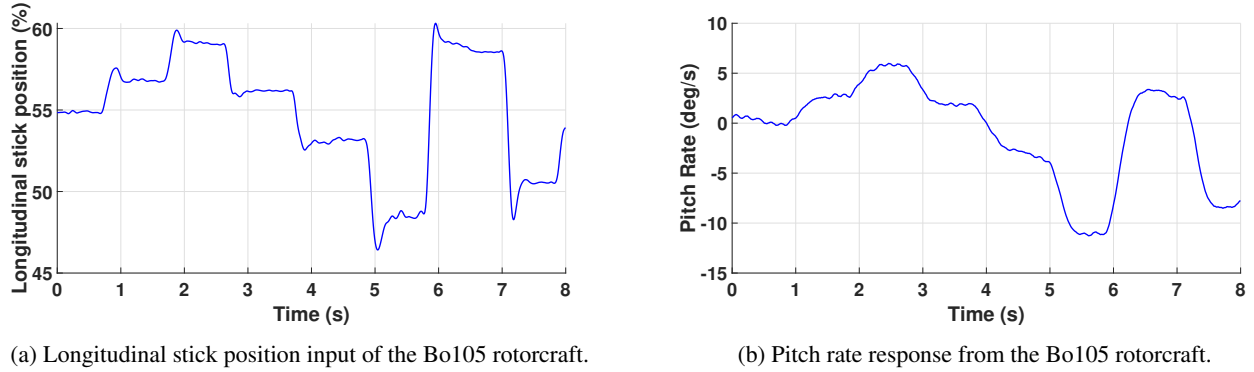


Fig. 1: Example of the Bo105 flight test data

dataset was delivered to the University of Liverpool as part of the GARTEUR HC(AG-16) Rotorcraft Pilot Couplings (Ref. [33]). An example of the flight test data from the Bo105 rotorcraft is shown in Figure 1, where Figure 1a is the longitudinal stick position and Figure 1b is the pitch rate response corresponding to the longitudinal stick position.

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

Figure 2 shows the primary inceptor time-history for each maneuver in Table 1. Also shown are the

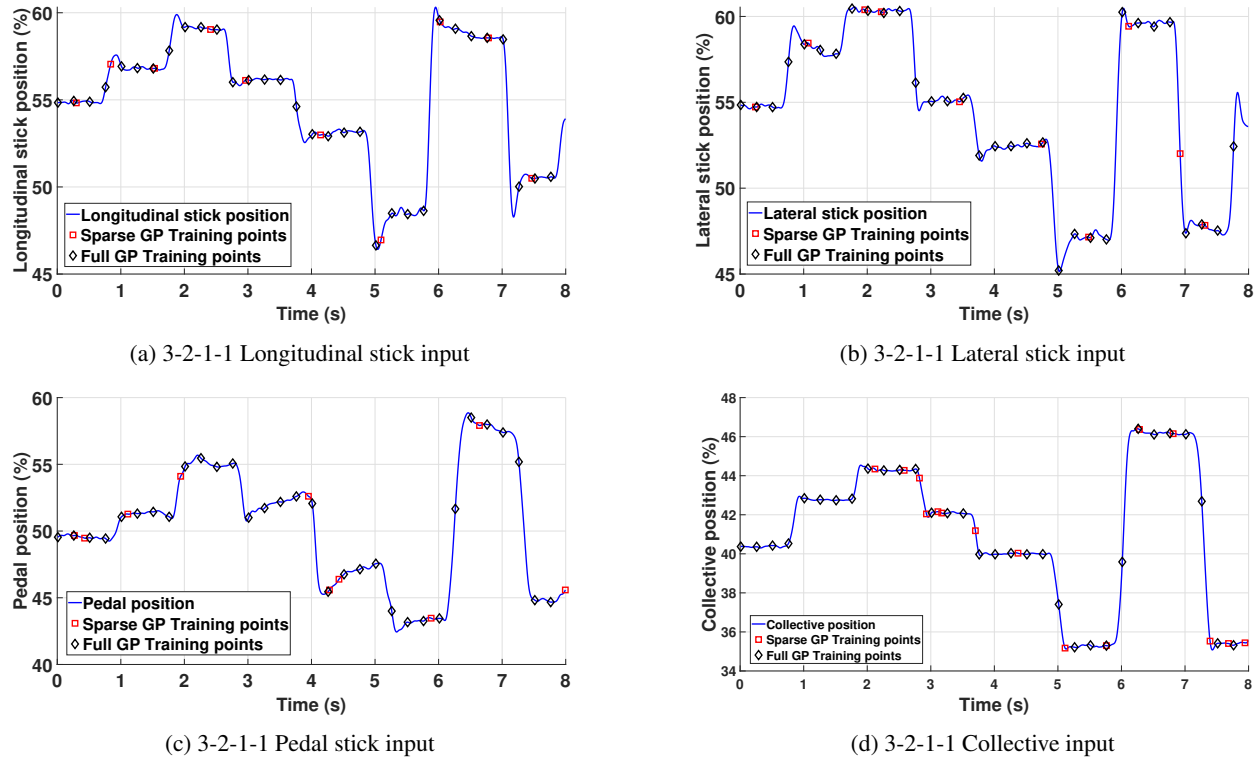


Fig. 2: Time history of each 3-2-1-1 maneuver

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. These were selected to be at a constant linear time interval to give reasonable coverage across the entire maneuver. The sparse GP training data points are shown by red squares; these data points are selected automatically by the variational sparse GP approach.

For each axis, rotorcraft response predictions were then made using 2 different forms of the GP model input structure. The first input structure contains the primary input for the corresponding response, as well as a lagged response. For example, to predict the n th pitch rate, the input structure ‘Longitudinal Input and one lagged Pitch Rate’ refers specifically to inputs of the form:

$$\mathbf{x}_n = \begin{pmatrix} \delta_n^x \\ y_{n-1} \end{pmatrix} \quad (38)$$

where y_{n-1} is the lagged response (i.e. q at time $n - 1$). The second input structure also contains the non-primary inputs. For example, to predict the n th pitch rate, the input structure ‘All Inputs and one

lagged Pitch Rate’ refers specifically to inputs of the form:

$$\mathbf{x}_n = \begin{pmatrix} \delta_n^x \\ \delta_n^y \\ \delta_n^p \\ \delta_n^o \\ y_{n-1} \end{pmatrix} \quad (39)$$

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 study was first extracted from Ref. [34] which itself incorporates open-source data collected from GARTEUR HC(AG06) activities in Ref. [35]. The flight control system data were first taken from Ref. [36]. The more complete database that was provided to the GARTEUR Action Group HC(AG16) (Ref. [37]) was then utilized. The entire rotorcraft modelling dataset has been most recently reported in Ref. [38].

FLIGHTLAB

The physics-based rotorcraft model was developed using FLIGHTLAB. ‘The objective of FLIGHTLAB is to promote Concurrent Engineering by providing a simulation tool capable of multidisciplinary support with selective fidelity modelling options’ (Ref.[39]). FLIGHTLAB uses a modular approach to simulation model building at a level appropriate to the simulation data available (so, e.g., an engine could be a simple power/thrust against throttle position look-up table or it could be a more detailed thermodynamic representation where individual components such as crankshafts and gearboxes are modelled). FLIGHTLAB’s capabilities are provided in Ref. [39] and a summary version is provided in Ref. [40]. Note, the FLIGHTLAB model shown in the current work was configured by an ‘expert’ user.

Model Response Prediction Comparison Method

In order to be able 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 Eq. 40.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i^* - y_i)^2} \quad (40)$$

where y^* is the 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.

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. Note that this analysis was conducted using the 'full' GP - the variational sparse GP is investigated in the Section that follows. The difference between the two approaches is that full GP uses 32 training points selected with equal time intervals and the variational sparse GP automatically selects the training points. Note that in the current Section, the GP is trained using 32 training points and predicts 800 response points, interpolating between the training points.

Longitudinal Stick Input to Pitch Rate Prediction

Figure 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 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. The OSAP for the input structure of all of the inputs is not shown for the full GP as it produces very similar results to the input structure using the on-axis input (for example longitudinal input to pitch rate). Note, however, that the RMSE values for all test cases are provided in the RMSE error Section.

Figure 4 shows the pitch rate response for:

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

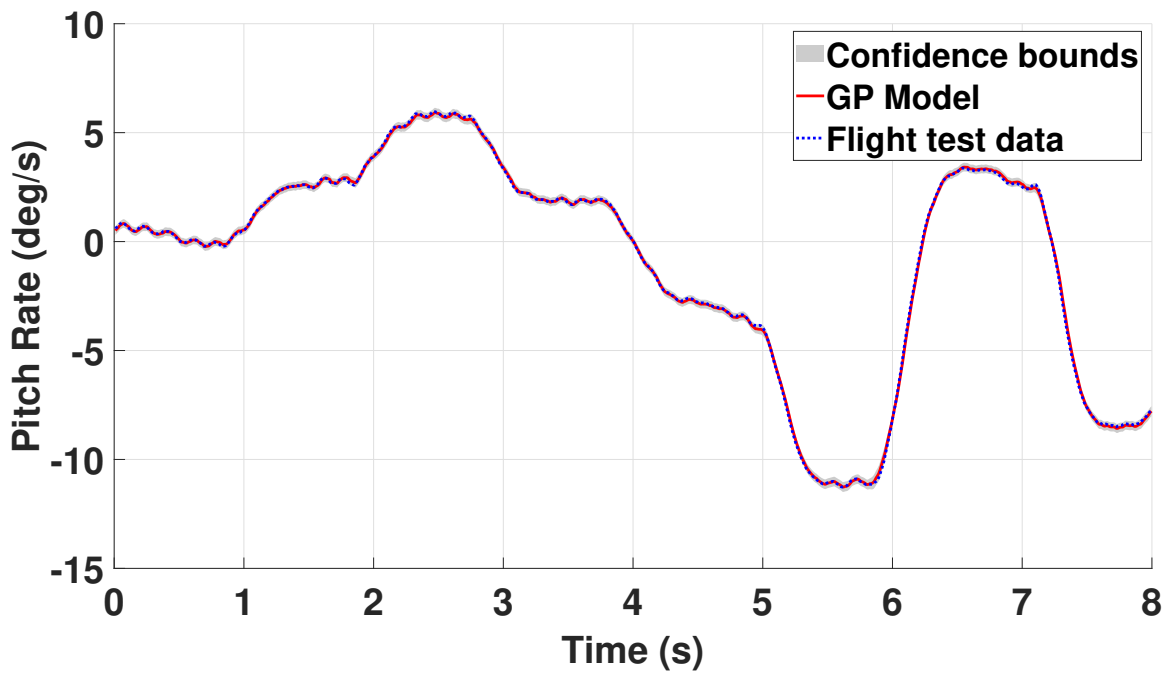


Fig. 3: One step ahead predictions of the pitch rate 3-2-1-1 response for the input structure containing the longitudinal input and one lagged pitch rate (Case 1 in Table 1).

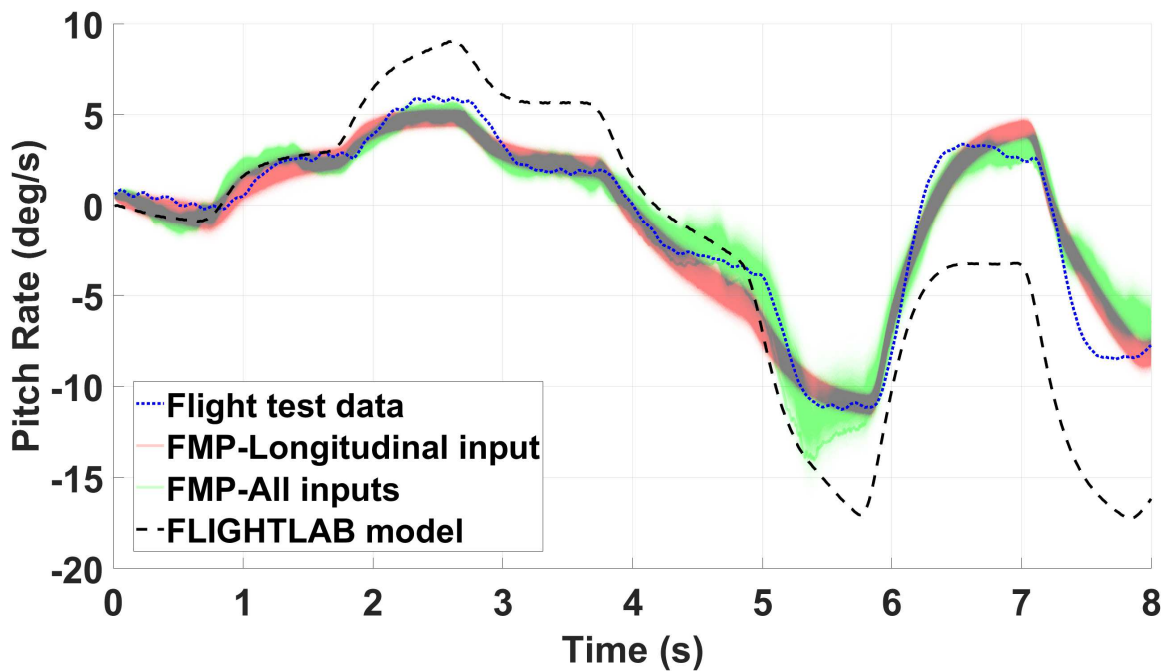


Fig. 4: Realization of the ‘full’ GP pitch rate prediction for input structure containing the Longitudinal input with one lagged pitch rate (Case 2 in Table 1) and ALL inputs with one lagged pitch rate (Case 4 in Table 1) with the comparison of the FLIGHTLAB model.

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. Figure 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 realizations were observed. A realization is one set of GP predictions.

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), as shown in Eq. 23, was used. It is clear from Fig. 4 that the predictions are not as accurate as for the OSAP model shown in Fig. 3. This is to be expected given that, for this model, the previous prediction becomes an input to the next prediction. Hence, predictive uncertainty is carried through the simulation.

It can be seen from the results presented, 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. These trends are easily observable in the RMS error data for these responses, shown in Table 2 towards the end of this Section.

An interesting observation of the GP response prediction in Fig. 4 is that the ensemble of GP predictions 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

The OSAP for the roll rate response that used the current lateral stick position and one lagged roll rate input structure (Case 5, Table 1) is shown in Fig. 5. It is clear that, as per the pitch rate response of Fig.

3, the GP predictions are accurate with very small (i.e. high) confidence bounds.

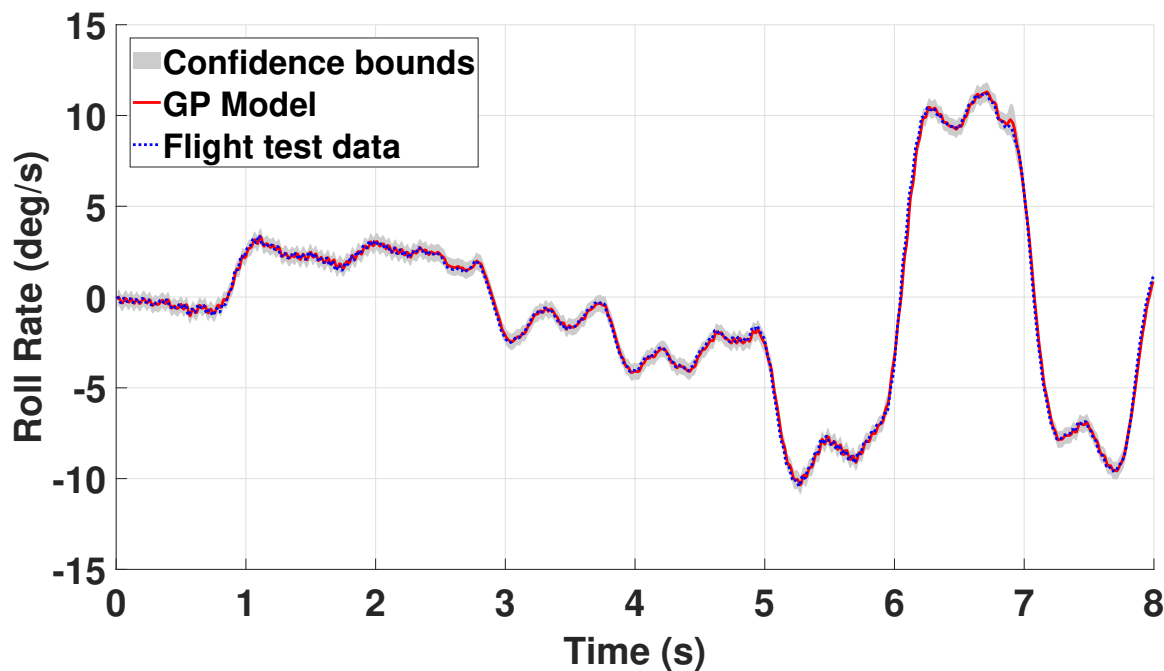


Fig. 5: One step ahead predictions of the roll rate 3-2-1-1 response for the input structure containing the lateral input and one lagged roll rate (Case 5 in Table 1).

Figure 6 shows the lateral response comparison between the flight test data and the GP and FLIGHTLAB models. 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 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. This is confirmed in Fig. 6, whereby 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. The RMS error

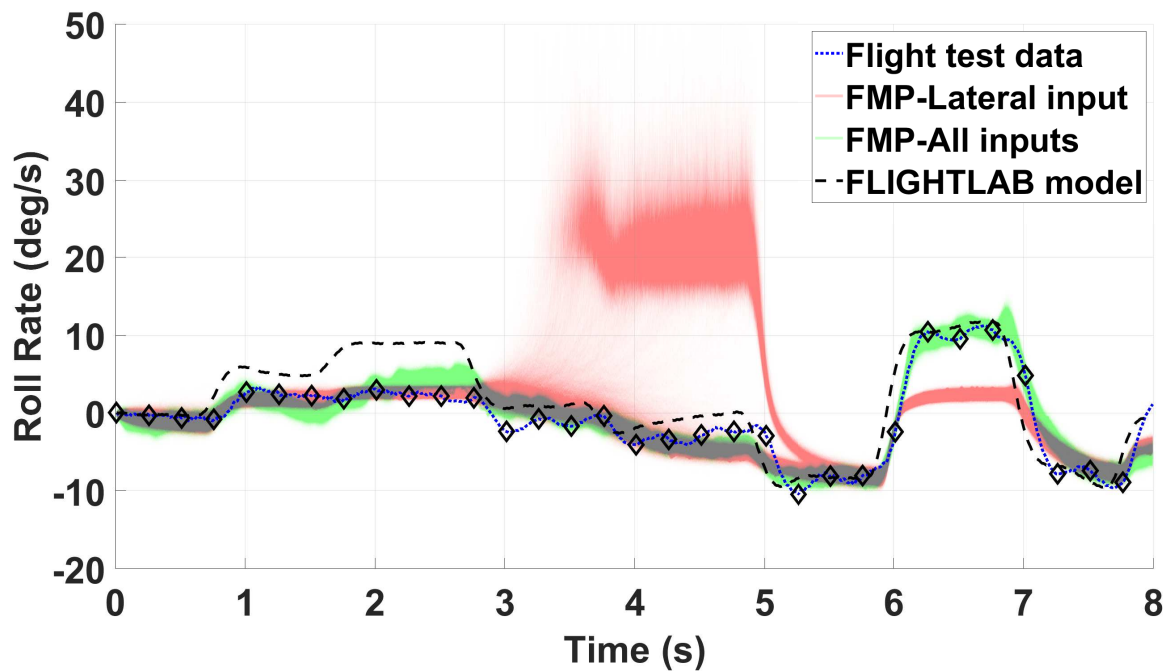


Fig. 6: Realization of the ‘full’ GP roll rate prediction for input structure containing the Lateral input with one lagged roll rate (Case 6 in Table 1) and ALL inputs with one lagged roll rate (Case 8 in Table 1) with the comparison of the FLIGHTLAB model.

values that provide objective data to support this statement are shown in Table 2.

Rudder Pedal Input to Yaw Rate Prediction

The GP OSAP for the yaw rate prediction using the current pedal input and one lagged yaw rate input structure (Case 9, Table 1) is shown in Fig. 7. As per the results shown in Fig. 3 and Fig. 5, the predictions are accurate and hence, the corresponding confidence bounds very small.

Figure 8 shows the comparisons between the GP FMP models, the FLIGHTLAB model and the flight test data. All of the model predictions appear less accurate for this axis than for the 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 data, 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.

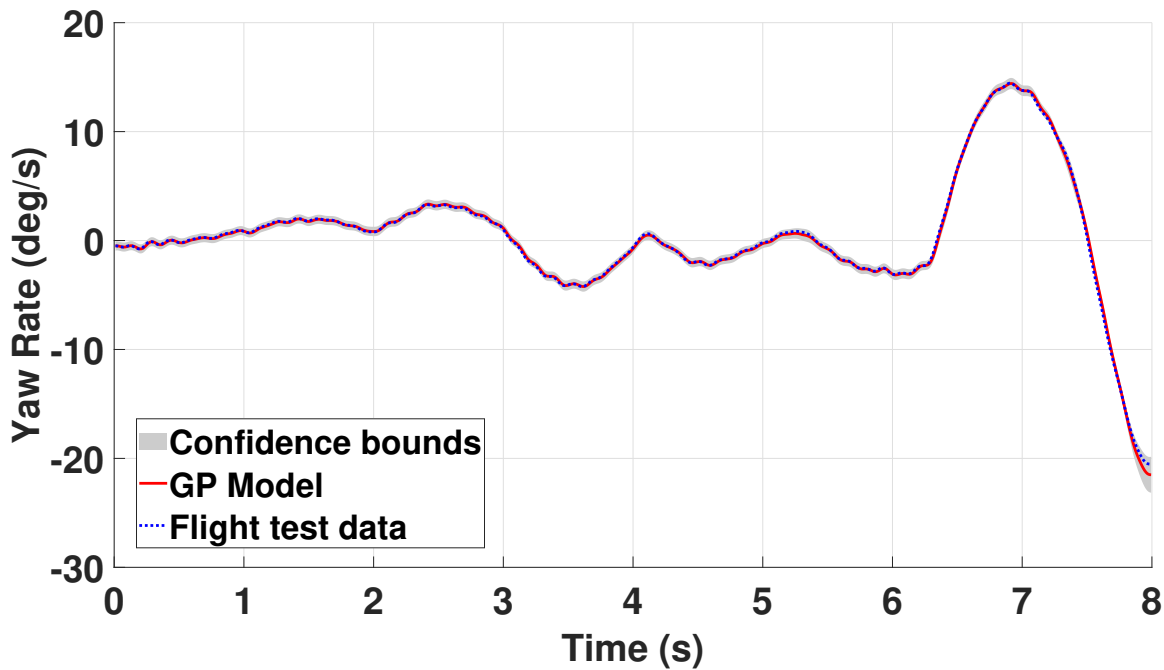


Fig. 7: One step ahead predictions of the yaw rate 3-2-1-1 response for the input structure containing the pedal input and one lagged yaw rate (Case 9 in Table 1).

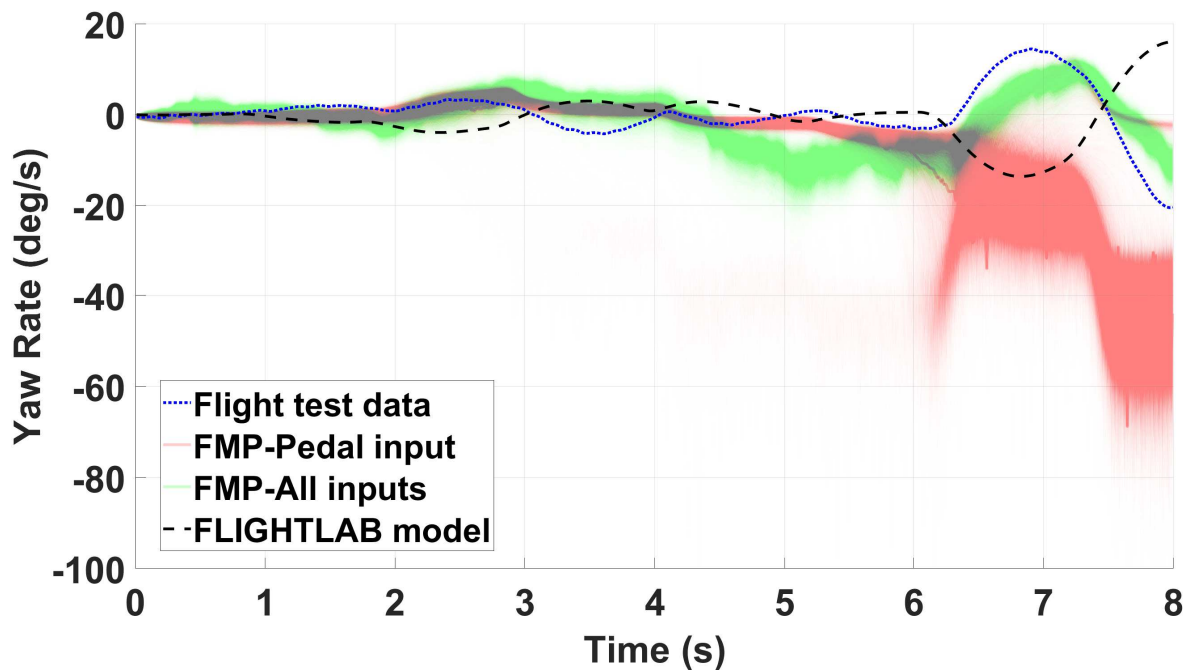


Fig. 8: Realisation of the 'full' GP yaw rate prediction for input structure containing the Pedal input with one lagged yaw rate (Case 10 in Table 1) and ALL inputs with one lagged yaw rate (Case 12 in Table 1) with the comparison of the FLIGHTLAB model.

Collective Input to Heave Response Prediction

Figure 9 shows the OSAP for the GP model using the input structure containing the current collective lever and one lagged heave term (Case 13, Table 1). Compared to Fig. 3, Fig. 5 and Fig. 7 the confidence bounds are somewhat larger, however, the GP predictions remain accurate.

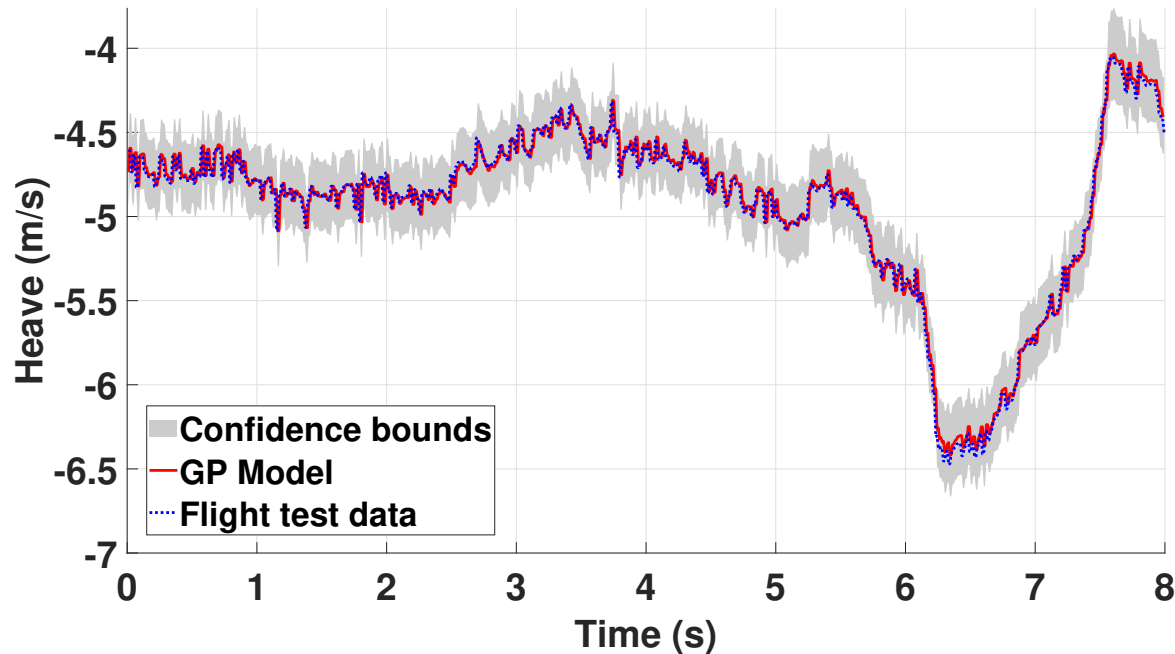


Fig. 9: One step ahead predictions of the heave 3-2-1-1 response for the input structure containing the collective input and one lagged heave (Case 13 in Table 1).

Figure 10 shows the heave response comparison between the FLIGHTLAB and GP models and the flight test data. It is visually that the single inceptor input GP model and the FLIGHTLAB model struggle to accurately capture the dynamics. The GP model, in this case, again showing the bifurcation in the 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.

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

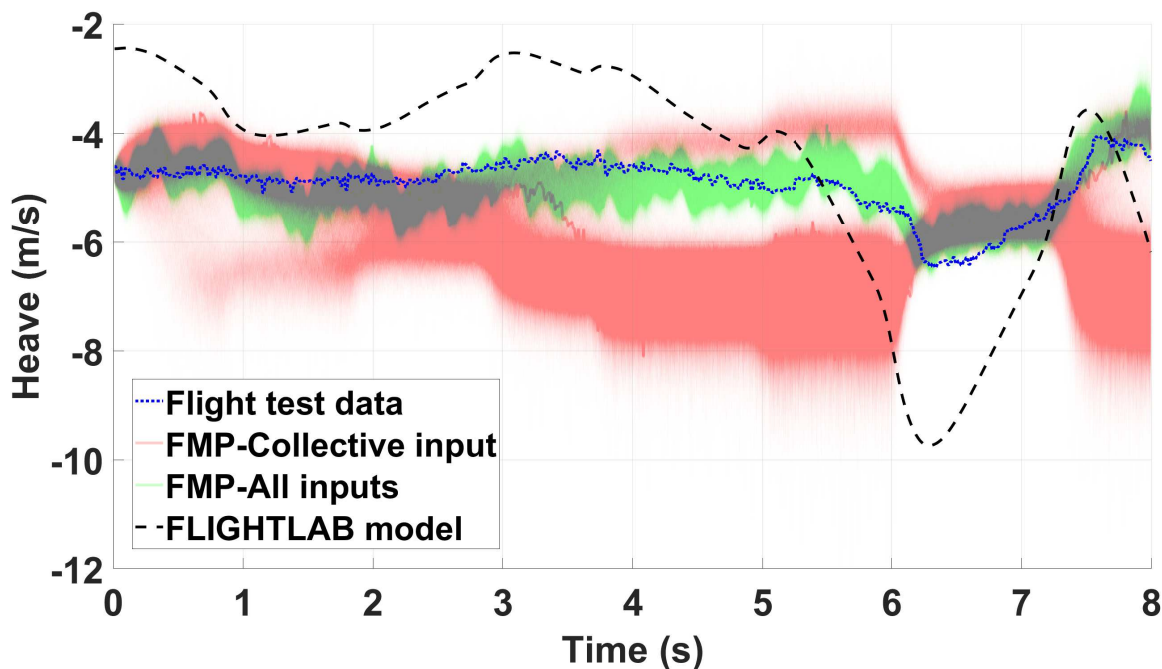


Fig. 10: Realization of the ‘full’ GP heave prediction for input structure containing the Collective input with one lagged heave (Case 14 in Table 1) and ALL inputs with one lagged heave (Case 16 in Table 1) with the comparison of the FLIGHTLAB model.

Lower bound) was used to select training points from the initial training set. In this Section, for the sake of brevity, the OSAP for the variational sparse GPs will not be shown as the predictions are very accurate, as shown in Fig. 3, Fig. 5, Fig. 7 and Fig. 9 for the ‘full’ GP.

In the current Section, only the results for the all inputs and a relevant lagged output input structure, Eq. 23 will be shown. The comparisons of Fig. 6, Fig. 8 and Fig. 10 indicate that all of the inputs are required to closely predict the roll, yaw and heave responses. The Figures are not shown; however, the RMSE values are displayed in Table 2.

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. Figure 11 shows the convergence of the lower bound for 32 training points after the initial five training points were chosen. By inspection, Figure 11 indicates that the variational sparse GP could potentially use only 10 training points to create a model that predicts similar results to the ‘full’ GP. The

selection of the number of training points for the variational sparse GP is based on the judgment of convergence of the lower bound in Fig. 11.

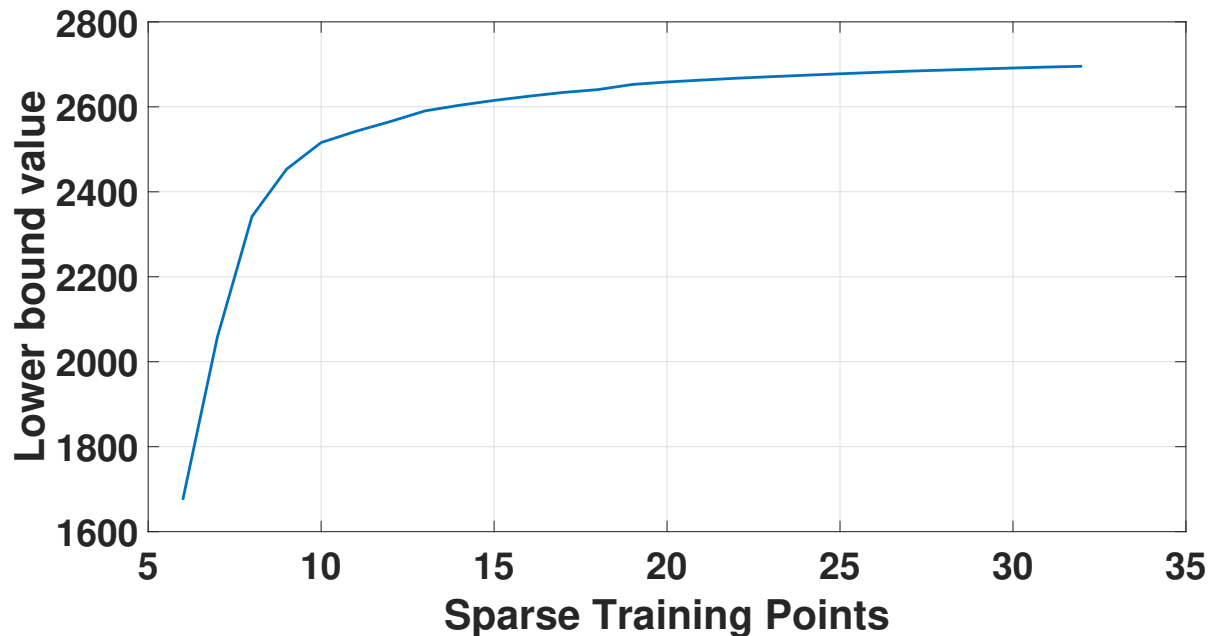


Fig. 11: Convergence of the lower bound for the sparse GP pitch rate prediction using 32 training points.

The convergence of the hyperparameters (α and β) for the variational sparse GP pitch rate predictions are shown in Fig. 12. It is clear that the hyperparameters have converged after 7 training points. However, given that the lower bound in Fig. 11 did not converge until 10 training points were used; this latter value was selected for the sparse GP model generation. To avoid repetition, in the Sections (lateral, directional and heave axes), the convergence of the lower bound and hyperparameters are not shown as the results are the same.

The sparse GP FMP predictions for pitch rate during the 3-2-1-1 maneuver, using only 10 training points, are shown in Fig. 13. 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.

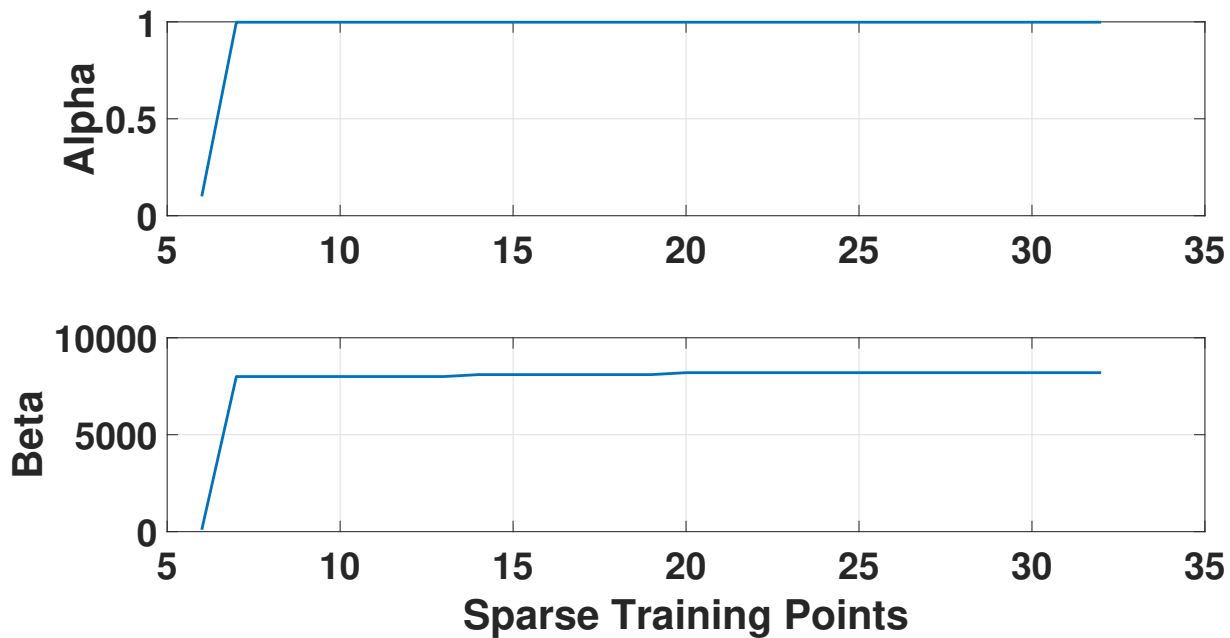


Fig. 12: Convergence of the hyperparameters (α and β) for the sparse GP pitch rate prediction using 32 training points.

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. The RMS difference in these responses can be found in Table 1.

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. 14 and 15 in a similar way to Fig. 13. As with the longitudinal axis predictions, 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 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

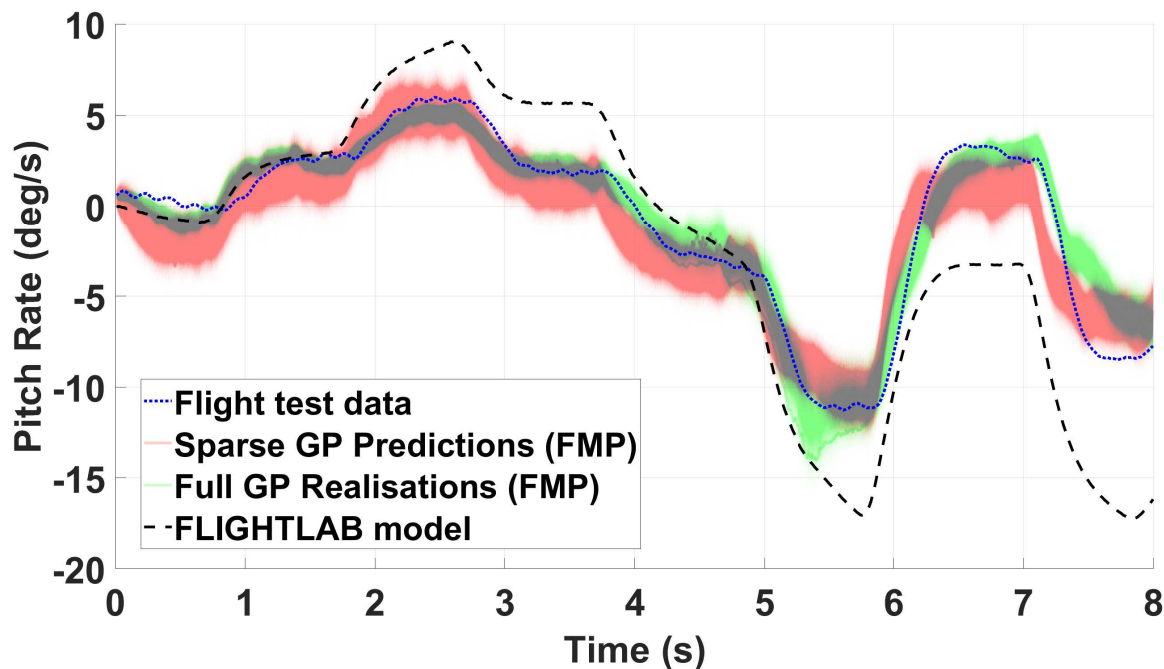


Fig. 13: Comparison of the variational sparse GP pitch rate prediction for the input structure containing ALL inputs and one lagged pitch rate using 10 training points with the Full GP using the same input structure (Case 4 in Table 1) and the FLIGHTLAB model.

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 hyperparameters and the lower bound took longer than the previous cases. With this slight modification to the analysis procedure, Fig. 16 shows the model prediction comparison for Case 16 in Table 1. As before the sparse GP model creates a prediction that is close to, but not quite as accurate as that 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.

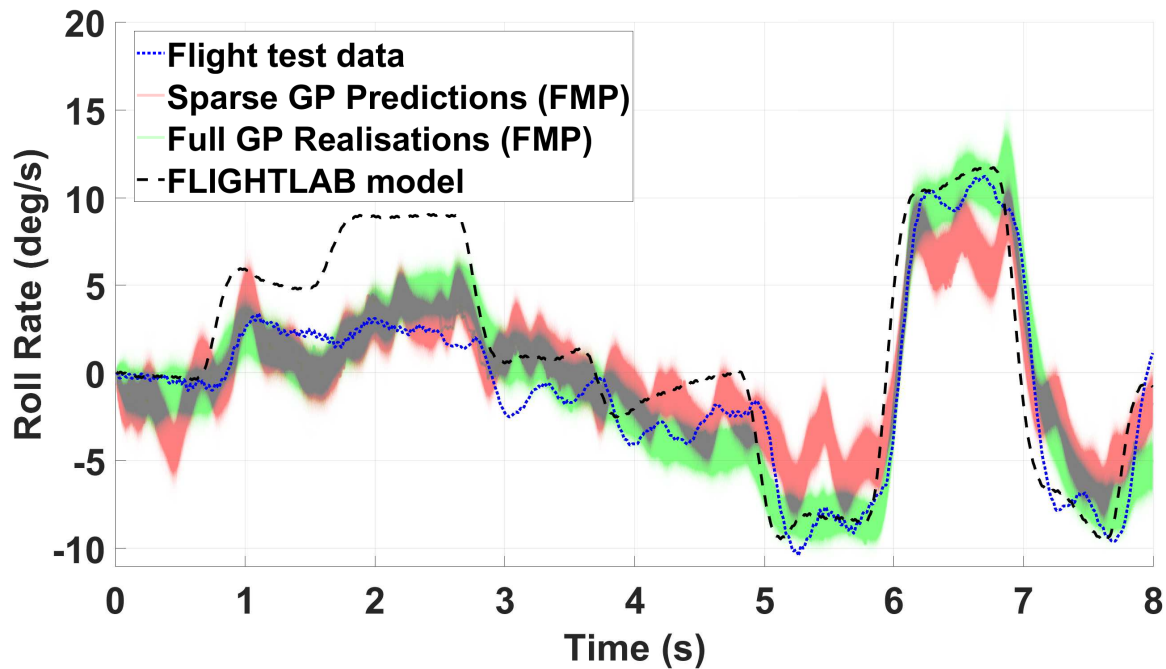


Fig. 14: Comparison of the variational sparse GP roll rate prediction for the input structure containing ALL inputs and one lagged roll rate using 10 training points with the Full GP using the same input structure (Case 8 in Table 1) and the FLIGHTLAB model.

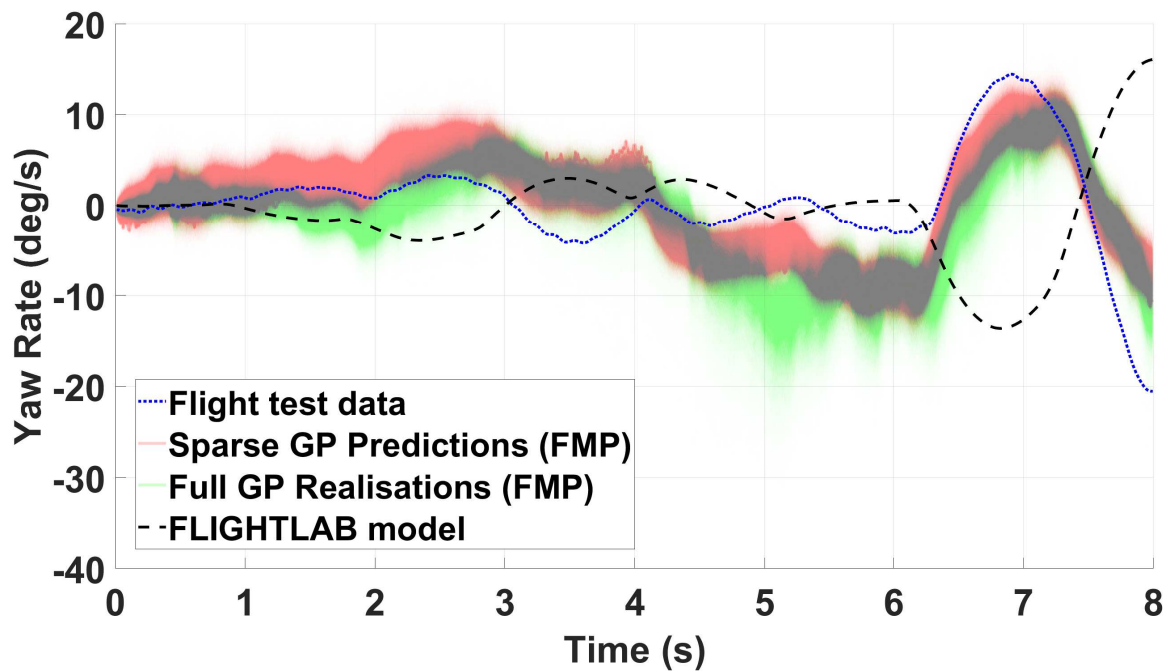


Fig. 15: Comparison of the variational sparse GP yaw rate prediction for the input structure containing ALL inputs and one lagged yaw rate using 10 training points with the Full GP using the same input structure (Case 12 in Table 1) and the FLIGHTLAB model.

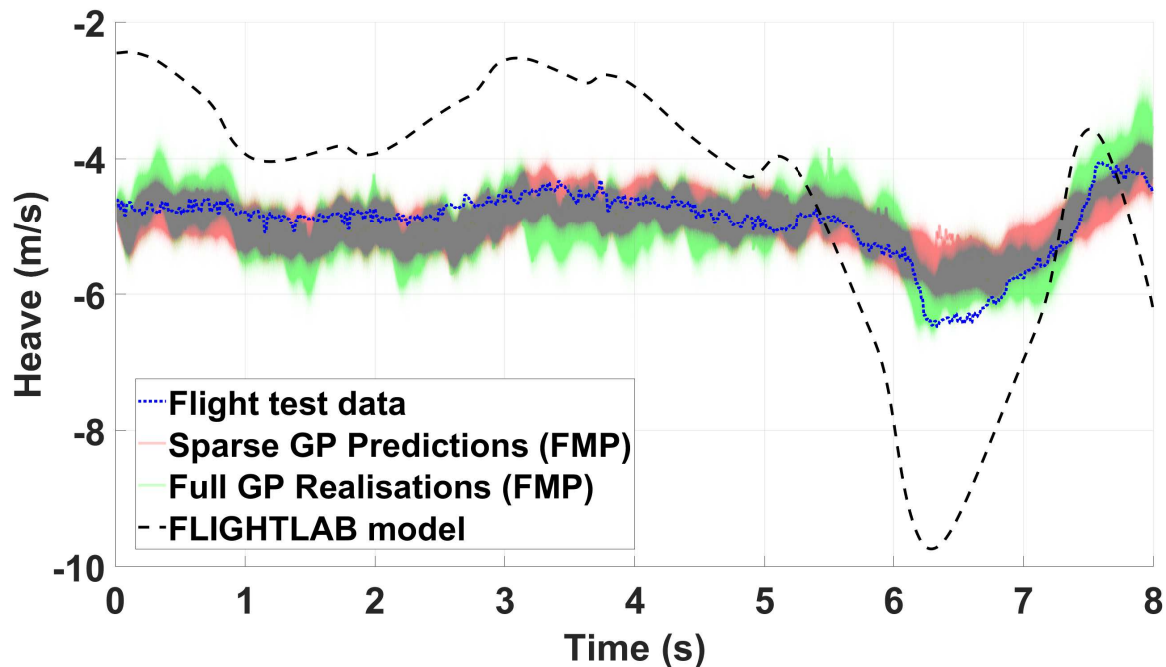


Fig. 16: Comparison of the variational sparse GP heave prediction for the input structure containing ALL inputs and one lagged heave using 15 training points with the Full GP using the same input structure (Case 16 in Table 1) and the FLIGHTLAB model.

Root Mean Squared Error (RMSE)

To summarize the error between the models and flight test data, the root mean squared error (RMSE) for all of the test cases described above are shown in Table 2. The GP input column refers to the GP input structure, where for example ‘long’ is the longitudinal stick position with one lagged pitch rate. The GP model and variational sparse GP models show improved RMSE values compared to the physical-law based FLIGHTLAB model in all cases. The variational sparse GP model for each response was initially created using 32 points; this was done to observe the lower bound and hyperparameter convergence. For example, the pitch rate response the lower bound convergence is shown in Fig. 11 and the hyperparameter convergence is in Fig. 12. Based upon engineering judgment, only 10 training points were selected for sparse GP model pitch rate, roll rate and yaw rate predictions, and 15 training points were selected for the heave corresponding heave prediction, to realize similar results to that of the ‘full’ GP. Note that for the FMP predictions, the mean of the realizations is used to calculate the RMS error.

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.

Discussion

In an attempt to assess the utility of Gaussian Process machine learning modelling techniques for helicopter flight dynamics real-time response predictions, a number of different GP models have been developed using flight test data as the basis. The use of the GP techniques allows a statistical analysis of the response to be performed to provide confidence levels in the predictions made. To be of any utility for real-time purposes, these models must also run in real time. For the sparse GP models that used 10 training points as its basis, a full set of realizations (i.e. 8 seconds of predictions) took between 0.03-0.045 seconds. For the sparse GP model that used 15 training points, the response prediction took between 0.035 - 0.05 seconds. This strongly suggests that this form of model can be used in real-time.

The first type of model developed were the 'one-step ahead' models that used the primary inceptor and one lagged response term for the axis of interest to make predictions into the future. These models were able to predict the on-axis response very accurately and with high confidence. Of course, such models are of little/no use for flight simulation purposes as they are only able to make a prediction one time-step into the future. So, whilst the results were impressive, their utility is limited in the context of this paper.

What has been termed a 'full' GP model was created by sub-sampling the flight test response data. This was due to only having one dataset for each maneuver, and using a subset of the available data allowed for interpolation between the training points. Response predictions were then made in the primary vehicle axes either by using the primary control inceptor and the primary response data or using all of the available inceptor data and the primary axis response data as inputs. Both types of model were able to make subjectively good predictions of the vehicle response. The model input structure that used the full set of control inceptors performed more accurately than the single inceptor version, as might be expected. Given the limited amount of data used to train these models, their subsequent performance is notable. The interesting point about the results, compared to a traditional flight dynamics approach, is that the prediction responses are provided with the confidence level with which the model has been able to make that prediction. This information might prove useful not only in terms of both deciding when the GP model

might take-over from or hand back to a physics-based model but also as a means to provide evidence to regulatory authorities when certifying simulation training facilities.

Noting that the ‘full’ GP models were already a sub-sample of the available data, variational sparse GP machine learning techniques were utilized to assess how well the models would perform with a reduced dataset in order to assess the efficacy of these methods where real-world volumes of data are available. For the rotational axes, very similar results were achieved when the training data was reduced to just over one-third of its original size. For the heave translational axis, the reduction was limited to one-half. Nevertheless, given the sparse data used across the model training maneuvers, the resulting response predictions were impressive.

The variational sparse GP method attempted to use fewer data points to create a similar GP response to that of the ‘full’ GP. The ‘full’ GP used 32 training points, evenly spaced over the time series, while the variational sparse GP utilized only 10 training points in the examples shown here. The results are shown in Fig. 13, Fig. 14, Fig. 15 and Fig. 16 provide evidence that a relatively small number of training points can be used to produce a similar model. This would be beneficial in terms of real-world scalability, allowing this method to be applied to larger data sets.

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. 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 [41] and [42]. One possible use envisaged for these machine-learning modelling techniques is to complement existing physics-based techniques and ‘take-over’ when the physical models become less accurate.

There are known issues with machine learning techniques. Specifically, when used correctly, they are very good at interpolating within the training data but much less so at interpolating beyond it. Similarly, they are ‘black-box’ techniques that utilize model states that are difficult or impossible to interpret in a physical sense. To the first point, there are much larger datasets available for production aircraft and so the available flight envelope of data in which to interpolate should cover at least the normal operational environment. It is acknowledged that these techniques will likely not cope well in flight regimes that have not been encountered in the training data. However, the statistical nature of the methods presented may provide a means by which erroneous response predictions can be filtered out. To the second point, even in physical-based models, depending upon the structure used, measurement data for all of the internal physical variables may not be available, reducing 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 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 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 has been assessed by comparing them to 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 fea-

ture 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

- ¹K. S. Shy, J. J. Hageman, and J. H. Le, *The Role of Aircraft Simulation in Improving Flight Safety Through Control Training*. National Aeronautics and Space Administration, Dryden Flight Research Center, 2002.
- ²H. H. Bell and W. L. Waag, “Evaluating the effectiveness of flight simulators for training combat skills: A review,” *The international journal of aviation psychology*, vol. 8, no. 3, pp. 223–242, 1998.
- ³D. Allerton, “The impact of flight simulation in aerospace,” *The Aeronautical Journal*, vol. 114, no. 1162, pp. 747–756, 2010.
- ⁴G. D. Padfield, *Helicopter flight dynamics*. John Wiley & Sons, 2008.
- ⁵A. Punjani and P. Abbeel, “Deep learning helicopter dynamics models,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 3223–3230, IEEE, 2015.
- ⁶P. L. Myers III, A. W. Starr, and K. Mullins, “Flight Simulator Fidelity, Training Transfer, and the Role of Instructors in Optimizing Learning,” *International Journal of Aviation, Aeronautics, and Aerospace*, vol. 5, no. 1, p. 6, 2018.
- ⁷Anon, “JAR-FSTD(H), Helicopter Flight Simulation Training Devices.” May 2008.
- ⁸Anon, “FAA Advisory Circular AC120-63 Helicopter Simulator Qualification.” November 1994.
- ⁹P. Perfect, E. Timson, M. D. White, G. D. Padfield, R. Erdos, and A. W. Gubbels, “A rating scale for the subjective assessment of simulation fidelity,” *The Aeronautical Journal*, vol. 118, no. 1206, pp. 953–974, 2014.
- ¹⁰P. Perfect, M. White, G. Padfield, and A. Gubbels, “Rotorcraft simulation fidelity: new methods for quantification and assessment,” *The Aeronautical Journal*, vol. 117, no. 1189, pp. 235–282, 2013.
- ¹¹M. D. White, P. Perfect, G. D. Padfield, A. W. Gubbels, and A. C. Berryman, “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, no. 4, pp. 663–686, 2013.
- ¹²E. Tobias, M. Tischler, T. Berger, and S. G. Hagerott, “Full flight-envelope simulation and piloted fi-

delity assessment of a business jet using a model stitching architecture,” in *AIAA Modeling and Simulation Technologies Conference*, p. 1594, 2015.

¹³L. Lu, G. Padfield, M. White, and P. Perfect, “Fidelity enhancement of a rotorcraft simulation model through system identification,” *The Aeronautical Journal*, vol. 115, no. 1170, pp. 453–470, 2011.

¹⁴S. Manso, “Simulation and system identification of helicopter dynamics using support vector regression,” *The Aeronautical Journal*, vol. 119, no. 1222, pp. 1541–1560, 2015.

¹⁵M. V. Kumar, S. Omkar, R. Ganguli, P. Sampath, and S. Suresh, “Identification of helicopter dynamics using recurrent neural networks and flight data,” *Journal of the American Helicopter Society*, vol. 51, no. 2, pp. 164–174, 2006.

¹⁶S. Omkar, D. Mudigere, J. Senthilnath, and M. V. Kumar, “Identification of Helicopter Dynamics based on Flight Data using Nature Inspired Techniques,” *arXiv preprint arXiv:1411.3251*, 2014.

¹⁷M. K. Titsias, “Variational model selection for sparse Gaussian process regression,” *Report, University of Manchester, UK*, 2009.

¹⁸M. Titsias, “Variational Learning of Inducing Variables in Sparse Gaussian Processes,” *Artificial Intelligence and Statistics*, pp. 5:567–574, 2009.

¹⁹Anon, “FLIGHTLAB Development Software, Advanced Rotorcraft Tehcnology Inc, Sunnydale, California.” <http://www.flightlab.com/flightlab.html>.

²⁰R. D. Jackson, M. Jump, and P. L. Green, “Towards Gaussian Process models of complex rotorcraft dynamics,” in *Vertical Flight society - Forum 74*, 2018.

²¹B. Berger, F. Rauscher, and B. Lohmann, “Analysing Gaussian processes for stationary black-box combustion engine modelling,” in *Proceedings of the 18th IFAC World Congress, Milano, Italy*, 2011.

²²T. Gutjahr, H. Kleinegräber, H. Ulmer, T. Kruse, and C. Eckstein, “New Approaches for Modeling Dynamic Engine Behavior with Gaussian Processes,” in *7th Conference on DoE in Engine Development. Berlin: Expert Verlag*, 2013.

²³B. Berger and F. Rauscher, “Robust Gaussian Process Modelling for Engine Calibration,” in *Proceedings of the 7th Vienna International Conference on Mathematical Modelling (MATHMOD)*, pp. 159–164, 2012.

- ²⁴O. Dieterich, J. Götz, M. Pavel, B. DangVu, P. Masarati, G. Quaranta, M. Gennaretti, J. Serafini, M. Jump, and L. Liu, “Rigid Body and Aeroelastic Rotorcraft-Pilot Coupling (RPC)–Prediction Tools and Means for Prevention,” *GARTEUR LIMITED*, 2008.
- ²⁵C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- ²⁶C. E. Rasmussen and C. Williams, “Gaussian processes for machine learning,” *Cited on*, pp. 1 – 31, 2006.
- ²⁷D. Higdon, J. Gattiker, B. Williams, and M. Rightley, “Computer model calibration using high-dimensional output,” *Journal of the American Statistical Association*, vol. 103, no. 482, 2008.
- ²⁸A. Girard, *Approximate methods for propagation of uncertainty with Gaussian process models*. PhD thesis, University of Glasgow, 2004.
- ²⁹W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, Vol.27, pp. 97–109, April 1970.
- ³⁰P. Green and K. Worden, “Bayesian and Markov Chain Monte Carlo methods for identifying non-linear systems in the presence of uncertainty,” *Phil. Trans. R. Soc. A*, vol. 373, no. 2051, p. 20140405, 2015.
- ³¹P. Green, “Bayesian system identification of a nonlinear dynamical system using a novel variant of simulated annealing,” *Mechanical Systems and Signal Processing*, vol. 52, pp. 133–146, 2015.
- ³²E. Snelson and Z. Ghahramani, “Sparse Gaussian processes using pseudo-inputs,” in *Advances in neural information processing systems*, pp. 1257–1264, 2006.
- ³³J. Götz, “Garteur HC AG16/WP-7 - Bo105: Flight Test Data.” April 2009.
- ³⁴G. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*. Oxford, Blackwell Science, 1996.
- ³⁵Anon, “GARTEUR HC(AG06) : Mathematical Modelling for the Prediction of Helicopter Flying Qualities.” 1996.
- ³⁶R. Heffley, J. Lehman, and R. van Winkle, “A Compilation and Analysis of Helicopter Handling Qualities Data, Volume One: Data Compilation.” 1979.
- ³⁷Anon, “GARTEUR HC(AG16) : Rigid Body and Aeroelastic Rotorcraft - Pilot Coupling - Prediction Tools and Means for Prevention.” 2009.

³⁸B. Dang Vu, “ARISTOTEL Deliverable D2.1 - Baseline Air Vehicle Model Configuration Data.” January 2011.

³⁹Anon, “FLIGHTLAB Theory Manual (Vol. One).” March 2004.

⁴⁰D. Lawrence, *The Flying Qualities of the Wright Flyers*. PhD thesis, Department of Engineering, University of Liverpool, September, 2004.

⁴¹C. Zang, H. Xin, and J. Driscoll, “Development and validation of an engineering simulation model in FLIGHTLAB with customized modeling enhancements, 9-11 may 2017,” *73rd American Helicopter Society Forum, Fort Worth, Texas, US*, May 2017.

⁴²H. Xin, C. Zhang, and J. T. Driscoll, “Enhancement of an Engineering Simulation Model to Improve the Correlation with Flight Test Data in Climb/Descent and Autorotation,” in *Proceedings of the 75th Annual Forum*, (Philadelphia), VFS International, May 2019.