Towards Gaussian Process Models of Complex Rotorcraft Dynamics

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 (also known as white box models) are widely applied in the aerospace industry, providing models for dynamic systems such as helicopter flight simulators. To meet the criteria of real-time simulation, simplifications to the underlying physics sometimes have to be applied, leading to errors in the model's predictions. Grey-box models use both physics-based and data-based models. They have potential to reduce the difference between a simulator's and real rotorcraft's response. In the current work, a preliminary step to the grey-box approach, a machine learnt data-based, i.e 'black box' model is applied to the dynamic response of a helicopter. The machine learning methods used are probabilistic and can capture uncertainties associated with the model's prediction. In the current paper, machine learning is used to create a Gaussian Process (GP) non-linear autoregressive (NARX) model that predicts pitch, roll and yaw rate. The predictions are compared to a physical law based model created using FLIGHTLAB software. The GP outperforms the FLIGHTLAB model in terms of root mean squared error, when predicting the pitch, roll and yaw rate of a Bo105 helicopter.

NOTATION

^{T}	Transpose
α	Hyperparameter
β	Precision of noise
$\boldsymbol{\Phi}$	Design Matrix
$\boldsymbol{\phi}$	Vector of 'basis functions'
$\boldsymbol{C}$	Covariance matrix with additional noise parameter, such as $\boldsymbol{C} = \boldsymbol{K} + \boldsymbol{I}\beta^{-1}$
$\boldsymbol{C}_{N+1}$	$(N+1) \times (N+1)$ covariance matrix
$\boldsymbol{f}$	Vector of latent function
$\boldsymbol{I}$	Identity Matrix
$\boldsymbol{K}$	Covariance Matrix (Noise Free)
$\boldsymbol{k}$	Vector, short for $\boldsymbol{k}(\boldsymbol{x}_n, \boldsymbol{x}^*)$
$\boldsymbol{w}$	Vector of parameters
$\boldsymbol{x}$	Input Vector
$\boldsymbol{x}^*$	New Input Vector
$\boldsymbol{y}$	Vector of training observations $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]^T$
$\boldsymbol{y}^*$	Vector of training observations including new prediction $\boldsymbol{y} = [y_1, y_2, \ldots, y_N, y^*]^T$
δ^o	Collective lever
δ^p	Pedal position
δ^x	Longitudinal stick position
δ^y	Lateral stick position
ε	Noise term goverened by $\varepsilon \sim \mathcal{N}(\varepsilon \mid 0, \beta^{-1})$
γ	Precision of the distribution $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, \gamma^{-1}\boldsymbol{I}$
$\mathbb{E}$	Expected value
GP	Gaussian Process
$\mathcal{N}$	Normal Distribution
μ	Mean
σ^2	Variance
cov	Covariance
θ	Hyperparameters to be optimised, α and β
c	Short for $\beta^{-1} + k(\boldsymbol{x}^*, \boldsymbol{x}^*)$
C_{nm}	Elements of Covariance matrix ($\boldsymbol{C}$)
f	Latent function values
I_{nm}	Elements of identity matrix ($\boldsymbol{I}$)
k	Kernel Function

Presented at the AHS International 74th Annual Forum & Technology Display, Phoenix, Arizona, USA, May 14–17, 2018.

n	Time step
R	Number of Monte Carlo runs
v	Generic system input
y^*	Prediction from the Gaussian Process given $\boldsymbol{x}^*$

INTRODUCTION

Flight simulators are a vital part of the aircraft life cycle. They are used in design and development phases, testing and qualification activities as well as in training and research (Ref. 1, 2). Due to their availability and cost compared to the corresponding real actual aircraft, their use continues to increase. Simulators can also help to address the increased demand for new pilots, who are needed to replace the current ageing population of pilots. Moreover, the military is increasing the use of simulators for mission rehearsal in land, sea and air contexts.

The heart of any flight simulation facility is the flight dynamics model. Techniques to design and develop such models are well known and documented (Ref. 3, 4). However, there is a requirement for the entire simulation system to run in real-time and this can lead to simplifications to the underlying physics having to be made, particularly for more complex aircraft such as rotorcraft. These simplifications that are applied mean that the flight model cannot necessarily capture all of the complex dynamics that would be present during the equivalent real scenario. They can lead to significant differences between the model and the real aircraft. These differences can, in the worst case, have a negative impact, for example, on training for the crew using the simulator.

The quality of the flight dynamics model speaks to the 'fidelity' of the simulation device. The 'engineering fidelity' of such a device is typically measured using a series of quantitative requirements contained within simulator qualification documents such as (Ref. 5, 6). 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 seek methods that can better meet this need (Ref. 7), this 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 accurately represent the situation.

The aim, then, of the research presented in this paper, was to (start to) develop simulation methods that can more accurately capture the complex dynamics of rotorcraft, while still being able to be run in real-time. The research is based on the hypothesis that current flight dynamics models can be improved using machine learning; data-based models that, once trained, can predict the *model error* [1] that is present in current simulators. This approach aims to generate a 'grey box' model that combines physical-law based simulations (i.e. 'white box') with data based simulations (i.e. 'black box'). The 'grey box' approach ensures that the 'black box' (machine-learnt) model will step-in only when the simplified physical-law model deviates from the ideal. In the current study, as a preliminary step towards this goal, the authors aim to develop and validate a black box model using machine learning methods that can emulate the dynamic behaviour of a rotorcraft. Crucially, the methods used are probabilistic and are therefore able to capture the uncertainties associated with such an approach.

In this paper, machine learning is used to create a Gaussian process (GP) non-linear autoregressive model that predicts pitch, roll and yaw rate of the Bo105 helicopter trained only upon a longitudinal cyclic input. The autoregressive model makes a prediction using the longitudinal cyclic position as well as previous observations of the relevant output.

The Gaussian process model for pitch, roll and yaw rate are compared to physical law based models, which are implemented using Advanced Rotorcraft Technology's (ART) FLIGHTLAB software (Ref. 9). The comparison provides an excellent basis for future work where our approach will predict *model error*.

NUMERICAL METHODS

Gaussian Processes

Gaussian Processes (GPs) have been widely used in recent years for many different applications. In the current work, GPs are used to perform regression (they can also perform other tasks such as classification (Ref. 10)). An advantageous property of GPs is that they can be used to quantify the uncertainties in one's predictions which, here, produces a worst case scenario given the model uncertainty. This property of being able to quantify errors is useful. For example, in the current context whereby the authors wish to extend the work to predict model error, GPs could be used to quantify the uncertainties involved in capturing the discrepancies between physical-law based simulators and reality. Another beneficial 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 helicopter dynamics.

Linear Regression To aid the understanding of the Gaussian process, consider a model which is defined as a linear combination of fixed basis functions (Ref. 10, 11):

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

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{w}$ is a vector of parameters to be identified, $\boldsymbol{\phi}$ is a vector of 'basis functions' and f is a latent function. One can choose the prior distribution over $\boldsymbol{w}$ to be:

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

[1] *Model error* captures the inaccuracy of the physical law based model compared to that of the flight test data. For more information on *model error* see the study by Kennedy and O'Hagan (Ref. 8)

where γ is the precision of the distribution, and $\boldsymbol{I}$ is an identity matrix. The function values are given by vector $\boldsymbol{f}$:

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

where N is the number of of training points. Using equation (1), the vector $\boldsymbol{f}$ is given by:

$$\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{w} \tag{4}$$

where $\boldsymbol{\Phi}$ is a design matrix (for more information see Bishop (Ref. 10)) . Given the prior (equation (2)), one can then show that $\boldsymbol{f}$ is Gaussian, with mean and covariance matrix:

$$\mathbb{E}[\boldsymbol{f}] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}] = \boldsymbol{0} \tag{5}$$

$$\text{cov}[\boldsymbol{f}] = \mathbb{E}[\boldsymbol{f}\boldsymbol{f}^T] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T]\boldsymbol{\Phi}^T = \frac{1}{\beta}\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{K} \tag{6}$$

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

$$K_{nm} = \frac{1}{\gamma}\phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_m). \tag{7}$$

where $n, m = 1, \ldots, N$.

Gaussian Process for regression Regression problems often include noise on the observed training data. To apply GPs to regression the defined training data should also include noise:

$$y_n = f_n + \varepsilon_n \tag{8}$$

where y_n represents the nth observation of the system's response, $f_n = f(x_n)$ and $\varepsilon_n \sim \mathcal{N}(\varepsilon_n \mid 0, \beta^{-1})$, where β is the precision of noise. Instead of defining basis functions, as in equation (1), with a GP one can simply define a prior of the form:

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{0}, \boldsymbol{K}) \tag{9}$$

where the covariance matrix, $\boldsymbol{K}$ is given by:

$$\boldsymbol{K} = \begin{pmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & k(\boldsymbol{x}_2,\boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_N,\boldsymbol{x}_1) \\ k(\boldsymbol{x}_1,\boldsymbol{x}_2) & k(\boldsymbol{x}_2,\boldsymbol{x}_2) & \ldots & k(\boldsymbol{x}_N,\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ k(\boldsymbol{x}_1,\boldsymbol{x}_N) & k(\boldsymbol{x}_2,\boldsymbol{x}_N) & \ldots & k(\boldsymbol{x}_N,\boldsymbol{x}_N) \end{pmatrix}$$

which is produced by a user-defined kernel function (k). An example of such a kernel function is the squared exponential (Ref. 10):

$$\boldsymbol{K}_{nm} = k(\boldsymbol{x}_n,\boldsymbol{x}_m) = \exp\left(-\tfrac{\alpha}{2}(\boldsymbol{x}_n - \boldsymbol{x}_m)^T(\boldsymbol{x}_n - \boldsymbol{x}_m)\right) \tag{10}$$

where the 'hyperparameter' (α), induces correlations that depend on the 'closeness' of $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$. Choosing different hyperparameters can affect how accurate the GP model is. To achieve 'optimum' values for the hyperparameters, an optimisation technique is required (these techniques are discussed in a later Section).

The distribution over $\boldsymbol{y}$, conditional on $\boldsymbol{f}$, is given by:

$$p(\boldsymbol{y} \mid \boldsymbol{f}) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}, \beta^{-1}\boldsymbol{I}). \tag{11}$$

The marginal distribution of $\boldsymbol{y}$ is defined as:

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y} \mid \boldsymbol{f})p(\boldsymbol{f})d\boldsymbol{f} \tag{12}$$

which, given equations (9) and (11), allows us to write

$$p(\boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{0}, \boldsymbol{C}) \tag{13}$$

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

$$C_{nm} = C(\boldsymbol{x}_n,\boldsymbol{x}_m) = k(\boldsymbol{x}_n,\boldsymbol{x}_m) + \beta^{-1}I_{nm} \tag{14}$$

where I_{nm} is an element of identity matrix $\boldsymbol{I}$.

Prediction The main aim of GP regression is to make predictions for new input data that are not in the training data. Given a new input vector, $\boldsymbol{x}^*$, one can then estimate the probability of a new point y^* given previous observations $\boldsymbol{y}$. To find y^* the predictive distribution is evaluated, which is given by $p(y^* \mid \boldsymbol{y})$. Defining $\boldsymbol{y}^* = [y_1, y_2, \ldots, y_N, y^*]^T$, the joint distribution over $\boldsymbol{y}^*$ is

$$p(\boldsymbol{y}^*) = \mathcal{N}(\boldsymbol{y}^* \mid \boldsymbol{0}, \boldsymbol{C}_{N+1}) \tag{15}$$

where $\boldsymbol{C}_{N+1}$ is a $(N+1) \times (N+1)$ covariance matrix. $\boldsymbol{C}_{N+1}$ can be shown to be:

$$\boldsymbol{C}_{N+1} = \begin{bmatrix} \boldsymbol{C}_N & \boldsymbol{k} \\ \boldsymbol{k}^T & c \end{bmatrix} \tag{16}$$

where:

$$\boldsymbol{k} = k(\boldsymbol{x}_n,\boldsymbol{x}^*), \qquad n = 1, \ldots, N \tag{17}$$

and

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

The mean and variance of y^* given $\boldsymbol{y}$ can be shown to be (Ref. 10):

$$\mu(y^* \mid \boldsymbol{y}) = \boldsymbol{k}^T\boldsymbol{C}_N^{-1}\boldsymbol{y}$$

$$\sigma^2(y^* \mid \boldsymbol{y}) = c - \boldsymbol{k}^T\boldsymbol{C}_N\boldsymbol{k}$$

such that:

$$p(y^* \mid \boldsymbol{y}) = \mathcal{N}(y^* \mid \mu, \sigma^2) \tag{19}$$

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

$$y^* \sim \mathbf{GP}(\boldsymbol{x}^*) \tag{20}$$

Choice of Kernel

The kernel chosen for the current study was used by Higdon et al (Ref. 12). The kernel takes the form of:

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \prod_i^N \alpha^{(4(\boldsymbol{x}_n^i - \boldsymbol{x}_m^i)^2)} \tag{21}$$

where $\boldsymbol{x}^i$ is used to represent the ith element of the vector $\boldsymbol{x}$. The kernel of equation (21) allows the hyperparameter (α) to always be between zero and one. This is beneficial when using a property of GPs that allows the relevance of inputs to be determined (known as 'Automatic Relevance Determination'). The relevance property is not explored in the present paper, however it is of future interest for the authors.

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 format:

$$y_n = f\left(y_{n-1}, y_{n-2}, y_{n-3}, \ldots, v_n, v_{n-1}, v_{n-2}, v_{n-3}, \ldots\right) + \varepsilon_n \tag{22}$$

where v is a generic system input and, as before, y represents system observations, n is the time step and f is the function that we wish to model. The NARX structure uses information from 'lagged' terms (observations and inputs) and the current input to help predict y_n. The NARX structure has been used as the input structure when modelling non-linear dynamical system such as the Duffing oscillator (Ref. 13, 14). It has also been used to develop models of helicopter dynamics (Ref. 15, 16, 17).

The helicopter investigated in the current paper is treated as having four inputs; longitudinal stick position (δ^x), lateral stick position (δ^y), pedal position (δ^p) and collective lever (δ^o). The outputs correspond to the helicopter's three axes: pitch rate, roll rate and yaw rate. The NARX structure is required to capture the behaviour of the helicopter's dynamics (Ref. 15, 16, 17). Considering an example where the goal is to map the longitudinal stick position to pitch rate the NARX input structure for the GP, is:

$$\boldsymbol{x}_n = \begin{Bmatrix} \delta_n^x \\ y_{n-1} \end{Bmatrix} \tag{23}$$

where y is the relevant observation (pitch, roll or yaw rate).
When using the GP NARX model the kernel from equation (21) takes the form:

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \alpha^{4(\delta_n^x - \delta_m^x)^2} \times \alpha^{4(y_{n-1} - y_{m-1})^2}. \tag{24}$$

In the following, the inputs and outputs were normalised to ensure that all values are between 0 and 1. The model generated here will be used to produce what is known as 'one-step-at-a-time 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. Using **GP** to represent the prediction made by a Gaussian Process then, in our specific case, a OSAP is defined as:

$$y_n^* = \mathbf{GP}\left(\delta_n^x, y_{n-1}\right) \tag{25}$$

where y_{n-1} is the previous observation of the relevant quantity (i.e. roll, pitch or yaw). During '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 utilise previous predictions as part of the model input. To illustrate this, we consider the situation where a single prediction, at time n, has already been made according to equation (25). Following this, predictions of y_{n+1}^* would be realised according to

$$y_{n+1}^* = \mathbf{GP}\left(\delta_{n+1}^x, y_n^*\right). \tag{26}$$

The key aspect to note with regard to equation (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 sample from taken from the GP.
For a more detailed algorithm for FMP, see the thesis by Girard (Ref. 18).

Optimisation of hyperparameters for Gaussian Processes training

The GP process requires the optimisation of hyperparameters to create an optimal model. To facilitate this, one can generate

samples from the posterior parameter distribution, which is given by Bayes' rule:

$$p(\theta \mid \mathbf{y}) \propto p(\mathbf{y} \mid \theta)p(\theta) \tag{27}$$

where $\theta = [\alpha, \beta]^T$ .The techniques used in the current paper are discussed in the next section.

Markov Chain Monte Carlo (MCMC) MCMC can be used to generate samples from the posterior hyperparameter distribution given in equation (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. Utilising MCMC to generate samples from the posterior hyperparameter distribution is of interest for the authors. This enables the incorporation of hyperparameter uncertainty into the GP prediction.

The best known MCMC method is the Metropolis algorithm (MA) (Ref. 10). The target distribution, $\pi(\theta)$, is equal to $p(\theta \mid \mathbf{y})$. The first step in each iteration of MA is to propose a new state θ', where the current state of the Markov chain is $\theta^{(r)}$ (Ref. 19). The proposal is taken from a probability density function (PDF) $q(\theta' \mid \theta^{(r)})$ which is conditional on the current state. The proposal is accepted as the new state of the Markov chain with probability:

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

where $\pi^*(\theta)$ is the un-normalised target distribution. If accepted, the new state of the Markov chain is $\theta^{(r+1)} = \theta'$, otherwise $\theta^{(r+1)} = \theta^{(r)}$. 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 is that the method slowly increases the influence of the likelihood, via a variable ζ (Ref. 20). Increasing the influence of the likelihood increases the influence of the data and this assists algorithm convergence. As with the MA, a prior is chosen to generate a candidate hyperparameter $\theta^{(s)}$, a small initial value of ζ is also chosen. 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 constant change in Shannon entropy of the target distribution (Ref. 20). The optimisation stops when ζ reaches one, in which case the whole likelihood has been introduced.

The advantage of using this method over the standard MA is that is it easier to tune and often demonstrates better convergence, as the proposal width is being updated after each run of the MA.

Propagating hyperparameter uncertainty MCMC creates samples of the hyperparameters (θ). The GP also has uncertainty associated with its predictions, which can be combined with the hyperparameter uncertainty. Including both sources of uncertainty, the uncertainty in one's predictions can be written as:

$$y^* \sim \mathcal{N}(\mu(\theta), \sigma^2(\theta)) \tag{29}$$

where:

$$\theta \sim p(\theta \mid \mathbf{y}). \tag{30}$$

Sampling from equation (29), using samples from equation (30), is known as 'Ancestral sampling' where the posterior samples are generated using MCMC. For more information on Ancestral sample see Bishop (Ref. 10).

RESULTS

In this Section, the 3-axis rotational responses of the FLIGHTLAB Bo105 model to a longitudinal 3-2-1-1 cyclic input are compared with the GP model's output. Both are compared with data from equivalent Bo105 flight test data. The flight test control input from the Bo105 helicopter has been applied to the physical-law based FLIGHTLAB model. The input data is shown in Figure 1:

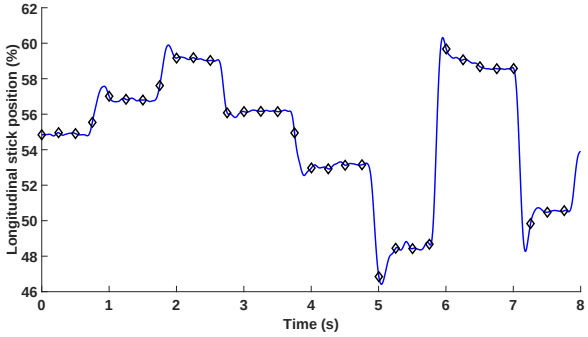

Fig. 1: 3-2-1-1 Longitudinal stick input

Using the theory, cited in the previous Section, a GP model was also created to predict the Bo105 pitch, roll and yaw rates from the 3-2-1-1 longitudinal cyclic input. We note that, here, the GP models do not use all of the available training data.. The GPs only use 32 'training' points from Figure 1, which are denoted by diamonds. All three GP models (pitch, roll and yaw) are compared to the Bo105 flight model predictions and the flight test data in the following sections.

Pitch Rate

The pitch rate response is investigated first. Figure 2 shows the OSAP predictions made by the GP. As was discussed previously, it would be expected that the predictions are very close the Bo105 flight test data. The RMS error between the GP model and the Flight Test data is **0.0012**.

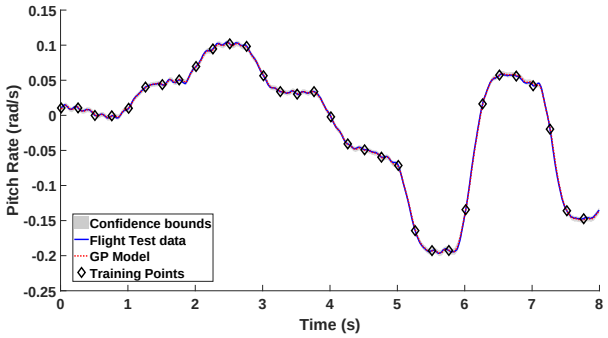

Fig. 2: Gaussian process one step ahead pitch rate predictions using hyperparameters located by simulated annealing

Figure 3 shows the predicted helicopter response from the FMP. This is a better test of the GP model. It is clear to see that the predictions are not as accurate as for the OSAP model. However, this is to be expected given that, for this model, the previous prediction becomes an input to the next prediction and, hence, predictive uncertainty is carried through the simulation.

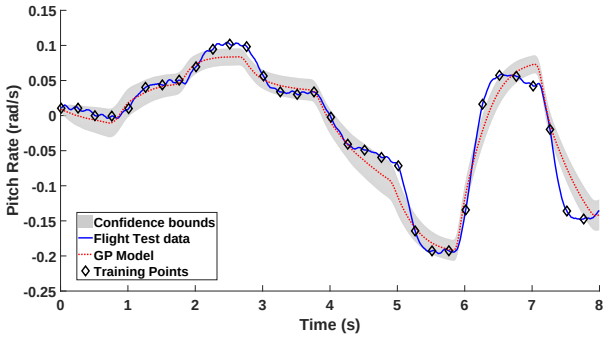

Fig. 3: Gaussian process full model pitch rate predictions using hyperparameters located by simulated annealing

Note that, in Figures 2 and 3, hyperparameter uncertainty is not considered. The hyperparameter and the full model predictions uncertainty are included in Figure 4. Note that between Figure 3 and 4 that the confidence bounds are very similar. This implies that the hyperparameter uncertainty does not have a significant affect compared to the use of the FMP. To calculate hyperparameter uncertainty, MCMC samples are required. The posterior hyperparameter samples for the GP predicting pitch rate are shown in Figure 6.

It can be seen in Figures 3 and 4 that the confidence bounds do not encompass all of the flight test data. It is currently unclear why this occurs, therefore it is of future interest for the authors. One reason could be, when using Monte Carlo samples for the FMP, the mean and variance are taken from a mixture of Gaussian's (Ref. 18). The confidence bounds are calculated for three standard deviations. The belief that 99.7% of the data falls within the three standard deviations is based on the assumptions that the predictions have come from a Gaussian. In the case of FMP, the predictions have come from a mixture of Gaussian's.. Another possible reason is that the GP assume the likelihood is also a Gaussian, this may not be the case for a real system such as the Bo105 helicopter. For more information on non Gaussian likelihoods see the study by Saul et al. (Ref. 21).

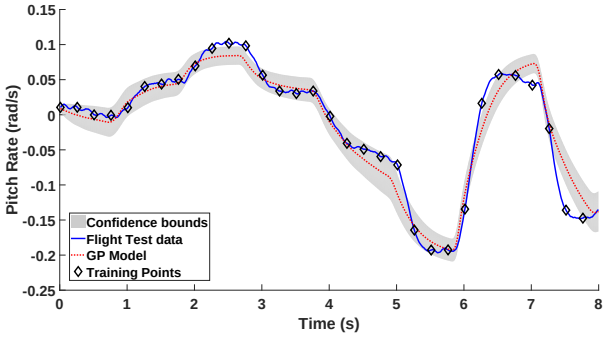

Fig. 4: Gaussian process full model pitch rate predictions with incorporated hyperparameter uncertainty from the simulated annealing results

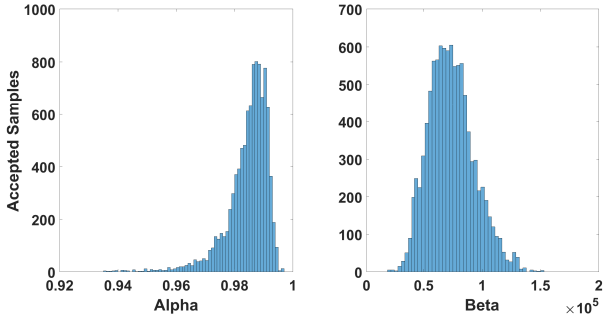

Fig. 6: Pitch rate Simulated annealing results

The comparison of the flight test data, FLIGHTLAB and GP models, are shown in Figure 5. Both models capture the essence of the real aircraft response. However, it is apparent that the GP model outperforms the FLIGHTLAB model in this case; in general, the GP predictions remain much closer to the flight test truth data throughout the maneuver but particularly in its latter stages. To try to quantify this improvement, the root mean squared errors between prediction and truth data for both models were computed (RMSE). The results of this exercise are given in Table 1. These confirm what is easily observable in the Figure; that, for the primary axis response, the GP model outperforms the FLIGHTLAB model.

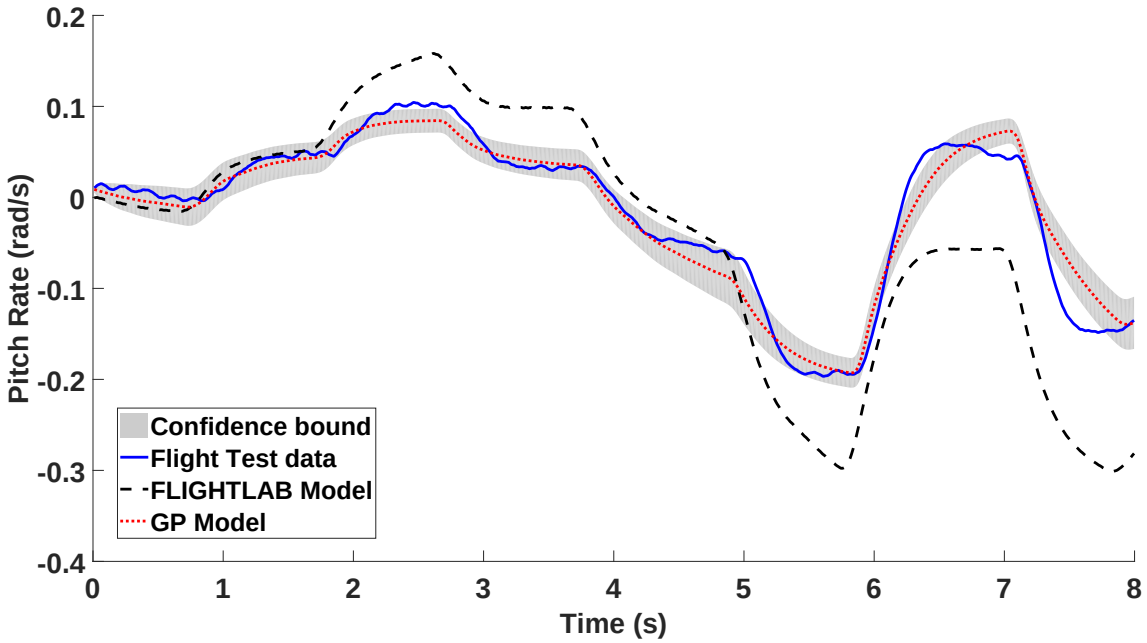

Fig. 5: Comparison of the pitch rate response of the real helicopter to the FLIGHTLAB and GP model

Prediction of pitch rate, corresponding to Figure 5	
Model	**RMSE**
Gaussian Process	0.019
FLIGHTLAB Model	0.074

Table 1: RMSE of the predictions of pitch rate for the GP and FLIGHTLAB models compared to the flight test data of the Bo105.

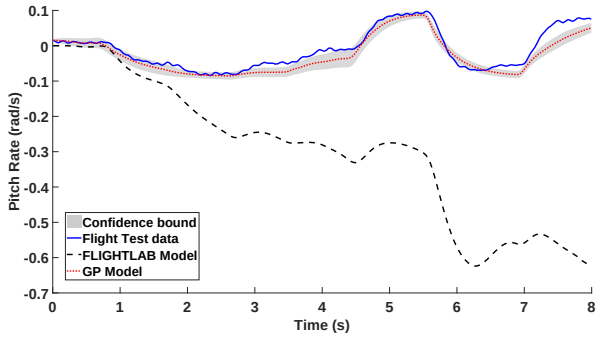

Fig. 7: Gaussian process full model pitch rate predictions with 'unseen' data using the incorporated hyperparameter uncertainty from the simulated annealing results and including a comparison between the FLIGHTLAB, GP model and the flight test data

The GP was trained using only a subset of the training data. In the previous figures, the GP is interpolating between points in the training set. This could justify why the GP in Figure 2 shows a very accurate fit. A more challenging test for the GP model is to predict data that has not been 'seen' by it before i.e. data that it has not been trained on. Figure 7 shows the GP-predicted response to another 3-2-1-1 flight test maneuver. The GP was trained on a 3-2-1-1 maneuver where the initial longitudinal cyclic input was in a positive input direction. The next, validation case in Figure 7 is the opposite i.e. the initial longitudinal cyclic input was in the negative direction. As such, this prediction is based upon previously 'unseen' data.

In a qualitative sense, Figure 7 shows a reasonably good match between the GP model prediction and the flight test truth data. The global features of the response are captured. It is arguably less good than the prediction of Figure 4 and does not capture some of the higher frequency behaviour observable in the flight test data.

Table 2 shows the RMS error for the FLIGHTLAB and GP data to the 'unseen' validation data. The GP model is not as accurate in terms of RMS error compared to Table 1, however this is expected as this is data that the GP has not been trained on. The GP model far outperforms the FLIGHTLAB model.

Prediction of pitch rate, corresponding to Figure 7	
Model	**RMSE**
Gaussian Process	0.022
FLIGHTLAB Model	0.357

Table 2: RMSE of the predictions of the validation 'unseen' pitch rate for the GP and FLIGHTLAB models compared to the flight test data of the Bo105.

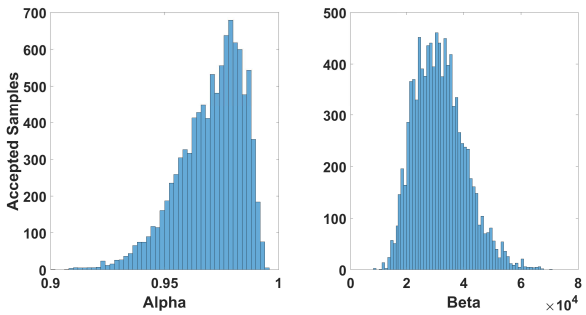

Fig. 9: Roll rate Simulated annealing results

Roll Rate

The helicopter roll response to a longitudinal cyclic input is an off-axis response and is known to be hard to capture for real-time simulation using existing physics-based modelling techniques. For the sake of brevity, the OSAP and FMP prediction without HP uncertainty are not shown for roll (and yaw) rate. The conclusions for these are the same as those drawn for the pitch rate predictions. Figure 8 shows the prediction of roll rate response made by GPs FMP with hyperparameter uncertainty included. The main character of the response is captured, particularly towards the end of the maneuver. The wider GP confidence bounds compared to the pitch rate response are indicative a reduced confidence of the model in its predictions; presumably a function of the off-axis nature of the response. It is interesting to note, however, that the fight test data does lie, to a greater extent within those confidence bounds.

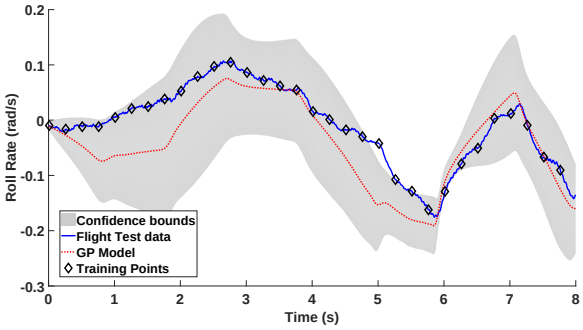

Fig. 8: Gaussian process full model roll rate predictions with incorporated hyperparameter uncertainty from the simulated annealing results

Figure 9 shows the accepted samples for the hyperparameters using simulated annealing for the prediction of roll rate. The accepted samples are used to generate the additional hyperparameters uncertainty in the FMP.

Figure 10 shows the comparison of the GP predictions of roll rate to that of the FLIGHTLAB model and the Bo105 helicopter data. The FLIGHTLAB model captures the initial part of the response very well but then becomes less accurate as the maneuver progresses. Conversely, the GP predictions do less well at the start of the maneuver but become increasingly accurate towards the final stages. The RMS error analysis of the maneuver indicates that, on average, the GP outperforms the FLIGHTLAB model. These data are shown in Table 3. This shows that RMS error analysis, while of some value, is somewhat of a blunt tool in this case as it does not capture the nuances of the response predictions as the maneuver progresses.

Prediction of roll rate, corresponding to Figure 10	
Model	**RMSE**
Gaussian Process	0.047
FLIGHTLAB Model	0.059

Table 3: RMSE of the predictions of roll rate for the GP and FLIGHTLAB models compared to the flight test data of the Bo105.

Yaw Rate

Like roll rate, the yaw rate induced by a longitudinal cyclic input is an off-axis response and this, again, can be difficult to model accurately using physics-based modelling techniques. The predicted yaw rate response of the Bo105 for FMP for the GP with hyperparameters is given in Figure 11. It is clear that the GP struggles to make an accurate prediction, with a significant portion of the maneuver being predicted to be in the opposite direction to the flight test data. Once again, however, much of the flight test response is encompassed within this uncertainty.

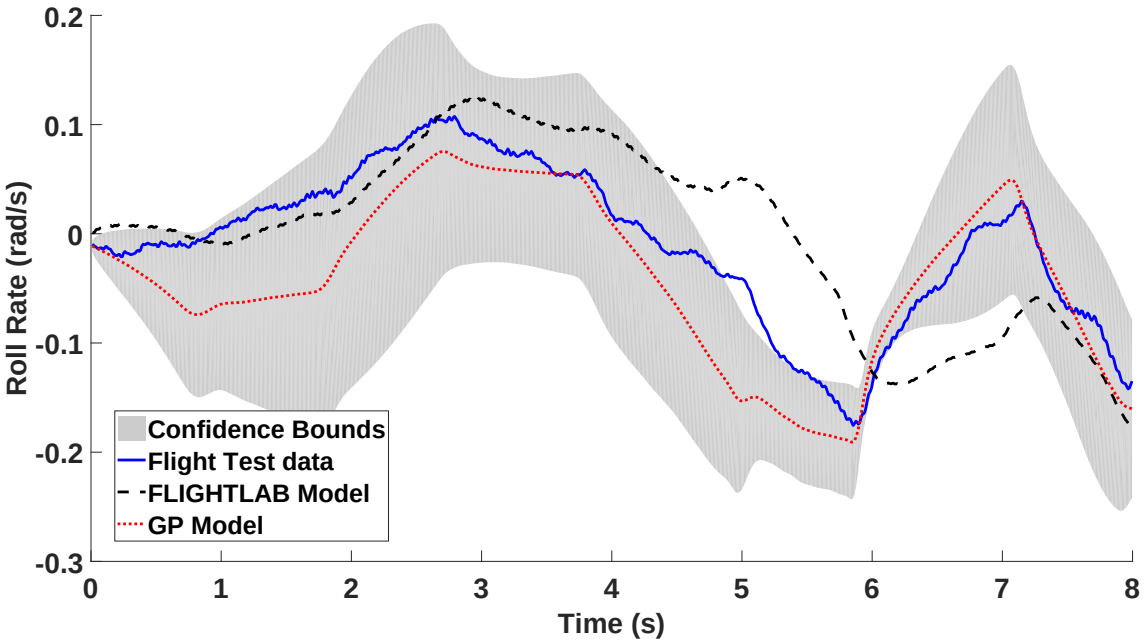

Fig. 10: Comparison of the roll rate response of the real helicopter to the FLIGHTLAB and GP model

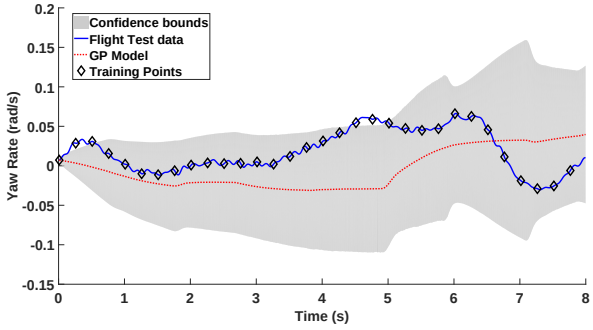

Fig. 11: Gaussian process full model yaw rate predictions with incorporated hyperparameter uncertainty from the simulated annealing results

The accepted samples of the GP model for yaw rate are shown in Figure 9. The accepted samples are used in the hyperparameter uncertainty.

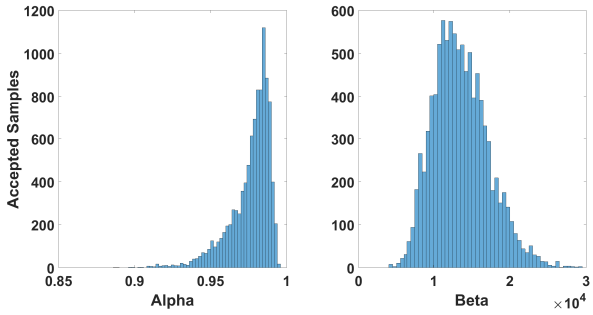

Fig. 12: Yaw rate Simulated annealing results

The comparison of the GP model for the prediction of yaw rate to the FLIGHTLAB model and the Bo105 helicopter data is shown in Figure 13. Both models make quite poor predictions overall with the FLIGHTLAB model over predicting the yaw response quite significantly in the later stages of the maneuver. This is confirmed in the RMSE values shown in Table 4. The GP has a slightly smaller value of RMS error.

Prediction of yaw rate, corresponding to Figure 13	
Model	**RMSE**
Gaussian Process	0.043
FLIGHTLAB Model	0.052

Table 4: RMSE of the predictions of yaw rate for the GP and FLIGHTLAB models compared to the flight test data of the Bo105.

DISCUSSION

This investigation has presented some early results relating to the potential use of Gaussian Process models to the application of real-time helicopter response prediction. The on-axis predictions of Figures 5 and 7 show promise for the methods. The method works well on the data that it was trained on, as might be expected. However, it also works well on unseen data. The slightly surprising result is how well the GP model works for off-axis response rates. In this regard, the GP model performed no less well than the physics-based FLIGHTLAB model and, in a global sense at least, introduced a lower overall magnitude of error into the response. That said, the GP models do not yet predict the finer points of the responses

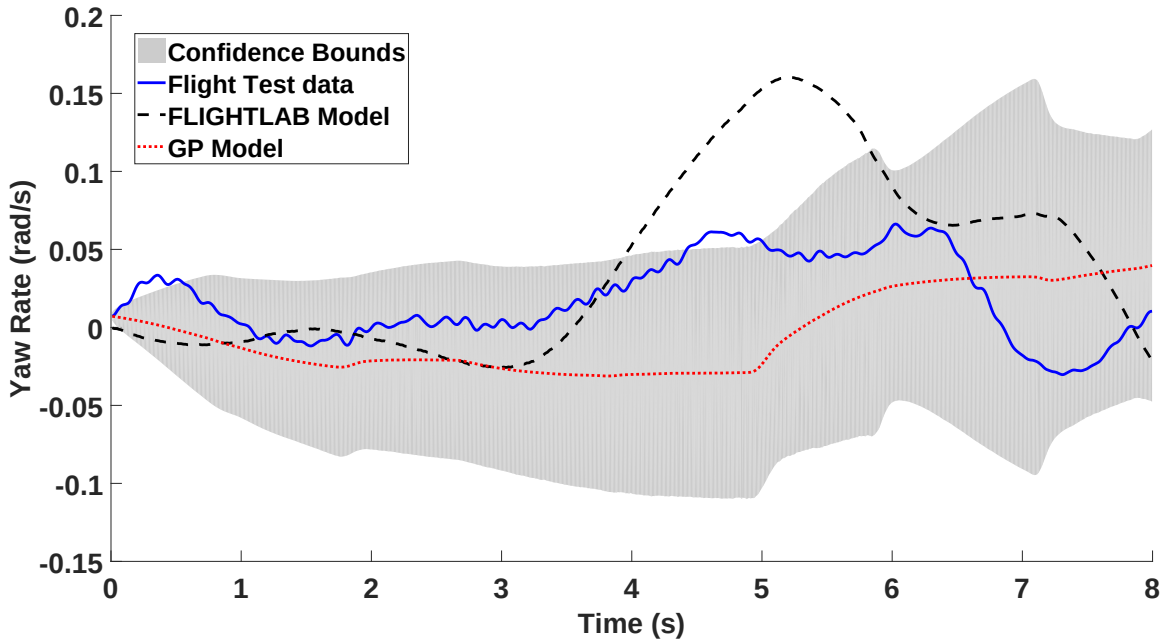

Fig. 13: Comparison of the yaw rate response of the real helicopter to the FLIGHTLAB and GP model

shown in the flight test data. It is hoped that this might be improved when the models are trained using lateral and rudder pedal response data.

Where this method shows promise is in its ability to indicate estimates of the uncertainty in its predictions. A key advantage over GPs relative to other methods, is that GPs can provide confidence bounds on their predictions. It can be seen from Figures 5 and 13 that the level of confidence in the pitch rate prediction is higher than the yaw rate, for example, as might be expected from the source of the training data. This ability to provide the degree of uncertainty might, in the longer term, be able to inform differences in, for example, pilot opinion or workload/handling qualities rating between the real vehicle and simulator experiments.

CONCLUSION

Gaussian Process models, using only a small amount of training data, can produce excellent predictions of on-axis longitudinal helicopter response data. In both a qualitative and quantitative sense (lower RMSE), the GP-predicted responses are an improvement on a physics-based model developed for real-time operation. The GPs also produce, to some degree, 'better' predictions for the off-axis responses of roll and yaw rate using training data based only upon longitudinal axis inputs. The results shown in the paper provide an excellent basis for the prediction of *model error* in future work.

The inclusion of the hyperparameter uncertainty into the full model predictions did not add any significant uncertainty to the model predictions. For future use, it should not be necessary to include this in the model.

FUTURE WORK

As noted above, the on-axis predictions from the GP model are very good, the off-axis predictions less so. It is anticipated that the prediction of roll and yaw rate could be improved by incorporating other inputs into the NARX structure, i.e. response data from flight test points that use lateral cyclic and rudder pedal position. This will form an early part of the future work.

As this work is only a preliminary step towards the author's main aim which is the prediction of *model error*, this would be one of the next steps. Further future work would include implementing our data-based model into e.g. FLIGHTLAB software for simulations purposes.

The GP was only trained on a limited amount of data; these training points were chosen by selecting every 25th point. One of the problems associated with GPs is their computational cost of training ($O(N^3)$, where N is the number of training points). It is therefore beneficial to choose a subset of the available data to use for training. A method called 'sparse Gaussian Processes' (Ref. 22) can be used on the training data to automatically select points which are more 'information rich'. A useful property of GPs is that they can determine the relative relevance of the input data; this information can be used to analyse whether or not the correct NARX structure has been used. This property could be utilized to establish which other helicopter inputs to the NARX structure should be included to improve the accuracy of the predicted response.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) Institut für Flugsystemtech-

nik for the provision of the flight test data and the permission to use it for this study.

REFERENCES

1. Shy, KS, Hageman, JJ, and Le, JH. The Role of Aircraft Simulation in Improving Flight Safety Through Control Training. National Aeronautics and Space Administration, Dryden Flight Research Center, 2002.
2. Bell, HH and Waag, WL. Evaluating the effectiveness of flight simulators for training combat skills: A review. The international journal of aviation psychology 1998;8:223–242.
3. Padfield, GD. Helicopter flight dynamics. John Wiley & Sons, 2008.
4. Punjani, A and Abbeel, P. Deep learning helicopter dynamics models. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015:3223–3230.
5. Anon. JAR-FSTD H, Helicopter Flight Simulation Training Devices. May, 2008.
6. Anon. FAA Advisory Circular AC120-63 Helicopter Simulator Qualification. November, 1994.
7. Perfect, P, Timson, E, White, MD, Padfield, GD, Erdos, R, and Gubbels, AW. A rating scale for the subjective assessment of simulation fidelity. The Aeronautical Journal 2014;118:953–974.
8. Kennedy, MC and O'Hagan, A. Bayesian calibration of computer models. Journal of the Royal Statistical Society. Series B, Statistical Methodology 2001:425–464.
9. Anon. FLIGHTLAB Development Software, Advanced Rotorcraft Tehcnology Inc, Sunnydale, California. `http://www.flightlab.com/flightlab.html`.
10. Bishop, C. Pattern recognition and machine learning. springer, 2006.
11. Rasmussen, CE and Williams, C. Gaussian processes for machine learning. 2006. Cited on 2006:1–31.
12. Higdon, D, Gattiker, J, Williams, B, and Rightley, M. Computer model calibration using high-dimensional output. Journal of the American Statistical Association 2008;103.
13. Worden, K, Manson, G, and Cross, E. On Gaussian Process NARX Models and Their Higher-Order Frequency Response Functions. In: *Solving Computationally Expensive Engineering Problems*. Springer, 2014:315–335.
14. Worden, K, Surace, C, and Becker, W. Uncertainty Bounds on Higher-Order FRFs from Gaussian Process NARX Models. Procedia Engineering 2017;199:1994–2000.
15. Manso, S. Simulation and system identification of helicopter dynamics using support vector regression. The Aeronautical Journal 2015;119:1541–1560.
16. Kumar, MV, Omkar, S, Ganguli, R, Sampath, P, and Suresh, S. Identification of helicopter dynamics using recurrent neural networks and flight data. Journal of the American Helicopter Society 2006;51:164–174.
17. Omkar, S, Mudigere, D, Senthilnath, J, and Kumar, MV. Identification of Helicopter Dynamics based on Flight Data using Nature Inspired Techniques. arXiv preprint arXiv:1411.3251 2014.
18. Girard, A. Approximate methods for propagation of uncertainty with Gaussian process models. PhD thesis. University of Glasgow, 2004.
19. Green, P and Worden, K. Bayesian and Markov Chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty. Phil. Trans. R. Soc. A 2015;373:20140405.
20. Green, P. Bayesian system identification of a nonlinear dynamical system using a novel variant of simulated annealing. Mechanical Systems and Signal Processing 2015;52:133–146.
21. Saul, AD, Hensman, J, Vehtari, A, and Lawrence, ND. Chained gaussian processes. In: *Artificial Intelligence and Statistics*. 2016:1431–1440.
22. Titsias, MK. Variational Model Selection for Sparse Gaussian Process Regression.