Simulation and prediction of main rotor, tail rotor, and engine parameters from flight tests

M N Beaulieu and **R M Botez***
Laboratoire Avancé de Recherche en Commande, Avionique et Aéroservoélasticité, École de Technologie Supérieure, Montréal, Québec, Canada

The manuscript was received on 15 November 2007 and was accepted after revision for publication on 30 April 2008.

DOI: 10.1243/09544100JAERO313

Abstract: In the framework of this research project, the main rotor torque, tail rotor torque, engine torque, and main rotor speed of a helicopter in forward flight are estimated by using a state-space model from flight tests data. The state-space model inputs are non-linear terms made of combinations of pilot controls and helicopter states. The model simulates the helicopter outputs while knowing the states and controls at all times. It was also implemented as a prediction tool, for possible use in an envelope protection flight control system in which the states, controls, and outputs are known at the present time, and predict the future helicopter states and controls following to pilot controls time history. The state-space model parameters are identified by using the subspace identification method, a relatively recent non-iterative algorithm, which constructs an observability matrix from input and output data and uses this matrix to obtain the state-space matrices. The obtained parameters are then optimized with the Levenberg–Marquardt output-error method. A comparison of the results with and without optimization is also conducted. The results show that the subspace method provides a good estimate of the outputs within the FAA tolerance bands and that these results can further be improved by use of the minimization algorithm. The generated model using the subspace method is found to be very good for prediction applications, which makes it a promising model for flight control simulator applications.

Keywords: helicopter, rotor, torque, flight test, modelling, state-space, subspace

1 INTRODUCTION

In order to ensure safe and efficient operation of a rotorcraft, it is very important to understand the relationships between the parameters related to the aerodynamics and controls of rotors and engines, such as the main rotor torque, the pilot inputs, and the helicopter states. The focus of this paper is the generation of mathematical model for the main rotor torque, tail rotor torque, engine torque, and main rotor speed of a helicopter.

The model uses the rotorcraft states and the pilot controls as inputs. Different implementations of such a mathematical model can be used in different applications such as flight simulators and envelope protection control systems.

In a flight simulator application, the rotorcraft states are given by the flight simulator model, which uses these parameters in conjunction with the pilot control inputs to estimate the main rotor torque and displays its value in the cockpit. It is very important for the pilot safety to learn how his manoeuvres affect the rotors and the engines, because of the fact that when these parameters exceed certain limits, they can have a detrimental effect on the helicopter's fatigue life and its handling qualities safety.

Such a mathematical model can also be used in envelope protection control systems. In this case, it is necessary to ensure that some values remain within their prescribed limits. In this paper, the limit parameters are the model outputs as defined previously. In order to prevent such a limit violation, it is necessary

*Corresponding author: *Laboratoire Avancé de Recherche en Commande, Avionique et Aéroservoélasticité, École de Technologie Supérieure, 1100 Notre-Dame West, Montréal, Québec, Canada H3C 1K3. email: ruxandra.botez@etsmtl.ca*

to know the relationship between the future control inputs and the future values of the limit parameters. This relationship can be found by constructing a predictive model, which uses the states and outputs of the current values from flight test data to predict the future values of the outputs based on assumed futures values of the pilot inputs. Inverting this model provides the future control inputs that would result in a limit violation. Once these relationships are known, there are two possible options: autonomous restraint, aural and visual limit cues, and carefree manoeuvring control systems [**1**].

In the first option, the control system can override the pilot and change the control inputs in order to avoid reaching the limit, which is suitable whenever a violation of the limit can lead to a catastrophic accident. Examples of such limits are the rotorcraft stall and airspeed limits. The main advantage of this option is that the pilot workload is greatly reduced. However, such systems de-emphasize pilot judgement in critical decisions. In the second option, the system only warns the pilot of approaching limits without overriding his actions. This method is the opposite of the first one in the sense that the pilot has full authority over the aircraft, but a greater workload. Whenever the pilot needs to perform very aggressive manoeuvres, it may also lead him to be over cautious to avoid violating the limit, which leads to a decreased aircraft performance.

There is also a third option, which is a compromise between the first two options. In this case, a progressive resistance is implemented on the helicopter controls as the limit is approached. The pilot does not need to monitor his controls to know how far the limit is because he knows it intuitively from the resistance exerted on the controls. If he wants to perform an aggressive manoeuvre, the pilot can chose to follow the force cue and allow it to guide him along the helicopter limit or he can chose to override it if he requires a greater vehicle performance regardless of the risk (for example, to avoid an obstacle). This option is suitable when a limit violation (e.g. main rotor torque or main rotor speed) can be detrimental, but not necessarily catastrophic. According to a survey of 70 UK military helicopter pilots [**2**], 75 per cent of the pilots estimated that monitoring the torque limit had a significant impact on mission performance, 60 per cent of them also believed the same for rotor speed limits. In this paper, a predictive model that could be used in such a control system is generated. The model structure used in this study so as to estimate the limit parameters is a state-space model with non-linear inputs. This state-space model was constructed with B-427 type helicopter flight test data. The flight test data from this type alone were used to generate and validate the model described in this paper. Therefore, the model would not likely be valid for other helicopter types. However, assuming that good flight test data were available for another helicopter, and by applying a similar methodology would lead to another model with similar accuracy and structure.

In order to obtain the torque response to a broad range of motions, 2311 [**3**] manoeuvres were performed on each helicopter control at different true airspeeds, altitudes, gross weights, and centre of gravity positions. The parameters in the state-space matrices were identified by means of the subspace identification method. The MATLAB® implementation of the subspace system identification method [**4**–**6**] is an efficient non-iterative algorithm, which uses inputs and outputs data to directly obtain the system observability matrix, which is further used to obtain the **A**, **B**, **C**, and **D** state-space matrices. Since, it is non-iterative, the subspace identification method is much faster than alternative methods, which require optimization. Furthermore, it is not affected by iterations, such as the possible convergence of the solution towards the local minimum instead of the global minimum and does not require *apriori* knowledge of the parameter values. Following the application of the subspace method to obtain the initial guesses for the parameter values in the state-space matrices, an output error method based on the past Levenberg–Marquardt minimization algorithm was used to refine the results. The applications of these methods in the Aerospace Industry will be detailed as follows.

In 1995, Howitt [**7**] used a simplified mathematical model to estimate the engine torque and main rotor torque of the BO-105 helicopter following a collective step input and designed carefree control laws based on this simplified model. Many of the research efforts that followed used neural networks to predict the limit parameters' future value. Menon *et al.* [**8**] were able to predict the main rotor speed of a helicopter at one sample time instant in the future with an adaptive linear neural network model. In this algorithm, the neural networks' weights were adjusted online using the prediction error and showed one case with a low prediction error (less than 0.25 per cent). Neural networks were also used to predict the main rotor hub moments for a load-monitoring technology application on the SH-60 helicopter [**9**]. In 1999, Horn [**10**] and Horn *et al.* [**11**] used offline trained neural networks to predict the control inputs that would result in violation of the torque, load factor or angle of attack limits on the V-22 aircraft in dynamic trim. The dynamic trim was defined as the condition for which the fast aircraft states (angular rates, and so on) have reached steady state and the slow states (Euler angles, TAS, etc.) continued to vary in time. However, they did not worry

about the limited parameters' peak value following a control input. Yavrucuk *et al.* [**12**, **13**] and Yavrucuk and Prasad [**14**] also estimated the dynamic trim limited parameter of the rotorcraft, but used an approximate linear model corrected by an online trained adaptive neural network. Vineet *et al.* [**15**], Horn and Sahani [**16**], and Sahani [**17**] also used a neural network to estimate the value of a limited parameter dynamic trim and added an approximate linear model to obtain its peak value.

In section 2, the authors explain the flight conditions covered by the model as well as the manoeuvres used in its identification. Then, the implementation of the model in the simulation and the details on the subspace identification algorithm will be given. The method to use the model as a prediction tool for an envelope protection control system application will be presented further.

2 METHODOLOGY

2.1 Flight conditions and manoeuvres used in model identification and validation

Different state-space models were identified for flight test conditions expressed by different altitudes, gross weights, and centre of gravity positions in forward flight (Table 1).

For each flight test conditions, a number of different forward flight records (Table 1) were used to identify the models and other records were used to validate them. One record comprising a flight time history that starts when the helicopter is at trim, then the pilot performs a manoeuvre and records the effects of his manoeuvre on the helicopter states and controls. The manoeuvres are called 2311 as they consist of four step inputs lasting for 2, 3, 1, and 1 s, respectively. Such manoeuvres are used because they excite both the short- and long-period frequencies of the helicopter motion. For each record, a 2311 manoeuvre

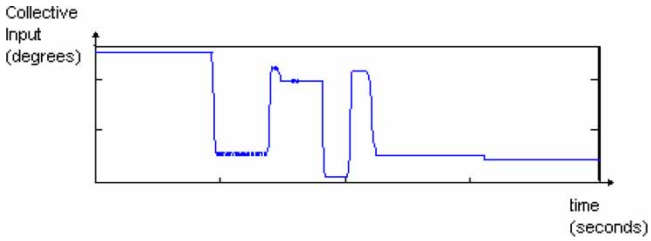

Fig. 1 Sample time history of a control position during a 2311 manoeuvre

was performed on one of the helicopter's controls (collective, longitudinal cyclic, lateral cyclic, and pedal). Figure 1 shows an example of these 2311 manoeuvres. The following section describes the inputs and outputs of the model.

2.2 Inputs and outputs of the state-space model

In this paper, the main rotor torque, tail rotor torque, engine torque, and main rotor speed are represented by using the state-space models. Figure 2 shows the structure of the mathematical model for the estimation of these parameters.

In this figure, the mathematical model inputs are the helicopter states, the pilot control inputs, and rate of changes of their control inputs. The helicopter states are the linear velocities (u, v, w) and angular velocities (p, q, r) in the body axis coordinate system. The helicopter states were measured on the helicopter during the flight test program. If the model was implemented in a flight simulator, the states could be provided from the simulator flight model, and should have very close values to the states values from the flight test data if the helicopter flight model would be properly designed. The actual state-space model inputs are the previously mentioned inputs and higher order terms made of products of different inputs. For example, for the main

Table 1 Flight test conditions used to identify and validate the proposed model in forward flight

Flight test condition	Gross weight (light or heavy)	Centre of gravity position (Forward Mid or Aft)	Altitude range (1000 ft)	True airspeed range (knots)	Number of records used to identify the model	Number of records used to validate the model
1	Heavy	Aft	4–8	60–160	42	12
2	Heavy	Aft	8–12	50–160	49	20
3	Light	Aft	3–6	40–110	28	12
4	Light	Aft	6–8	35–130	69	22
5	Light	Aft	8–10	50–70	32	12
6	Heavy	Mid	5–8	30–130	21	4
7	Heavy	Fwd	0–4.5	40–140	33	12
8	Heavy	Fwd	4.5–7	30–130	30	8
9	Heavy	Fwd	7–10	30–130	56	20
10	Light	Fwd	4–9	30–130	49	16
Total					409	138

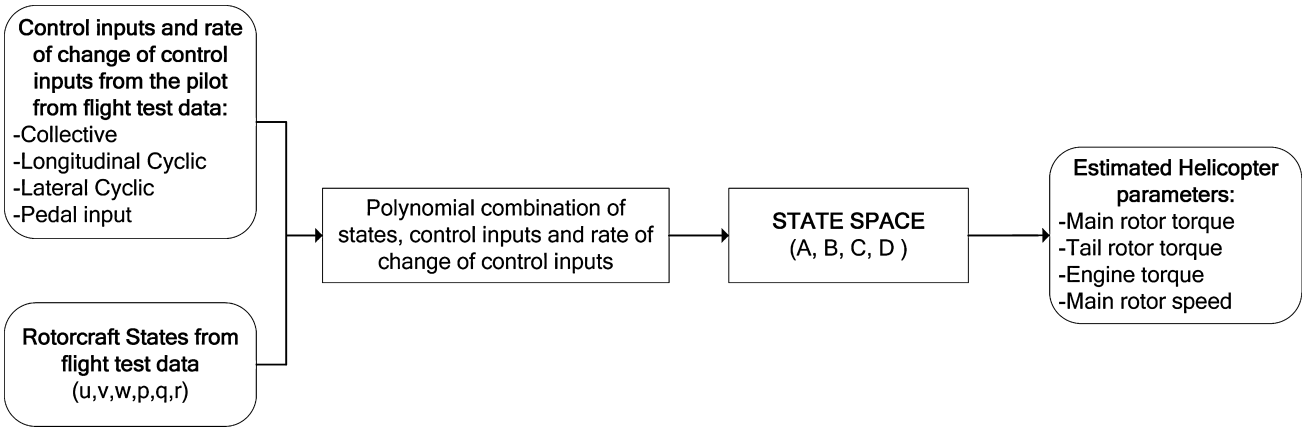

Fig. 2 State-space model architecture for the identification of the main rotor torque

rotor torque, the state-space model inputs are

$$\begin{aligned}\text{Inputs}_{\text{main rotor torque}} &= \left[\text{coll},\ \text{long},\ \text{lat},\ \text{ped},\ \frac{\text{dcoll}}{\text{d}t}, u, v, w, p, q, r,\right.\\ &\quad \text{coll}^2, \text{coll}\cdot\frac{\text{dcoll}}{\text{d}t}, \text{coll}\cdot q, \text{coll}\cdot u, u\\ &\quad \cdot\frac{\text{dcoll}}{\text{d}t}, v\cdot\frac{\text{dcoll}}{\text{d}t}, q\cdot\frac{\text{dcoll}}{\text{d}t}, \text{coll}^2\cdot\frac{\text{dcoll}}{\text{d}t},\\ &\quad \left.\text{coll}^2\cdot u,\ w^2,\ r\cdot w,\ \text{ped}^2,\ r\cdot\text{ped}\right] \end{aligned} \quad (1)$$

where coll is the collective position, long the longitudinal cyclic position, lat the lateral cyclic position, and ped the pedal position. Note that even though state-space models usually represent linear models, this model is highly non-linear as the state-space model inputs are non-linear. The same model structure was used for the tail rotor torque, the engine torque, and the main rotor speed, but with different higher order terms selected by trial-and-error, and each non-linear term was kept only if it improved the results.

The criteria used to judge the effect of a given non-linear term addition was mainly given by the mean prediction error between the model's result and the flight test data. Usually, the addition of a given non-linear term improved the results of the model when the mean error was computed against the flight test data used for model identification. However, in some cases, the addition of a non-linear term degraded the mean error on the validation data (not used in the identification process). Therefore, the addition of this non-linear term approached the model to the data used to generate it and reduced the generality of the mathematical model. Such a model overfits the data and when this happens, the non-linear term was discarded. Another criterion to reject a non-linear term was the model's mean error was improved when the percentage of time when the model error was greater than a tolerance band (e.g. 3 per cent for the main rotor torque). This tolerance band is described in later sections of this paper. Usually, one set of records was used for the identification and another set of records was set aside for validation.

2.3 Subspace identification method

The parameters of the state-space matrices (Fig. 2) were obtained by using the subspace system identification algorithm, which is briefly described in this section. Generally, a discrete linear model is defined by equations (2a) and (3b)

$$\boldsymbol{x}\,(t+\Delta T)_{n\times 1} = \hat{\boldsymbol{x}}\,(t+\Delta T)_{n\times 1} + K_{n\times o}e(t)_{o\times 1} \quad (2a)$$

$$\boldsymbol{y}(t)_{o\times 1} = \hat{\boldsymbol{y}}(t)_{o\times 1} + e(t)_{o\times 1} \quad (2b)$$

where t is the time and ΔT the time increment equal to the record sampling rate. In equation (2b), the vector $\boldsymbol{y}(t)$ represents the system's measured outputs from flight test data, $\hat{\boldsymbol{y}}(t)$ the system's estimated outputs, and $e(t)$ the error between the flight tests and the estimated outputs. If the system is properly identified, the vector $\boldsymbol{e}$ is a white Gaussian noise vector with a zero mean value. The index o is the system's number of outputs, for example, if the output is only the main rotor torque, then it has a value of one. In equation (2a), the vector $\boldsymbol{x}(t)$ of length n represents the system's true states expressed as linear combinations between previous inputs and previous outputs. The value of n represents the model's order and the size of the matrix **A** given in equation (3a). An higher value of n implies a more complex system of equations.

The system's estimated states are represented by the vector $\hat{\boldsymbol{x}}(t)$ and the matrix **K**, called the noise disturbance matrix, represents the effect of the measurement noise on the state noise. The *estimated* states $\hat{\boldsymbol{x}}(t)$ are used to find the estimated output vectors $\hat{\boldsymbol{y}}(t)$ as shown in the widely known state-space system of

equations (3a) and (3b)

$$\hat{\boldsymbol{x}}(t+\Delta T)_{n\times 1} = \mathbf{A}_{n\times n}\hat{\boldsymbol{x}}(t)_{n\times 1} + \mathbf{B}_{n\times m}\boldsymbol{u}(t)_{m\times 1} \tag{3a}$$

$$\hat{\boldsymbol{y}}(t)_{o\times 1} = \mathbf{C}_{o\times n}\hat{\boldsymbol{x}}(t)_{n\times 1} + \mathrm{D}_{o\times m}\boldsymbol{u}(t)_{m\times 1} \tag{3b}$$

where $\boldsymbol{u}(t)$ is the system's inputs assembled in a vector of size m, for example, in this paper the size of the subscript m is equal to the number of linear and nonlinear inputs described in equation (1). Matrix **A** is the state matrix whose rank is equal to the system order. Matrix **B** represents the effect on the states of each input defined in equation (1). Matrix **C** relates the outputs to the system's states. Matrix **D** relates the outputs to the system's inputs. The non-zero **D** matrix is equivalent to a system in which the inputs influence the outputs without time delay. Since the system identified in this study is a dynamic one, there is always a time delay between the inputs and outputs, and therefore the **D** matrix is a null matrix. When the **D** matrix is null, the outputs $\hat{\boldsymbol{y}}(t)$ are only functions of the state vectors $\hat{\boldsymbol{x}}(t)$, which are written as functions of the inputs and states at the previous time step (equation (3a)). If the model should be used for a flight simulation, the model error is unknown because no flight test measurement is available and only equations (3a) and (3b) are used to simulate the outputs. In other words, it is not necessary to take the noise covariance matrix **K** into account.

The terms of the matrices **A**, **B**, **C**, **D**, and **K** are usually estimated by means of various parameter estimation methods. Most classical parameter estimation methods start with a set of first guesses, which can be based on physical insight of the system and iterate from these guesses to minimize the error between the model and the given data with a minimization algorithm. If the initial parameter guesses are far from their true values, the minimization algorithm may converge towards a local minimum, which is a disadvantage in this method.

In the framework of this study, the authors have chosen the subspace identification algorithm. Its main advantage resides in the fact that it is a non-iterative algorithm, which does not require any initial guess of the terms in these matrices, and therefore finds the parameters solely from the known inputs and outputs. For this reason, the subspace identification method is much faster than the classical estimation methods and has no problem related to optimization, such as the possible convergence of the solution towards the local minimum instead of the global minimum. Furthermore, this method does not require any *apriori* knowledge of the system. The subspace identification algorithm is implemented with the MATLAB® System Identification Toolbox. The basic theory behind this algorithm is described in reference [**5**] and the manner in which the algorithm is implemented in MATLAB® is presented in reference [**6**]. The subspace system identification method has been successfully used in recent literature for different applications such as fibre optic research [**4**] and the identification of aeroelastic instabilities on an F/A-18 aircraft [**18**].

The main concept behind the subspace method is the definition of the system observability matrix $\boldsymbol{\Gamma}_r$ in equation (4) from modern control theories, where the forward prediction horizon is represented by r [**5**]. This matrix can be obtained from the system's inputs $\boldsymbol{u}(t)$ and outputs $\boldsymbol{y}(t)$ and its expression is as follows

$$\boldsymbol{\Gamma}_r \overset{\text{def}}{=} \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{r-1} \end{bmatrix} \tag{4}$$

Once this observability matrix $\boldsymbol{\Gamma}_r$ is known [**5**], the state-space matrices (**A**, **B**, **C**, **D**, **K**) can be obtained.

When the subspace method is used, the order of the state-space system can be defined by the user so as to obtain better results. In general, a higher order system will give a better match for the data records used during the model identification process. However, choosing a too high order can lead to a loss of generality which may result into a degradation of the results on the records set aside for the validation process. The model orders which offer the best compromise are: 3 for the main rotor torque, 4 for the tail rotor torque, 2 for the engine torque, and 2 for the main rotor speed model.

2.4 Refinement of the subspace identification method by use of the Levenberg–Marquardt minimization algorithm

Following the application of the subspace method, the parameter values in the state-space matrices are further refined by using the Levenberg–Marquardt minimization algorithm [**3**] with the parameters found by using the subspace method as the first guesses. In practice, it would not be necessary to perform an optimization, but it was done in this paper to be able to compare the results with and without optimization and thus evaluate if a model identified with the subspace method was optimal. The identification results are obtained and further compared with and without this minimization method. The cost function to be minimized is the output error, which is defined by the following equation

$$J(\hat{\boldsymbol{\theta}}) = \frac{2}{N}\sum_{i=1}^{N}\left[\boldsymbol{y}(t_i) - \hat{\boldsymbol{y}}\left(t_i|\,\hat{\boldsymbol{\theta}}\right)\right]^2 \tag{5}$$

where $J(\hat{\boldsymbol{\theta}})$ is the cost function, $\boldsymbol{y}(t_i)$ the outputs from the flight test data at time t_i, and $\hat{\boldsymbol{y}}(t_i|\hat{\boldsymbol{\theta}})$ the estimated

outputs at time t_i, which depends on the model's estimated parameter values $\hat{\boldsymbol{\theta}}$. In this equation, the error between the estimated outputs and outputs from flight test data is summed over the length of the record where N correspond to the number of data points in the data vector. This cost function represents the least square error between the estimated outputs (such as the main rotor torque) and the outputs from flight test for a given value of the vector $\hat{\boldsymbol{\theta}}$ of estimated parameters.

In order to minimize the cost function defined in equation (5), it was decided to use the Levenberg–Marquardt minimization algorithm. This algorithm was chosen because it combines the advantages of two well-known algorithms: the gradient descent algorithm and the Gauss–Newton algorithm. In both these algorithms, the $\hat{\boldsymbol{\theta}}$ parameter estimate is updated at each iteration by using the following equation

$$\hat{\boldsymbol{\theta}}_{j+1} = \hat{\boldsymbol{\theta}}_j + \Delta\hat{\boldsymbol{\theta}}_j \tag{6}$$

where j is the iteration number and $\Delta\hat{\boldsymbol{\theta}}$ an increment vector for each parameter $\hat{\boldsymbol{\theta}}$ in the state-space matrices. The optimization algorithm determines these increments' value. For both algorithms, this increment is proportional to the cost function $J(\hat{\boldsymbol{\theta}})$ values, which means that as the algorithm converges towards a minimum, the $\Delta\hat{\boldsymbol{\theta}}$ increments are reduced. Both algorithms are now explained as follows.

2.4.1 Gradient descent algorithm

In the gradient descent algorithm, the gradient of the $J(\hat{\boldsymbol{\theta}})$ cost function is determined and the parameters are updated in the negative direction of this gradient, as expressed in the following equation

$$\hat{\boldsymbol{\theta}}_{j+1} = \hat{\boldsymbol{\theta}}_j - \frac{\partial J(\hat{\boldsymbol{\theta}}_j)}{\partial \hat{\boldsymbol{\theta}}_j} \tag{7}$$

The second-term on the right-hand side of equation (7) is obtained by differentiating the term given by equation (5) with respect to $\hat{\boldsymbol{\theta}}_j$. The change in parameters $\Delta\hat{\boldsymbol{\theta}}_j$ from equation (6) therefore it becomes

$$\begin{aligned}\Delta\hat{\boldsymbol{\theta}}_j &= -\frac{\partial J\left(\hat{\boldsymbol{\theta}}_j\right)}{\partial \hat{\boldsymbol{\theta}}_j} \\ &= \frac{2}{N}\sum_{i=1}^{N}\left[\frac{\partial \hat{\boldsymbol{y}}\left(t_i|\hat{\boldsymbol{\theta}}_j\right)}{\partial \hat{\boldsymbol{\theta}}_j}\right]^{\mathrm{T}}\left[\boldsymbol{y}\left(t_i\right) - \hat{\boldsymbol{y}}\left(t_i|\hat{\boldsymbol{\theta}}_j\right)\right] \\ &= -G\end{aligned} \tag{8}$$

where G is the gradient and each parameter in the $\hat{\boldsymbol{\theta}}_j$ vector at the jth iteration. This algorithm is known to be fairly robust, but may require a significant number of iterations to reach a minimum. The Gauss–Newton algorithm, which usually converges faster than the gradient descent algorithm, and for this reason, is described as follows.

2.4.2 Gauss–Newton algorithm

The Gauss–Newton algorithm is derived from the postulate that at a minimum, the cost function gradient is zero as shown in equation (9)

$$\frac{\partial J(\hat{\boldsymbol{\theta}}_j)}{\partial \hat{\boldsymbol{\theta}}_j} = 0 \tag{9}$$

The value of the cost function gradient at iteration $j+1$ can be calculated by a Taylor series expansion of its value at iteration j, as shown in the following equation

$$\left(\frac{\partial J(\hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}}\right)_{j+1} \approx \left(\frac{\partial J(\hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}}\right)_j + \left(\frac{\partial^2 J(\hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}^2}\right)_j \Delta\hat{\boldsymbol{\theta}} \tag{10}$$

The gradient is set at iteration $j+1$ to zero in equation (10) and isolating the term $\Delta\hat{\boldsymbol{\theta}}$ yields to the following equation

$$\Delta\hat{\boldsymbol{\theta}} = -\left(\frac{\partial J^2(\hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}^2}\right)_j^{-1}\left(\frac{\partial J(\hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}}\right)_j \tag{11}$$

The double derivative in equation (11) may be evaluated from equation (8) in the following manner

$$\begin{aligned}\frac{\partial J^2(\hat{\boldsymbol{\theta}}_j)}{\partial \hat{\boldsymbol{\theta}}_j^2} &= \frac{2}{N}\sum_{i=1}^{N}\left[\frac{\partial \hat{\boldsymbol{y}}(t_i|\hat{\boldsymbol{\theta}}_j)}{\partial \hat{\boldsymbol{\theta}}_j}\right]^{\mathrm{T}}\frac{\partial \hat{\boldsymbol{y}}(t_i|\hat{\boldsymbol{\theta}}_j)}{\partial \hat{\boldsymbol{\theta}}_j} \\ &\quad + \frac{2}{N}\sum_{i=1}^{N}\left[\frac{\partial^2 \hat{\boldsymbol{y}}(t_i|\hat{\boldsymbol{\theta}}_j)}{\partial \hat{\boldsymbol{\theta}}_j^2}\right]^{\mathrm{T}}\left[\boldsymbol{y}(t_i) - \hat{\boldsymbol{y}}(t_i|\hat{\boldsymbol{\theta}}_j)\right]\end{aligned} \tag{12}$$

In case when equation (12) was used to complete this second derivative, this method would be referred to as the Newton method. However, in the Gauss–Newton method, the second term is neglected for two reasons.

1. It requires a lot of calculations to obtain the output's second derivative.
2. It tends to zero as the optimization algorithm converges since the term $[\boldsymbol{y}(t_i) - \hat{\boldsymbol{y}}(t_i|\hat{\boldsymbol{\theta}}_j)]$ tends to zero.

Equation (13) therefore becomes

$$\frac{\partial J^2(\hat{\boldsymbol{\theta}}_j)}{\partial \hat{\boldsymbol{\theta}}_j^2} \approx \frac{2}{N}\sum_{i=1}^{N}\left[\frac{\partial \hat{\boldsymbol{y}}(t_i|\hat{\boldsymbol{\theta}}_j)}{\partial \hat{\boldsymbol{\theta}}_j}\right]^{\mathrm{T}}\frac{\partial \hat{\boldsymbol{y}}(t_i|\hat{\boldsymbol{\theta}}_j)}{\partial \hat{\boldsymbol{\theta}}_j} = F \tag{13}$$

Substituting equations (8) and (13) in equation (11), yields the following equation for the parameter

increment in the Gauss–Newton algorithm, where F represents the second derivative

$$F\Delta\hat{\boldsymbol{\theta}} = -G \tag{14}$$

The Gauss–Newton algorithm is known to converge faster than the gradient descent algorithm, however, it is very sensitive to the initial conditions and may converge towards a local minimum. A good optimization approach consists in the combination of the advantages of the gradient descent algorithm and the Gauss–Newton algorithm of the Levenberg–Marquardt method used in the framework of this research. The parameter vector increment in the Levenberg–Marquardt algorithm is defined by the following equation

$$[(1-\lambda)F + \lambda I]\Delta\hat{\boldsymbol{\theta}} = -G \tag{15}$$

From equations (8) and (15), it clearly appears that the Levenberg–Marquardt method is a linear combination of the gradient descent and the Gauss–Newton methods. The coefficient λ is called the Levenberg–Marquardt parameter. If its value is zero, then the algorithm is a pure Gauss–Newton algorithm; if its value is one, then the algorithm is a pure gradient descent algorithm. This parameter varies as the optimization is carried on so as to ensure the fastest possible convergence rate.

The subspace method was used to obtain the initial guesses of the state-space model parameters. The Levenberg–Marquardt method was further used to refine these parameters' values. The results will show the model error with and without this optimization. The implementation method presented in Fig. 2 is valid for the model's implementation in a flight simulator. As discussed previously, these limits parameter models could also be used in a carefree envelope protection control system. In this case, the model implementation would be different than the one presented in Fig. 2, and will be discussed in the following section.

2.5 Implementation of the prediction model

In the case of a carefree envelope protection control system, it is necessary to limit the future value of limit parameters (main and tail rotor torque and speed). During the flight, the limit parameter's current value, the helicopter states, and its control inputs are known. In order to prevent any limit violation, it is necessary to predict the future values of these parameters for a given amount of time in the future (prediction horizon). According to Jeram [**19**], the required prediction horizons for a cueing application lie between 0.25 and 0.5 s. A high prediction horizon gives more time to the cueing system to warn the pilot of an incoming envelope limit, but this comes at the cost of a higher model error than a smaller prediction horizon. The appropriate prediction horizon should ultimately be determined by the pilot during a flight test program. The model used to predict the future values of limit parameters is illustrated in Fig. 3.

As shown in this figure, two state-space models are necessary to predict the future value of the helicopter's limit parameters. The first state-space model uses the current states from flight test data and an assumed time history of the control inputs to predict the future time histories of the helicopter states. In practice, the assumed future value of the pilot control inputs can be either a worse case scenario or more commonly, similar to the control inputs used to train the model [**19**]. The future states' time history is combined with the future value of the pilot's control inputs in polynomial functions to construct the inputs of the limit parameter models as in equation (1).

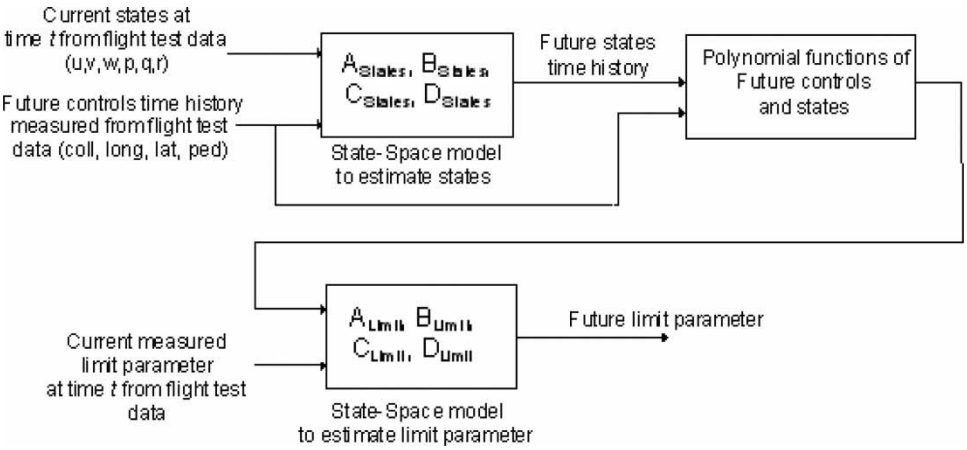

Fig. 3 Model structure used for the prediction of the future value of a limit parameter (main rotor torque, tail rotor torque, engine torque or main rotor speed)

The second state-space model in the figure is used to predict the future time history of a given limit parameter. Its inputs are the assumed future controls and the outputs of the first state-space model as described above.

The following sections deal with the inputs and outputs of the first state-space model shown in Fig. 3. Then, the method to be used for the prediction of a future outputs time history by means of a Kalman recursion [**20**, **21**] is discussed.

2.6 State-space model to predict the future helicopter states

The state-space model used to estimate the future helicopter states (the first state-space model presented in Fig. 3) is shown in the following system of equations obtained from equations (3a) and (3b)

$$\begin{bmatrix} \hat{\boldsymbol{u}}(t+\Delta t) \\ \hat{\boldsymbol{v}}(t+\Delta t) \\ \hat{\boldsymbol{w}}(t+\Delta t) \\ \hat{\boldsymbol{p}}(t+\Delta t) \\ \hat{\boldsymbol{q}}(t+\Delta t) \\ \hat{\boldsymbol{r}}(t+\Delta t) \end{bmatrix} = \mathbf{A}_{\text{states}} \begin{bmatrix} \hat{\boldsymbol{u}}(t) \\ \hat{\boldsymbol{v}}(t) \\ \hat{\boldsymbol{w}}(t) \\ \hat{\boldsymbol{p}}(t) \\ \hat{\boldsymbol{q}}(t) \\ \hat{\boldsymbol{r}}(t) \end{bmatrix} + \mathbf{B}_{\text{states}} \begin{bmatrix} \text{coll}(t) \\ \text{long}(t) \\ \text{lat}(t) \\ \text{ped}(t) \end{bmatrix} \tag{16a}$$

$$\begin{bmatrix} \hat{\boldsymbol{u}}(t+\Delta t) \\ \hat{\boldsymbol{v}}(t+\Delta t) \\ \hat{\boldsymbol{w}}(t+\Delta t) \\ \hat{\boldsymbol{p}}(t+\Delta t) \\ \hat{\boldsymbol{q}}(t+\Delta t) \\ \hat{\boldsymbol{r}}(t+\Delta t) \end{bmatrix} = \mathbf{C}_{\text{states}} \begin{bmatrix} \hat{\boldsymbol{u}}(t) \\ \hat{\boldsymbol{v}}(t) \\ \hat{\boldsymbol{w}}(t) \\ \hat{\boldsymbol{p}}(t) \\ \hat{\boldsymbol{q}}(t) \\ \hat{\boldsymbol{r}}(t) \end{bmatrix} + \mathbf{D}_{\text{states}} \begin{bmatrix} \text{coll}(t) \\ \text{long}(t) \\ \text{lat}(t) \\ \text{ped}(t) \end{bmatrix} \tag{16b}$$

In equations (16a) and (16b), the states which are the helicopter's linear and angular velocities in body axes, are estimated with an approximate linear state-space system. The term Δt represents the simulation's sample time. Since the outputs and the states are the same, the $\mathbf{C}_{\text{states}}$ matrix is the identity matrix, while the $\mathbf{D}_{\text{states}}$ matrix is a null matrix. Even though this linear model is very approximate to estimate the future helicopter states, it has been found to be accurate enough in the overall model to provide a proper prediction of the limit parameter's future values within a given prediction horizon.

2.7 Future prediction outputs based on the current outputs from flight test data and a state-space model

In this section, a state-space model, which jointly uses the outputs from flight test data and the future inputs time history, to predict its future outputs time history is implemented. This procedure was first developed by Kalman [**21**] and is widely used in control theory [**20**]. First, the one time step prediction is demonstrated, and then the multiple.

2.7.1 *One time step ahead prediction*

In this case, the output's current value is obtained from the flight test data. By knowing the flight test outputs $\boldsymbol{y}(t)$ and the estimated outputs $\hat{\boldsymbol{y}}(t)$, it is possible to determine the measurement error with the following equation derived from equation (2b), where the **D** matrix can be neglected since it is null

$$\boldsymbol{e}(t) = \boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t) = \boldsymbol{y}(t) - \mathbf{C}\hat{\boldsymbol{x}}(t) \tag{17}$$

This error should correspond to an uncorrelated perturbation with the inputs. From equations (17), (3a), and (2a), it is possible to obtain a new state-space equation as follows

$$\hat{\boldsymbol{x}}(t+\Delta t) = \mathbf{A}\hat{\boldsymbol{x}}(t) + \mathbf{B}\boldsymbol{u}(t) + \mathbf{K}\left[\boldsymbol{y}(t) - \mathbf{C}\hat{\boldsymbol{x}}(t)\right] \tag{18}$$

Equation (18) is known as the Kalman state observation equation [**20**, **21**]. By rearranging this equation and adding with equation (3b), neglecting the **D** matrix, yields the following equation

$$\begin{aligned} \hat{\boldsymbol{x}}(t+\Delta t) &= [\mathbf{A} - \mathbf{KC}]\hat{\boldsymbol{x}}(t) + \begin{bmatrix} \mathbf{K} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}(t) & \boldsymbol{u}(t) \end{bmatrix} \\ \hat{\boldsymbol{y}}(t) &= \mathbf{C}\hat{\boldsymbol{x}}(t) \end{aligned} \tag{19}$$

where $\boldsymbol{y}(t)$ represents the outputs from flight test and $\hat{\boldsymbol{y}}(t)$ the estimated outputs. Equation (19) applies when the outputs need to be predicted one time step in advance or, in other words, with a prediction horizon of one time step. Note that the length of one time step is equal to the flight test data sampling rate.

2.7.2 *Prediction with a higher horizon than one time step*

It is possible to predict the outputs further in time by using a larger prediction horizon. In this case, a pure simulation model, described in equation (3), is run using the last prediction found by equation (19) and the time length of the simulation's correspond to the prediction horizon. This scheme is best illustrated by an example or by a two time step prediction. If one wants to predict the value of the outputs at time 3, knowing only the flight test states x and inputs u at time 2 when time step is equal to 1, it is recommended to use equations (3a) and (3b) as follows

$$\hat{\boldsymbol{y}}(3) = \mathbf{C}\boldsymbol{x}(3) \tag{20a}$$

where

$$\boldsymbol{x}(3) = \mathbf{A}\boldsymbol{x}(2) + \mathbf{B}\boldsymbol{u}(2) \tag{20b}$$

In these equations, the numbers in parentheses correspond to the time samples. Equations (21a) and (21b)

may be combined to give

$$\hat{\boldsymbol{y}}(3) = \mathbf{C}[\mathbf{A}\boldsymbol{x}(2) + \mathbf{B}\boldsymbol{u}(2)] = \mathbf{CA}\boldsymbol{x}(2) + \mathbf{CB}\boldsymbol{u}(2) \quad (21)$$

The state $\boldsymbol{x}(2)$ is unknown, but can be estimated by using equation (20) where $t + \Delta t = 2$, as follows

$$\boldsymbol{x}(2) = [\mathbf{A} - \mathbf{KC}]\boldsymbol{x}(1) + \begin{bmatrix}\mathbf{K} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{y}(1) & \boldsymbol{u}(1)\end{bmatrix} \quad (22)$$

where $\boldsymbol{y}(1)$ are the outputs from flight test data, two steps ahead of the prediction, $\boldsymbol{u}(1)$ the system input's initial value, and $\boldsymbol{x}(1)$ the initial state. The same reasoning may be extended to any prediction horizon. The general equation used to predict the value of the outputs $\hat{\boldsymbol{y}}(t)$ using the outputs measured from flight test $\boldsymbol{y}(t-r)$ (prediction horizon of r) is

$$\hat{\boldsymbol{y}}(t) = \begin{bmatrix}\mathbf{CB} & \mathbf{CAB} & \dots & \mathbf{CA}^{r-2}\mathbf{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}(t-r+1) \\ \vdots \\ \boldsymbol{u}(t)\end{bmatrix} + \mathbf{CA}^{r-1}\boldsymbol{x}(t-r+1) \quad (23)$$

where the last term $x(t-r+1)$ is found with the following equation

$$\boldsymbol{x}(t-r+1) = [\mathbf{A} - \mathbf{KC}]\boldsymbol{x}(t-r) + \begin{bmatrix}\mathbf{K} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{y}(t-r) & \boldsymbol{u}(t-r)\end{bmatrix} \quad (24)$$

These equations are simply the extensions of equations (20) and (21) for a r prediction horizon. The limit case of a prediction is found when the horizon tends to infinity. In this case, the prediction outputs are the same as the simulation outputs.

3 RESULTS

The results obtained for the model simulation and prediction are now presented in the following sections. First, the time histories of the outputs measured from flight test and estimated outputs for a typical simulation case are presented. Second, the quantitative results are presented and the performances of the simulation and the predictions for different prediction horizons are compared. This section compares the results obtained by using the subspace system identification method alone with those obtained by using the same method improved with the Levenberg–Marquardt optimization algorithm.

3.1 Typical simulation model outputs for different manoeuvres

The results presented in Figs 4(a) to (d) were obtained from the model simulation (Fig. 2) where the rotorcraft states (u, v, w, p, q, r) are available from flight tests and the estimated outputs are simulated at each time step. In other words, in the following figures, the prediction horizon is infinite. In each figure, the model outputs given in middle are in a colour, whereas the tolerance bands (upper and lower) are in different colours.

The tolerance band is set to 3 per cent of the maximum torque and was selected according to the guidelines of the FAA flight simulator qualification advisory circular [**22**]. Since the FAA does not specify the tolerance band neither for the engine torque nor the tail rotor torque, a tolerance band of 3 per cent is used in this paper for these quantities. For the main rotor speed, the FAA usually specifies a tolerance band of 1.5 per cent, but since the error between the model outputs and the data outputs is minor, the tolerance band for the main rotor speed was set to 0.5 per cent in the following figures.

Figures 4(a) to (d) show the results for collective input excitations, while Figs 5(a) to (d) show the results for longitudinal cyclic excitation. Figures 6(a) to (d) show the results for lateral cyclic excitation and Figs 7(a) to (d) show results for pedal excitation. For each excitation, the time histories of the main rotor torque, tail rotor torque, engine torque, and main rotor speed are displayed.

The results shown in these figures are obtained from a model identified with the subspace system identification method and further optimized with the Levenberg–Marquardt minimization algorithm. These results are obtained for the second flight condition characterized by 49 records used to generate the model and 20 records used to validate it (as seen in Table 1). In this section, the results obtained for 4 out of 20 records used for validation is given.

Figures 4(a) to (d) show the outputs time histories following a 2311 multi-step input collective control for the next flight condition: mean altitude = 9323 ft; gross weight, GW = heavy; centre of gravity, CG position = Aft; mean true airspeed, TAS = 113 knots; rate of climb, ROC = −728 ft/min. These figures clearly show that the simulated outputs lay within the FAA tolerance bands for a simulation. One can also see that following a collective step input, the main rotor torque reaches a peak at a certain value (Fig. 4(a)) and oscillates towards an equilibrium position. This high-frequency oscillation comes from vibrations in the helicopter transmission due to the sudden nature of the control input.

In some cases, a larger error may appear for a few hundredths of a second if the model outputs are slightly out of phase with respect to the flight tests data. Figures 5(a) to (d) display the similar types of outputs following a longitudinal cyclic 2311 multi-step input for the following flight test conditions: mean altitude = 9167 ft; gross weight, GW = heavy; centre of gravity, CG position = Aft; mean true airspeed; TAS = 79 knots; rate of climb,

ROC = 367 ft/min. These figures clearly show that the model is within the tolerance bands. Figures 6(a) to (d) show the outputs following a lateral cyclic 2311 input for the next flight condition: mean altitude = 9581 ft; gross weight, GW = heavy; centre of gravity, CG position = Aft; mean true airspeed, TAS = 71 knots; rate of climb, ROC = 915 ft/min. Finally, the outputs following a pedal 2311 input are shown in Figs 7(a) to (d) for the following flight conditions: mean altitude = 9011 ft; gross weight, GW = heavy; centre of gravity, CG position = Aft; mean true airspeed, TAS = 81 knots; rate of climb, ROC = 277 ft/min.

As shown in Figs 7(a) to (d), the pedal inputs have influences on each of the outputs, particularly on the tail rotor torque. All results displayed so far are found to be within the FAA tolerance bands. The results

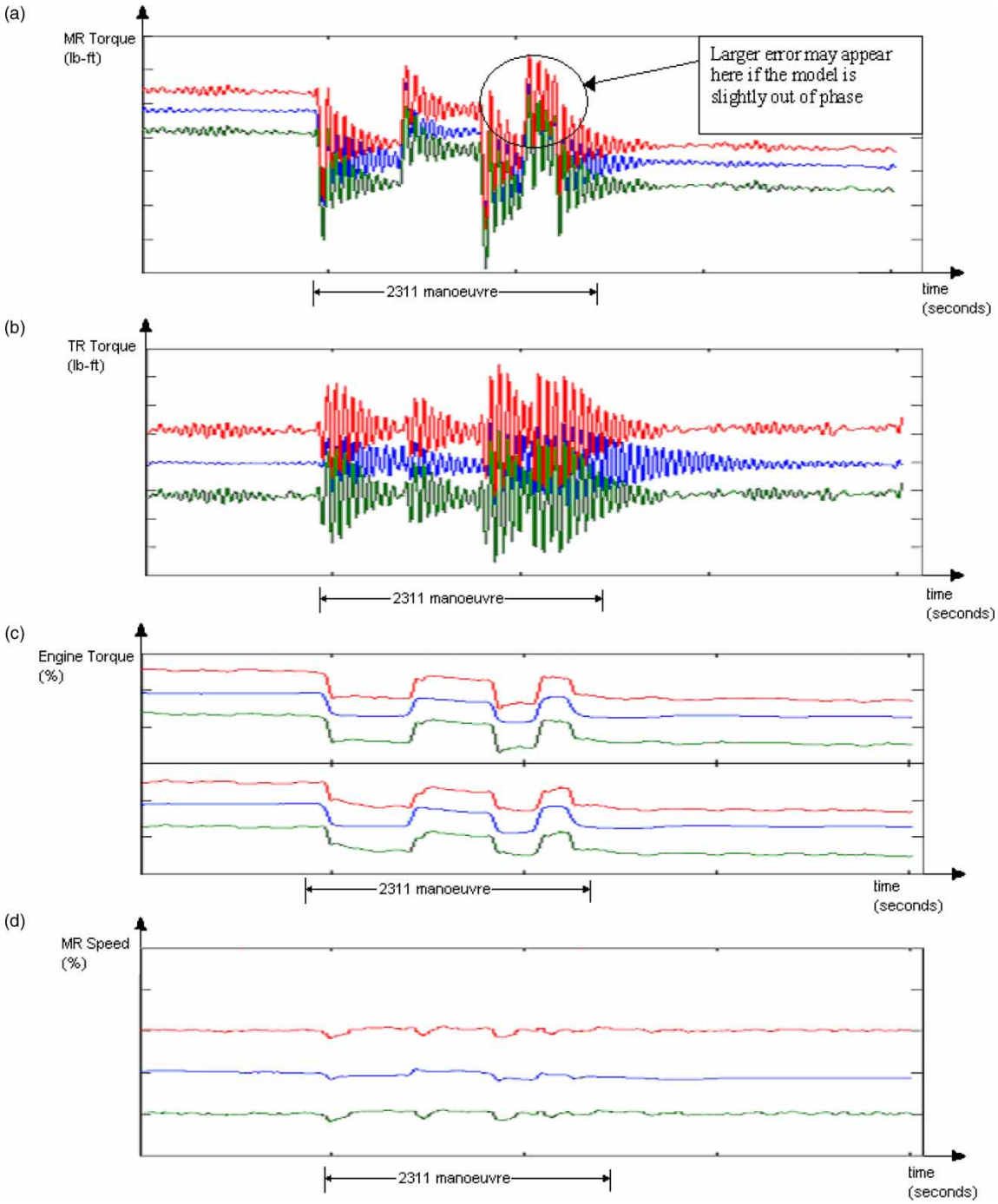

Fig. 4 The time histories following a collective 2311 input (3 per cent tolerance band): (a) main rotor torque; (b) tail rotor torque; (c) engines 1 and 2 torques; and (d) main rotor speed (0.5 per cent tolerance band)

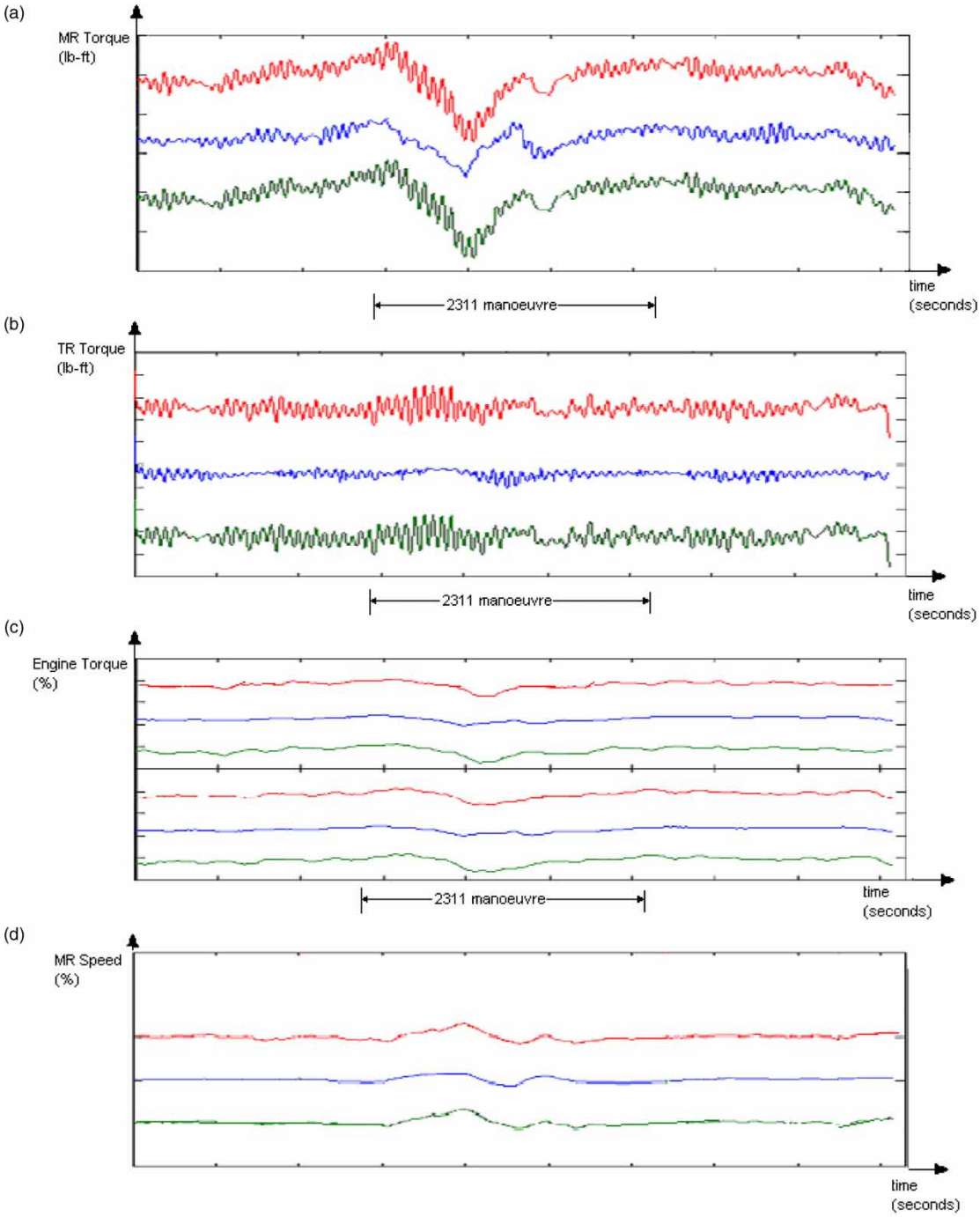

Fig. 5 The time histories following a longitudinal cyclic 2311 input (3 per cent tolerance band): (a) main rotor torque; (b) tail rotor torque; (c) engine torque; and (d) main rotor speed (0.5 per cent tolerance band)

represented in the previous figures are representative only to four validation time records. In the framework of this research, however, the model is validated by using 138 validation records. The quantitative results of these records, as well as the effects of using different prediction horizons are discussed in detail in the following section.

3.2 Quantitative simulation model performance

This section summarizes the results obtained for the time history of 138 validation records. Tables 2 to 5 show the results for a pure simulation that was implemented according to the architecture presented in Fig. 2. The values (per cent) of the mean error,

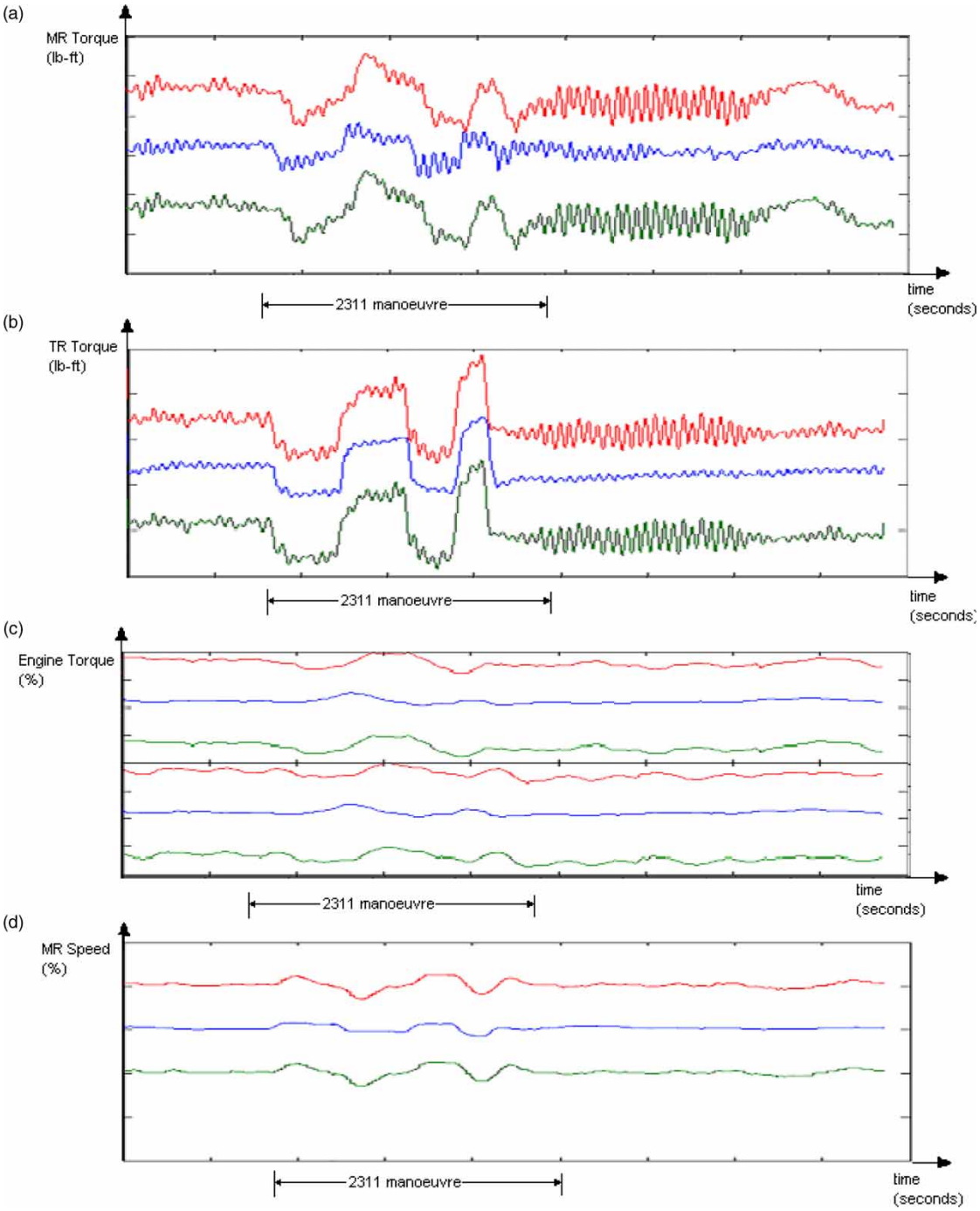

Fig. 6 The time histories following a lateral cyclic 2311 input (3 per cent tolerance band): (a) main rotor torque; (b) tail rotor torque; (c) engine torque; and (d) main rotor speed (0.5 per cent tolerance band

maximum error, and the percentages of the time when the model's error is greater than various tolerances bands (1 to 5 per cent) are displayed. Results are presented for the subspace method only and for the subspace method followed by an optimization.

Table 2 clearly show that even though the results are reasonable when using the subspace system identification method, they can further be improved by refining the resulting parameters with an optimization. In both cases, the mean error is minor (1.22 per cent when using the subspace method only and 0.85 per cent when refining it with an optimization). According to the FAA guidelines, a simulation model is satisfactory if it lies within a 3 per cent tolerance band.

Table 2 also shows that the model is out of tolerance 8.11 per cent of the time when the subspace method is

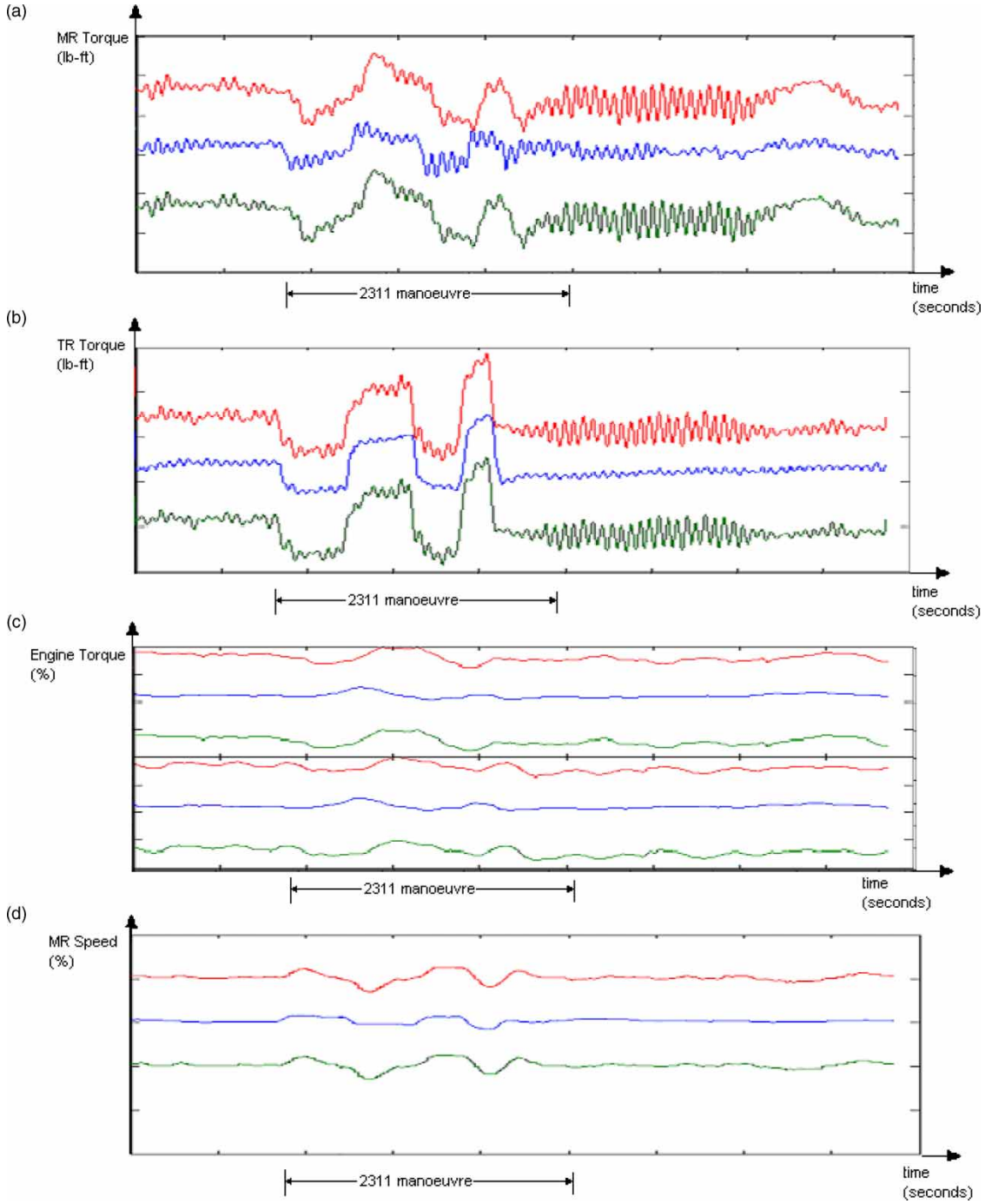

Fig. 7 The time histories following a pedal 2311 input (3 per cent tolerance band): (a) main rotor torque; (b) tail rotor torque; (c) engine torque; and (d) main rotor speed

used. However, it is out of tolerance only 2.8 per cent of the time when the optimization is added. It also clearly appears that most of the time, the model does not exceed the FAA's tolerance bands; the error exceeds 5 per cent only 0.33 per cent of the time when the subspace method is used together with an optimization, which means that even though the maximum error seems to be high, it only occurs for a very short period of time.

According to the observations, the maximum error usually occurs during the collective input time history, i.e. when high-frequency oscillations occur in the main rotor torque. If the model output becomes slightly out of phase with the data, a large error may appear for a few hundredths of a second during the oscillation. One conclusion that can be drawn from these results is that for the FAA certification, it is preferable to apply both the subspace method and an

Table 2 Results for a pure simulation of the main rotor torque output

	Subspace method only	Subspace method with optimization
Mean error (%)	1.22	0.85
Maximum error (%)	14.02	12.27
Percentage of record with more than		
1% error	47.18	30.71
2% error	18.84	8.13
3% error	8.11	2.81
4% error	2.53	0.95
5% error	0.74	0.33

Table 3 Results for a pure simulation of the tail rotor torque output

	Subspace method only	Subspace method with optimization
Mean error (%)	0.57	0.48
Maximum error (%)	6.91	6.98
Percentage of record with more than		
1% error	14.49	9.86
2% error	2.49	1.97
3% error	0.76	0.63
4% error	0.24	0.21
5% error	0.06	0.06

Table 4 Results for a pure simulation of the engine torque output

	Subspace method only	Subspace method with optimization
Mean error (%)	0.79	0.60
Maximum error (%)	5.20	3.85
Percentage of record with more than		
1% error	28.4	17.7
2% error	3.99	2.30
3% error	0.19	0.10
4% error	0.03	0.00
5% error	0.01	0.00

optimization algorithm in order to obtain a sufficiently good model. However, using the subspace method provides a good starting point.

The first observation that can be drawn from Tables 3 to 5 is that the maximum and mean errors decrease (or improve) constantly, which is probably due to the fact that the dynamic system to model is more complex for the main rotor torque (Fig. 4(a)) than for the other outputs because it has high-frequency oscillations and a damping that vary with time.

Table 5 Results for a pure simulation of the main rotor speed output

	Subspace method only	Subspace method with optimization
Mean error (%)	0.1	0.08
Maximum error (%)	1.68	1.61
Percentage of record with more than 1% error	0.32	0.28

The tail rotor torque and engine torque outputs are shown in Tables 3 and 4, where a tolerance band of 3 per cent is assumed, and it is found that there is no difference between out-of-tolerance records percentage when using the subspace method alone and out-of-tolerance record percentage when also resorting to an optimization. The same remark is true for the main rotor speed (Table 5) where, in this case, the FAA tolerance band is 1.5 per cent, which means that the subspace method is probably sufficient to model these outputs. It can also be observed from the results that if the tolerance band was set very tight (for example, 1 per cent), there would be more significant benefit in term of the amount of time the model spends out of tolerance in improving the estimates of the parameters with an optimization algorithm than if the tolerance band was higher. Therefore, combining optimization to the subspace method allows the improvement (minimization) of the error mainly when is already low and the optimization has a lesser influence upon the parts of the records with a larger error. In the following section, the performance of the above-discussed system upon using a prediction method will be analysed.

3.3 Quantitative prediction model performances

Tables 6 to 9 show the model's performance when implemented as a prediction tool, for the same flight conditions similar to Tables 2 to 5. For each helicopter output, the mean error and the overall percentage of the time history records for which the error is greater than 3 per cent are shown with respect to the prediction horizon. The model's performance was evaluated for prediction horizons of 0.12, 0.26, and 0.5 s, which are multiples of the data sampling rate. The results of a pure simulation are also presented in each table so as to compare them with the prediction results.

Several observations can be drawn from these tables. In each case, as anticipated, the mean error decreases when the prediction horizon is decreased. Since a simulation is mathematically equivalent to a prediction with an infinite horizon, all the prediction results are better than the simulation results. For example, the main rotor torque model identified with the subspace method and further refined with an optimization has a mean simulation error of 2.81 per cent (Table 5), but when the state-space model is used to give a prediction of 0.5 s ahead from the known flight test data, the mean error drops to 0.56 per cent. The mean error reduction is due to the fact that any prediction error in the model will last

a maximum of half-a-second before its correction by the measurements (from flight tests), whereas in a simulation, a modelling error can affect the outputs for a long period of time. For the same reason, the mean error decreases when the prediction horizon decreases. These tables also clearly show that the mean prediction error is always very small and never exceeds 0.6 per cent for any prediction horizon.

The second kind of results shown in these tables represents the percentage of time for which the model error is greater than 3 per cent. This percentage was chosen because it was used in the above discussion on simulation application of the model. For control system applications, there is however no fixed tolerance. The evaluation is rather based on the overall control system's performance, which is assessed by the pilot. Tables 6 to 8 reveal that a decreasing prediction horizon results in a lower percentage of the model outputs with an error greater than 3 per cent. There is no such result in Table 9 as the model error is always below this tolerance. In Tables 6 to 9, the worse results pertain to the main rotor torque whose prediction is out of the tolerances during 1.2 per cent of the time when the prediction horizon is 0.5 s and during 0.7 per cent of the time when the prediction horizon is 0.12 s (Table 6).

Finally, each table shows that both the models identified with the subspace method alone and with the subspace method followed by an optimization, gave excellent results. In the case of the main rotor torque outputs, there is almost no difference between the models identified with these two methods. Recall that for a pure simulation, a significant model performance gain is realized by using an optimization. For the tail rotor torque outputs (Table 7), resorting to an optimization allows to slightly reduce the model's mean error, but oddly, it is out of tolerance a little longer than with the sole use of the subspace method,

Table 6 Mean error prediction for the main rotor torque and percentage of records with an error rate greater than 3 per cent with respect to the prediction horizon

	Subspace method only		Subspace method followed by optimization	
Prediction horizon (seconds)	Mean error (%)	Percentage of record with more than 3% error	Mean error (%)	Percentage of record with more than 3% error
Infinite (simulation)	1.22	8.11	0.85	2.81
0.5	0.56	1.20	0.55	1.25
0.26	0.41	0.7	0.41	0.7
0.12	0.39	0.65	0.38	0.58

Table 7 Mean error prediction for the tail rotor torque and percentage of records with an error rate greater than 3 per cent with respect to the prediction horizon

	Subspace method only		Subspace method followed by optimization	
Prediction horizon (seconds)	Mean error (%)	Percentage of record with more than 3% error	Mean error (%)	Percentage of record with more than 3% error
Infinite (simulation)	0.57	0.76	0.48	0.63
0.5	0.41	0.36	0.39	0.38
0.26	0.32	0.20	0.31	0.22
0.12	0.36	0.30	0.29	0.16

Table 8 Mean error prediction for the engine torque and percentage of records with an error rate greater than 3 per cent with respect to the prediction horizon

	Subspace method only		Subspace method followed by optimization	
Prediction horizon (seconds)	Mean error (%)	Percentage of record with more than 3% error	Mean error (%)	Percentage of record with more than 3% error
Infinite (simulation)	0.79	0.19	0.60	0.10
0.5	0.42	0.04	0.42	0.07
0.26	0.25	0.03	0.15	0.00
0.12	0.20	0.00	0.10	0.00

Table 9 Mean error versus prediction horizon for the main rotor speed

Prediction horizon (seconds)	Subspace method only	Subspace method followed by optimization
	Mean error (%)	Mean error (%)
Infinite (simulation)	0.10	0.08
0.5	0.04	0.03
0.26	0.02	0.02
0.12	0.02	0.01

when the prediction horizon is high. The error also slightly increases when a 0.12 s prediction horizon is used, with respect to the error calculated with a 0.26 s prediction horizon (Table 7).

The use of an optimization slightly improves the main error between the model and the data for the engine torque and main rotor speed outputs however it may affect the percentage of records with an error greater than 3 per cent.

4 DISCUSSION

The following inferences can be drawn from the results.

4.1 For the model simulation implementation

1. All the simulation results, especially the main rotor torque outputs (Fig. 2), were improved when the subspace identification method was combined with an optimization. For the main rotor torque outputs, which are termed as worse result, 2.81 per cent (smaller than 3 per cent) of the simulated outputs were out of tolerance instead of 8.11 per cent when the subspace method alone was used.
2. The addition of an optimization to the subspace method may reduce the mean error, however, the benefits of an optimization decrease when the subspace method error is already small, which is especially true for the main rotor speed.

4.2 For the model prediction implementation

1. The mean error and the percentage of time when the error is greater than 3 per cent is very low for a prediction. Furthermore, as could be expected, the prediction error decreases when the prediction horizon decreases.
2. The values of the parameters are refined by the subspace method with an optimization, and therefore allow the improvement of the results. The errors are already found to be very low by using the subspace method alone – the reason for which optimization is not needed.

5 CONCLUSIONS

In summary, it can be concluded that the dynamics of the main rotor torque, tail rotor torque, engine torque, and main rotor speed can be properly estimated by using a state-space model. The subspace system identification method is an efficient non-iterative method that can provide a reasonably good estimation of these helicopter parameters without requiring any initial guess or any prior knowledge of the system's dynamics. The model identified with the subspace method can also be successfully refined with a Levenberg–Marquardt minimization algorithm.

The results were assessed according to the mean error and the percentage of times that the model error was greater than a certain tolerance. For a flight simulation certification, the FAA's tolerance band is 3 per cent for the main rotor torque and 1.5 per cent for the main rotor speed. Since there is neither a specified tail rotor torque tolerance, nor any specified engine torque tolerance at present, it is decided to use a 3 per cent tolerance band in this study. The results show that for the tail rotor torque, the engine torque and the main rotor speed, the modelling error is very small when the subspace identification method is used alone. The results for the main rotor torque are reasonable, except for 8.11 per cent of the records, which have an error greater than 3 per cent. This percentage was successfully reduced to 2.81 per cent with optimization. Note that the simulation results shown in this paper are for the records set aside for the validation, the results for records used for the identification are generally slightly better with mean errors up to 25 per cent lower. This is not surprising since these records were used to generate the model.

The model was also tested for a prediction implementation. In each case, the modelling error was very low and was reduced as the prediction horizon length was shortened. Using an optimization allowed to slightly improve the results achieved with the subspace method, but the benefit of adding an optimization was minor since the models identified with the subspace method were already excellent. The prediction results discussed in this paper also apply for the records set aside for validation. The mean errors for these records used in the identification are usually about the same or slightly better (up to 5 per cent). In conclusion, the subspace method is a promising method to estimate the helicopter's physical parameters, by means of flight test data, for simulation and control applications.

ACKNOWLEDGEMENTS

This work was possible due to the research funds received from Bell Helicopter Textron and the Consortium for Research and Innovation in Aerospace in Quebec (CRIAQ) in the CRIAQ 3.4 project entitled Development of a new parameter estimation technology for a global helicopter model.

REFERENCES

1 **Jeram, G. J., Nilesh H. S.,** and **Prasad, J. V. R.** Distributing limit protection between autonomous restraint and voluntary tactile cue. In the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 10–13 January 2005, pp. 1–14.

2 **Massey, C. P.** and **Wells, P.** Helicopter carefree handling systems.In Proceedings of the Royal Aeronautical Society Conference on Helicopter Handling Qualities and Control, London, UK, 15–17 November 1988, 17 pp.

3 **Jategaonkar, R. V.** *Flight vehicle system identification: a time domain methodology*, AIAA Progress in Aeronautics and Astronautics Series (Ed. F. K. Lu), 2006, pp. 1–500 (AIAA, Arlington).

4 **Galvao, R. K. H., Hadjiloucas, S., Becerra, V. M.,** and **Bowen, J. W.** Subspace system identification framework for the analysis of multimoded propagation of THz-transient signals. *Meas. Sci. Technol.*, 2005, **16**, 1037–1053.

5 **Lennart, L.** *System identification theory for the user*, 2nd edition, 1999, pp. 340–351 (Prentice Hall, Upper Saddle River, New Jersey).

6 **Lennart, L.** *System identification toolbox for use with Matlab®. User guide version 6*, 2006, pp. 1–416 (The Mathworks Inc., USA).

7 **Howitt, J.** Carefree handling for super-agility. In the American Helicopter Society 2nd Aeromechanics Specialist Conference, Fairfield County, 11–13 October 1995, pp. 4–58.

8 **Menon, P. K., Iragavarapu, V. R.,** and **Walley, M. S.** Estimation of rotorcraft limit envelope using neural networks. In Annual Forum Proceedings of the American Helicopter Society, Alexandria, Virginia, 1996, vol. 2, pp. 1423–1431.

9 **McCool, K., Flitter, L. A.,** and **Hass, D. J.** Development and flight test evaluation of a rotor system load monitoring technology. In the American Helicopter Society 54th Annual Forum, Washington, 20–22 May 1998, pp. 408–418.

10 **Horn, J.** *Flight envelope limit detection and avoidance*. PhD Thesis, Atlanta Georgia Institute of Technology, May 1999, pp. 1–110.

11 **Horn, J., Calise, A. J.,** and **Prasad, J. V. R.** Flight envelope limit detection and avoidance for rotorcraft. *J. Am. Helicopter Soc.*, 2002, **47**(6), 253–262.

12 **Yavrucuk, I., Prasad, J. V. R.,** and **Calise, A. J.** Adaptive limit detection and avoidance for carefree manoeuvring. In the AIAA Atmospheric Flight Mechanics Conference and Exhibit, Montreal, 6–9 August 2001, pp. 1–9.

13 **Yavrucuk, I., Prasad, J. V. R.,** and **Calise, A. J.** Carefree manoeuvring using adaptive neural networks. In the AIAA Atmospheric Flight Mechanics Conference and Exhibit, Monterey, California, 5–8 August 2002, pp. 291–299.

14 **Yavrucuk, I.** and **Prasad, J. V. R.** Adaptive limit margin prediction and control cueing for carefree manoeuvring of VTOL Aircraft. In the American Helicopter Society Flight Controls and Crew System Design Technical Specialist Meeting, Philadelphia, 9–11 October 2002, pp. 1–7.

15 **Vineet, S., Spauling, R., Faynberg, A., Horn, J.,** and **Sahani, N.** Simulation investigation of a comprehensive collective-axis tactile cueing system. In the American Helicopter Society 58th Annual Forum, Montreal, 11–13 June 2002, pp. 559–568.

16 **Horn, J.** and **Sahani, N.** Detection and avoidance of main rotor hub moment limits on rotorcraft. *J. Aircr.*, 2004, **41**(2), 372–379.

17 **Sahani, N. A.** *Envelope protection systems for piloted and unmanned rotorcraft*. PhD Thesis, Pennsylvania State University, December 2005.

18 **Brenner, M.** and **Feron, E.** Wavelet analysis of F/A-18 aeroelastic and aeroservoelastic flight test data. In the 38th AIAA/ASME/ASCE/AHS/ASC Structure, Structural Dynamics and Material Conference Exhibit, 7–10 April 1997, 23 pp.

19 **Jeram, G. J.** Open design for helicopter active control system. In the American Helicopter Society 58th Annual Forum, Montreal, Canada, 2002, 20 pp., available from www.vtol.org/pdf/lich-58.pdf.

20 **Rugh, W. J.** *Linear system theory*, 1993 (Prentice Hall Information and System Sciences Series, New Jersey).

21 **Kalman, R. E.** A new approach to linear filtering and prediction problems. *ASME J. Basic Eng., Ser. D*, March 1990, **82**, 35–45.

22 Helicopter simulator qualification, AC-120-63 FAA Advisory Circular, 1994.

APPENDIX

Notation

A, **B**, **C**, **D**	state-space matrices
coll	collective position
∂	partial derivative operator
$\boldsymbol{e}$	white noise vector
F	second derivative
G	gradient
I	identity matrix
J	cost function
K	noise disturbance matrix
lat	lateral cyclic position
long	longitudinal cyclic position
N	number of data points in the data vector
ped	pedal position
p,q,r	angular velocities along the body axes
t	time
u,v,w	linear velocities along body axes

$\boldsymbol{u}$	system input vector
$\boldsymbol{x}$	system state vector
$\boldsymbol{y}$	output vector
$\boldsymbol{\Gamma}$	observability matrix
Δ	increment in the parameter guess
Δt	time step
$\boldsymbol{\theta}$	parameter in the state-space matrices
λ	Levenberg–Marquardt Parameter

Subscripts

i	ith element of the data vector
j	jth iteration
limit	matrix used to estimate a limit parameter
m	number of inputs
n	number of states or order of the system
o	number of outputs
r	forward prediction horizon
states	matrix used to estimate the helicopter states

Superscripts

$\wedge$	estimated parameter