ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN ENGINEERING
M. Eng.

BY
DE JESUS MOTA, Sandrine

IDENTIFICATION AND VALIDATION OF A MODEL OF THE BELL-427 HELICOPTER FROM FLIGHT TEST DATA WITH A TIME-DOMAIN METHOD

MONTRÉAL, FEBRUARY 20 2009

BOARD OF EXAMINERS (THESIS M. ENG.)

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

Dr Ruxandra Mihaela Botez, Thesis Supervisor
Department of Automated Production Engineering at École de technologie supérieure

Dr Guy Gauthier, Jury President
Department of Automated Production Engineering at École de technologie supérieure

Joey Seto, Eng., External Jury Member
Senior Technical Specialist (Aero & HQ), Bell Helicopter Textron Canada

THIS THESIS HAS BEEN PRESENTED AND DEFENDED

IN FRONT OF A BOARD OF EXAMINERS

ON 15TH OF DECEMBER 2008

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

AKNOWLEDGEMENTS

I would like to express my deep gratitude to Professor Ruxandra Botez who gave me great support through this thesis. For the last year, she gave me the necessary resources and guidance to progress in this research. She gave me the opportunity to be more implicated in the aerospace field by the several conferences we participated.

I benefited very much from the assistance and collaboration of many members of the LARCASE team to carry on every aspects of this thesis. Dr. Michel Nadeau-Beaulieu, Mr. Andrei Popov and Mr. Nicolas Boëly provided me great support in the methods used and the goodness of the results. Mr. Sylvain Tétreault contributed to the compilation of results.

The work presented in this thesis was part of the CRIAQ 3.4 project which was funded by Bell Helicopter Textron (BHT) and the Consortium for Research and Innovation in Aerospace in Quebec (CRIAQ). Additional scholarships were obtained from École de technologie supérieure, which contribute to pursue this thesis without financial worries.

Finally, last but not the least, I would like to thank my family, Carlos, Fernanda and Amélie, and my boyfriend Clément Rollier for all their love support and encouragements through this research.

IDENTIFICATION ET VALIDATION D'UN MODELE DE L'HELICOPTERE BELL-427 A PARTIR DE DONNEES DE VOL AVEC UNE METHODE TEMPORELLE

DE JESUS MOTA, Sandrine

RÉSUMÉ

Une nouvelle technique pour l'identification d'un modèle d'hélicoptère Bell-427 est présentée. Le modèle est identifié et validé pour 22 conditions de vol, chacune étant définie par une altitude variant de 3000 pieds à 6000 pieds, une vitesse variant de 30 nœuds à 115 nœuds, un hélicoptère lourd et un centre de gravité longitudinal étant en avant et en arrière de l'hélicoptère. Pour identifier les modèles, des commandes de type 2-3-1-1 sont exécutées par le pilote dans le but d'exciter tous les modes du système. Pour identifier le mouvement longitudinal et latéral de l'hélicoptère, un nouveau jeu de données est construit en concaténant les données reliées à chaque commande du pilote. Le modèle construit est sous la forme d'espace d'état où les vitesses linéaires et angulaires, et les angles d'Euler représentent les états du système et les accélérations linéaires sont les sorties du système.

Deux problématiques majeures sont résolues dans ce mémoire. La première concerne l'équation d'état. Une relation de récurrence entre les états est définie. Puis, une optimisation basée sur la théorie des réseaux de neurones est réalisée et un réglage manuel et automatique des conditions initiales des états est fait afin de satisfaire les règles de la FAA. La seconde problématique est reliée à l'équation de sortie. À cause des effets aléatoires lors de la prise de mesure, les équations classiques du mouvement ne donnent pas de résultats satisfaisants pour observer les sorties du système à partir des états du système. Deux méthodes (une linéaire et une non-linéaire) sont présentées et une comparaison des trois méthodes est réalisée pour déterminer la méthode la plus appropriée. Dans le but d'effectuer une comparaison propre, une liste de critères est formée dont chacun est associé à un coefficient de pondération selon son importance. Avec cette comparaison, les sentiments subjectifs du pilote sont pris en compte.

Chaque modèle identifié à une condition de vol doit être validé par trois différents tests de vol non utilisés pour l'identification. De plus, chaque signal estimé doit rester dans la bande de tolérance définie par la FAA selon le type de mission de l'hélicoptère. Pour les tests de validation, les signaux estimés doivent rester dans les bandes de tolérance un minimum de trois secondes.

De bons résultats sont obtenus en boucle ouverte et en boucle fermée pour l'estimation des signaux d'état. La méthode linéaire s'est avérée la meilleure pour observer les sorties du système à partir des signaux d'état. Les matrices obtenues à chaque condition de vol sont interpolées pour obtenir le modèle de l'hélicoptère pour toutes les conditions de vol. Les tests de vol non utilisés pour l'identification et la validation du modèle sont utilisés pour l'interpolation du modèle. Les signaux estimés satisfont les règles de la FAA.

La généralisation du modèle pourrait être encore affinée ce qui permettrait d'implémenter le modèle dans un simulateur, utilisé pour l'entraînement des pilotes et de l'appliquer comme base pour l'étude d'autres hélicoptères.

Mots-clés : identification, validation, hélicoptère, FAA.

IDENTIFICATION AND VALIDATION OF A MODEL OF THE BELL-427 HELICOPTER FROM FLIGHT DATA TESTS WITH A TIME-DOMAIN METHOD

DE JESUS MOTA, Sandrine

ABSTRACT

A new technique for the Bell-427 helicopter model identification from flight data tests is here presented. A helicopter model is identified and validated for 22 flight conditions, which is defined by an altitude varying between 3,000 ft and 6,000 ft, a speed varying from 30 knots to 115 knots, a helicopter loading with a heavy gross weight and a longitudinal aft and forward center of gravity. To identify the models, 2-3-1-1 multistep control inputs are performed by the pilot to excite all helicopter modes. In order to identify the global motion of the helicopter, a new data set is constructed by concatenating the data related to each of the four control inputs. The model is represented in the state space form where linear and angular velocities and Euler angles are the state variables and linear accelerations are the output variables.

Two major problems are solved in this thesis. The first problem concerns the state equation. A recurrence relationship is set up. Then, an optimization based on neural network theory is performed and a manual and automatic tuning of the initial state conditions is done in order to satisfy the FAA rules. The second problem regards the output equation. Because of random effects when gathering data, classical equations of motion do not give good enough results to observe the system outputs from the system states. Thus, two other methods (one linear and one nonlinear) are presented and a comparison among the three methods is performed in order to find the powerful method. To realize a proper comparison, a list of criteria is done with weighted coefficients associated to each criterion. By use of this comparison, subjective feelings of the pilot are considered.

Each identified model in a flight condition is validated by three different flight tests not used to identify the model. Then, each estimated signal has to remain in a tolerance margin defined by the FAA according to the flight mission of the helicopter. Finally, for the validation tests, estimated signals must be within the tolerance margins for at least three seconds.

Good results are obtained in open-loop and closed-loop for the states identification. The best final score is obtained with the linear method in order to observe the system outputs from its states. The obtained matrices are interpolated to obtain the model for any flight condition. Flight tests not used for the identification and validation of the model are used for the model interpolation. The estimated signals satisfy the FAA rules.

The model generalization could be improved in order to implement it in a simulator, which would be use for the pilot training, and apply it as a basis for other helicopters study.

Key words: identification, validation, helicopter, FAA.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

ABBREVIATIONS

BFGS	Broyden-Fletcher-Goldfarb-Shanno
CG	Center of Gravity
CL	Closed-loop
DOF	Degree Of Freedom
FAA	Federal Aviation Administration
KKT	Karush-Kuhn-Tucker
L / H / A / F	Light / Heavy / Aft / Forward
MISO	Multiple Input Single Output
MIMO	Multiple Input Multiple Output
NN	Neural Network
OL	Open-loop

SYMBOLS AND UNITS

A, B, C, D	Matrices describing the discrete state-space model
A_x, A_y, A_z	Linear accelerations
Corr	Correlation coefficient
Cov	Covariance
coll	Collective command position
deg	Degree
deg/s	Degree per second
e	Error vector
FIT	Fit coefficient
ft	Feet
ft/min	Feet per minute
ft/s	Feet per second
ft/s^2	Feet per second squared
$\dot{h}$	Altitude rate
$[I]$	Inertia matrix
Id	Identity matrix
in	Inch
J	Cost function
lat	Lateral cyclic command position
long	Longitudinal cyclic command position
lb	Pound
m	Output number
n	State number
U	Theil's coefficient
u, v, w	Linear velocities
u	Input vector
p, q, r	Rates
s	Second
ped	Pedals command position
Var	Variance
x	State vector
y	Output vector
γ	Forgetting factor
λ	Lagrangian parameter
σ	Standard deviation
φ, θ, ψ	Attitude angles
η	Learning rate
μ	Momentum

Superscripts

^	Estimated
T	Transpose

Subscripts

des	Desired
in	Input
new	New value
old	Old value
out	Output
real	Measured

INTRODUCTION

The increasing need for high-performance aircraft or rotorcraft has initiated a higher use of system identification methods. Such mathematical models can be conceived for flight simulator, flight control system or handling qualities applications. In this thesis, a Bell-427 helicopter global model is built from flight test data.

Two problems regarding the model structure are solved. The first one regards the model degree. Several authors in the literature say that conventional six degrees of freedom (DOF) models were adequate to describe a rotorcraft dynamics under the hypothesis that the helicopter is a rigid body. Can the Bell-427 helicopter be considered as a rigid body so that a six DOF model is sufficient to characterize its dynamics? To compute this model, a state space linear system is conceived. The state variables are the linear and angular velocities and the Euler angles. The collective, longitudinal cyclic, lateral cyclic and pedals pilot controls are used as the system inputs. This state equation is studied in open-loop i.e. the states evolution is a function of the measured states and pilot controls. A closed-loop study is done in order to define the states evolution as a function of the pilot controls only. The last study is the more realistic one because in reality, only the pilot has a control on the helicopter dynamics he is flying. The closed-loop study is set up with the results found in open-loop. Then, an optimization based on neural network theory and a manual and automatic tuning of initial states conditions are used to clear the system to diverge and to increase the model efficiency.

The second problem regards the system outputs which are the linear accelerations. Generally, the classical equations of motions are used to describe a system. However, this method does not give accurate results to observe the system outputs from the state variables so that two methods are set up to obtain a powerful method. A linear method and a nonlinear method based on fuzzy logic are used. To properly compare the results obtained with the three methods (classical, linear and nonlinear), a list of criteria is considered. Weights are attributed to each criterion to underline the importance of all of them.

This procedure is followed for each flight condition, which is defined by an altitude, a speed, a center of gravity position and a helicopter loading. The pilot flies the helicopter for different missions which are level flight, ascending, descending, and autorotation flight. Level flight tests are used to identify the models. To evaluate the quality of the models, the Federal Aviation Administration (FAA) established rules. For each variable and mission, tolerance margins are defined. In order to validate the identified model, the model is computed for another set of data in input. The model responses are compared to the measured ones. If they are "too" different, then the model is not robust enough and the model should be again designed.

The last step of the model identification is its generalization in the flight envelope. Indeed, the mathematical models are valid only for a specific flight condition so that an interpolation between each flight condition must be done in order to obtain the system characteristics at any flight condition. In this thesis, an interpolation between the studied flight conditions is done according to the speed.

This thesis is organized as follows: in Chapter 1, a literature review of system identification process is presented. A direct link between literature review and the thesis problems is set up. In Chapter 2, the state equation is determined with a recurrence method in open-loop. The model optimization and the tuning of the initial conditions are then described in order to define the closed-loop model. In Chapter 3, the output equation is presented. The theories of the classical equations of motions, the fuzzy logic nonlinear method and the linear method are detailed. The comparison between results obtained with the three methods is presented. In Chapter 4, the global model is obtained by models interpolation as function of speeds. Due to space restraints, only results obtained for the flight condition HA6000ft-50kts (heavy gross weight, aft center of gravity, altitude of 6,000 ft and speed of 50 knots) are shown. The results of all flight conditions are put in appendix.

CHAPTER 1

SYSTEM IDENTIFICATION PROCESS

1.1 Definition

According to Jategaonkar (2006), system identification is the process of determination of an adequate mathematical model, described with differential equations containing unknown parameters which have to be determined from measured data such that the model response matches adequately the measured system responses. By "adequately", it is introduced a notion for which no "perfect fit" would be possible, as for real cases or products. For this reason, the Federal Aviation Administration (FAA) has set up rules in order to define the term "adequate".

System identification needs to be followed by a step called "model validation" to assess the model fidelity. If it turns out that the identified model does not meet the requirements, the model structure should be changed and the whole process should be repeated.

System identification provides an overall understanding of the flight vehicle's dynamics and yields an accurate and comprehensive data base for flight simulators, which are extensively used for pilot training and to minimize risk during experimental testing, which is very costly. Hamel and Kaletka (1997) highlighted four important aspects in system identification:

(1) The *maneuver*: The input signals have to be optimized in their spectral composition in order to excite all response modes from which parameters are to be estimated because "If it is not in the data, it cannot be modeled";

(2) The *measurements*: The measuring of experimental data (i.e. from a real process), sensors errors and measurement noise complicate the identification process;

(3) The *model structure*: Depending on the model application, different structures can be chosen to describe the model;

(4) The *identification method*: Various techniques are available to identify the model.

These four aspects are usually called the "*Quad-M*" requirements.

1.2 Manoeuvres

The data are sorted by different categories: (1) *the pilot command* which can be a collective, a longitudinal cyclic, a lateral cyclic or a pedal control, (2) *the flight mission* which can be a level flight, an ascending flight, a descending flight or an autorotational flight, (3) *the helicopter loading* depending on the gross weight and the longitudinal center of gravity (CG), (4) *the altitude* which varies between 3,000 ft and 6,000 ft and (5) *the speed* which is between 30 knots and 115 knots.

In this section, the way in which the flight tests data are sorted is defined.

1.2.1 Pilot command

A pilot manipulates the helicopter flight controls in order to correctly fly the helicopter. As previously said, four commands are used:

(1) The *collective* command changes angle of all main rotor blades at the same time and independently of their positions in order to control the altitude;

(2) The *longitudinal cyclic* command varies the main rotor blades pitch in order to control the altitude or to move forward or backward;

(3) The *lateral cyclic* command varies the main rotor blades pitch in order to move sideways;

(4) The *anti-torque pedals* command changes the pitch of the tail rotor blades, increasing or reducing the thrust produced by the tail rotor and causing the nose to yaw in the applied pedal direction.

The four commands are presented in the following figure:

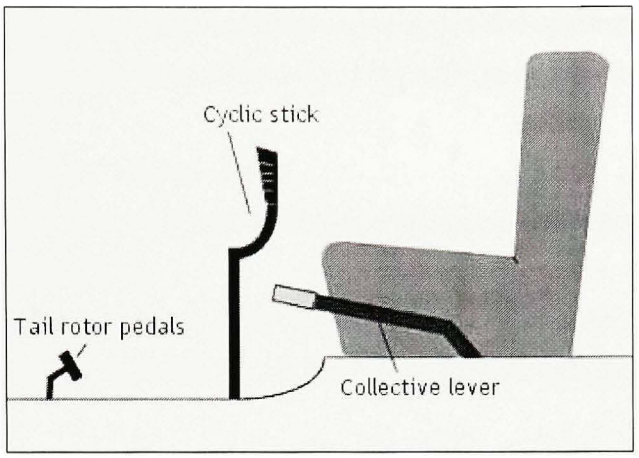

Figure 1.1 Localization of the commands in a helicopter.
(http://community.bistudio.com/wiki/Image:Helicopter-controls.jpg)

Gathered data basically limits, both in terms of scope and accuracy, the model development and parameter estimation were described (Jategaonkar, 2006). One of the most important aspects of data gathering is the choice of adequate inputs form to excite the aircraft motion in some *optimum* sense. Milliken (1951) presented the optimum input as the input which excites the best the frequency range of interest.

Generally, dynamic motion is excited by applying control pulse, step, multistep, or harmonic inputs. A variety of manoeuvres was usually necessary to excite dynamic motion about different axes using independent inputs on every control (Jategaonkar, 2006). These different control inputs are presented in the following figure:

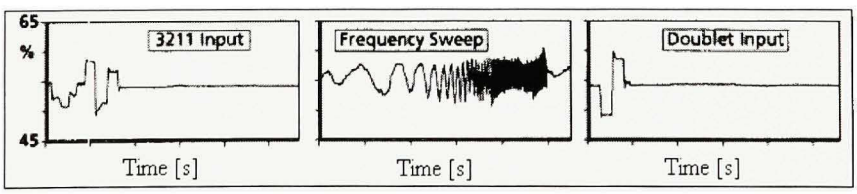

Figure 1.2 Control inputs.
From Hamel and Kaletka (1996), p. 263

The 3-2-1-1 input has a much wider spectrum compared to the spectrum of the impulse or doublet inputs. The main advantage of the 3-2-1-1 input lies in its simplicity and its ability to manually realize it.

Two minor aspects of 3-2-1-1 inputs are:

(1) Their asymmetry about the trim deflection, and as a consequence, they have nonzero energy at zero frequency;

(2) The first step being of larger duration, namely three units of Δt may lead to motions far from the initial trim condition before the application of following steps.

These undesirable effects can be minimized by modifying the input amplitudes or by time twisting the steps. The 2-3-1-1 input prevents the vehicle from going far from the trim condition, before the application of the larger duration time step.

In order to know which pilot command is relative to a flight test, the four commands are plotted which allows concluding the primarily command used by the pilot. Indeed, it is easy to distinguish a 2-3-1-1 command from another input type.

In the following figure, the flight test four controls are plotted:

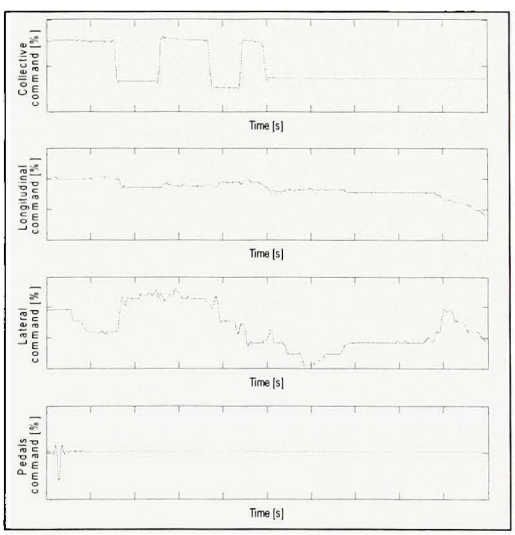

Figure 1.3 Command pilot selection.

By observing Figure 1.3, there is no ambiguity about the pilot controls during this flight test. The pilot performed a *collective* command. The four steps with different time lengths for the 2 Δt, 3 Δt, Δt and Δt of this command type are visible.

Is noted that the other commands are not constant during the flight test, which is due to the high correlation between all helicopter command inputs. This can be explained by the fact that the aerodynamically behaviour of a helicopter cannot be split into to a longitudinal and a lateral motion as for an aircraft study.

In order to identify the helicopter model, a new data set is constructed. The four pilot commands are concatenated so that, for all signals, the first quarter shows the influence of the collective primary control, the second quarter shows the influence of the longitudinal cyclic control, the third quarter shows the influence of the lateral cyclic control and the last quarter shows the influence of the pedals control, please see next figure:

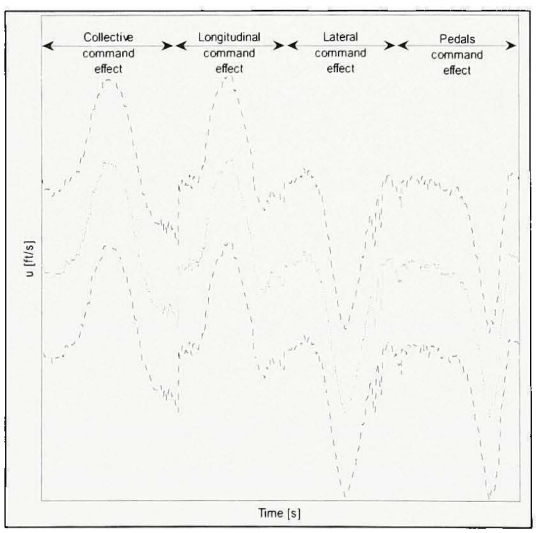

Figure 1.4 Example of the effect of the four concatenated commands.

1.2.2 Flight mission

The flight missions are split into four categories:

(1) The *flight level*, when the altitude rate denoted by $\dot{h}$ is between -750 ft/min and +750 ft/min. The altitude rate is defined by the following expression:

$$\dot{h}_{[ft/min]} = \frac{[h(\text{flight test end}) - h(\text{flight test start})]_{[ft]}}{[\text{flight test duration}]_{[s]}} \times 60 \tag{1.1}$$

All flight tests which enter in this category are arranged in the "Level Flight, ±500 ft/min" list.

(2) The *ascending flight*, when the altitude rate is higher than 750 ft/min. All flight tests entering into this category are arranged in the "+1000ft/min" list.

(3) The *descending flight*, when the altitude rate is lower than -750 ft/min and the engines are on. All flight tests entering into this category are arranged in the "-1000ft/min" list.

(4) The *autorotational flight*, when the altitude rate is lower than -750ft/min and the engines are off. All flight tests entering into this category are arranged in the "Autorotation" list.

1.2.3 Helicopter loading

Two parameters must be known to sort the flight test data according to the helicopter loading:

(1) The *gross weight* which can be "light" or "heavy". If the gross weight is lower than 5,600 lb, then the helicopter is considered as light (L). If the gross weight is higher than 6,000 lb, then the helicopter is heavy (H);

(2) The *longitudinal center of gravity* which can be "aft" or "forward". If the longitudinal center of gravity (CG) is far from the helicopter nose than 224 in, then the longitudinal CG is set to the aft (A) of the helicopter. If the longitudinal CG is less far than 220 in, then it is set to the forward (F) of the helicopter.

Thus, four conditions are presented: (L), (H), (A) and (F). Four combinations are considered in order to study the aerodynamically behaviour of the helicopter: HA, HF, LA and LF. In this thesis, only the results obtained from HA and HF flight conditions for altitudes varying from 3,000 ft to 6,000 ft are presented. The following figure shows the studied test points:

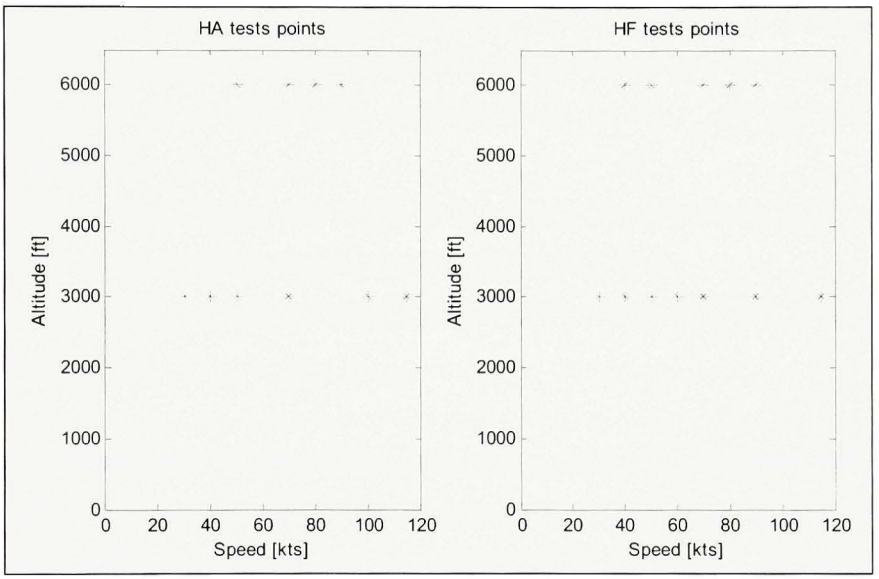

Figure 1.5 Tests points studied.

In this thesis, the helicopter is modeling for 22 flight conditions dependent on the flight mission, helicopter loading, speed and altitude.

1.3 Measurements

According to Hamel and Kaletka (1997), the measurement of certain variables depends on the model application and the identification method. For some techniques, measurements of the state vector variables are required. According to the model degree, a set of variables needs to be measured. For a six DOF rigid body model identification, the controls, the speeds, the linear accelerations, the rates and attitudes must be measured.

1.4 Model structure

The model structure determines the difficulty degree in defining the unknown parameters. According to Tischler and Remple (2006), the rotary-wing aircraft dynamics can be described with four main models:

(1) The *quasi-steady lateral directional* model (three DOF);

(2) The *quasi-steady longitudinal* model (three DOF): Both three DOF models are based on the assumptions that longitudinal and lateral directional degrees of freedom are not coupled. It could be a good application for tilt-rotor and the tandem-rotor applications;

(3) The *quasi-steady* model (six DOF): In this model, the rotor steady-state response is presented as an equivalent quasi-steady fuselage derivative;

(4) The *hybrid fully coupled* model (13 DOF): This model is highly accurate for all rotorcrafts.

According to Hamel and Kaletka (1997), most of the system identification work was still devoted to the determination of parametric fully coupled six DOF linear derivatives models, which were considered appropriate for the description of the rigid body dynamics for the low and medium frequency range.

1.5 Identification modelling methods

Two methods are mainly used to identify a model from flight data tests: a *frequency*-domain method, and a *time*-domain method. A comparison of the frequency and time-domain methods was given by Tischler and Kaletka (1987).

Frequency-domain identification used spectral methods to determine frequency responses between selected input and output pairs. Then, least-squares fitting techniques were used to obtain closed-form analytical transfer-function linear input-to-output models.

Time-domain identification required the selection of a state-space model structure, which may be linear or nonlinear. Model parameters were identified by least-square fitting of the response time-histories or by maximum likelihood methods.

Tischler and Kaletka (1987) presented the main advantages and inherent limitations of the frequency and time-domain methods. Then, Tischler and Remple (2006) defined the frequency sweep as the typical input for frequency-domain and the multistep input for time-domain methods.

According to Bohlin (2006), three types of modeling methods existed: *white* box, *grey* box, and *black* box methods. The first category required the user to provide the equations necessary to set up the model. This structure was very powerful when model was theoretical but it did not give good result when the environment gave random effects. At the other extreme, the black box method could be used for any data type and without prior knowledge of the system dynamics. The weakness of this method is that the reproducibility of its results is doubtful. The grey box method is a mixture of both methods. Information about the whole system might be known, while relationships between subsystems are not known.

1.6 Model constraints

1.6.1 Tolerance margins

In order to evaluate the model goodness, FAA has defined rules that describe the requirements to say that a model is "adequate". The estimated signals must be found within the tolerance margins during at least three seconds. The following table regroups the parameters tolerance margins for various flight missions accordingly with the FAA rules.

Table 1.1

Tolerance margins according to the parameters and the flight mission

Parameters	Variables	Flight mission	Tolerance margins	Flight mission	Tolerance margins
States	u	Level Flight Descending Autorotation	± 5ft/s	Ascending	± 5ft/s
	v		± 4ft/s		± 4ft/s
	w		± 3ft/s		± 1.66ft/s
	p		± 3deg/s		± 3deg/s
	q		± 3deg/s		± 3deg/s
	r		± 3deg/s		± 3deg/s
	φ		± 1.5deg		± 1.5deg
	θ		± 1.5deg		± 3deg
Outputs	A_x		± 3ft/s^2		± 3ft/s^2
	A_y		± 3ft/s^2		± 3ft/s^2
	A_z		± 3ft/s^2		± 3ft/s^2

1.6.2 Model performance

In order to quantify the model performance, two coefficients are calculated: the *correlation coefficient* which defines the trend of a signal, and the *fit coefficient* which measures the error between the measured and the estimated signals.

1.6.2.1 The correlation coefficient

The first method uses the correlation coefficient for the model validation. The correlation coefficient *Corr* is given by the following equation:

$$Corr = \frac{Cov(y,\hat{y})}{\sqrt{Var(y)Var(\hat{y})}} \tag{1.2}$$

where *Cov* is the covariance, *Var* is the variance, y is the measured output and $\hat{y}$ is the estimated output.

The correlation coefficient *Corr* equal to one denotes *perfect linear dependency* (no scatter) between the measured and the calculated or estimated outputs. A correlation coefficient equal to minus one (-1) denotes *inverse linear dependency* between the measured and the estimated outputs. A correlation coefficient of zero denotes the *linear independency* between the measured and the estimated outputs. The correlation coefficient computes the goodness of the model in a statistical sense, but provides little information about the model error. More information about the model error can be obtained by the *fit* coefficient calculation.

1.6.2.2 The fit coefficient

The Theil's inequality coefficient is used to define the fit of the estimated signals comparing to the measured ones, and is defined as follows:

$$FIT = 100(1-U) \tag{1.3}$$

Where:

$$U = \frac{\sqrt{\frac{1}{s-1}\sum_{i=1}^{s-1}(\hat{y}_i - y_i)^2}}{\sqrt{\frac{1}{s-1}\sum_{i=1}^{s-1}(\hat{y}_i)^2} + \sqrt{\frac{1}{s-1}\sum_{i=1}^{s-1}(y_i)^2}} \tag{1.4}$$

In Eq. (1.4), the variable $\hat{y}$ represents the estimated signal i.e. the model output, y is the measured signal, i.e. the real signal and s is the number of sample. The fit is expressed in percentage, and the U coefficient represents the ratio of the root-mean-square fit error and the root-mean-square values of the estimated and measured signals summed together. Its value is always found to be between zero and one where zero indicates a perfect fit and one the worst fit.

Although the acceptable value for U depends in the application, in general, a value in the range 0.25-0.3 indicates a good agreement (Jategaonkar, 2006).

1.6.3 Model plausibility

An over-parameterized model will give a good response match, but not necessarily a good system representation. The most direct way to check the plausibility of the estimated parameters is by their comparisons with estimates from other sources. Validation on complementary data not used for the estimation is sometimes also termed loosely the "*acid test*", and are used to check the model capability. In most of the model validation exercises, including those for the flight simulators, this approach of separating the data for model development and demonstration of model fidelity is adopted. Demonstration of the model fidelity on complementary data provides increased confidence in the model predictive capability.

For all flight conditions, the identified model is validated for *three* different flight tests, not used to identify the model. Different types of missions are used (when possible) in order to validate the longitudinal and lateral motion of the helicopter. If the estimated signals remain within the FAA tolerance margins, then the confidence in the model capability is increased.

1.7 Questions raised in this thesis

By analysis of the overview of system identification presented in this chapter, two problems related to the *Quad-M* are highlighted:

(1) Can the dynamical behaviour of the Bell-427 helicopter be defined by a six DOF linear model? This problem is based on the hypothesis that the helicopter is considered as a rigid body.

This problem is related to the *model structure* usually used to describe the rigid body dynamics.

(2) Three methods (classical equations of motion, linear block and nonlinear block) are used to estimate the linear accelerations from the state variables. Which method is the most appropriate to give the best results?

This problem is related to the *method* set up to identify the model.

In order to answer these questions, a six DOF state space model is built by a *time*-domain method. A comparison between results obtained with the three methods to observe the system outputs from its states is then performed. In case when the estimated signals satisfy the FAA rules, is concluded that model is well estimated.

CHAPTER 2

STATE EQUATION DETERMINATION

2.1 Formulation

Generally, a state space system with discrete dynamics is defined with the following equation:

$$x(k+1) = \mathrm{A}(k)x(k) + \mathrm{B}(k)u(k) \tag{2.1}$$

where $x \in \Re^n$ and $u \in \Re^m$ are the state vector (x) and the input vector (u) which is the pilot command vector. Matrices A and B are respectively the "state matrix", the "input matrix".

The *state* variables x are the subset of system variables that can represent the entire state of the system at any given time. The states variables were chosen as the linear and angular velocities u, v, w, p, q and r and the Euler angles of the helicopter around the X-axis and Y-axis (Hamel *et al.*, 1996), which are denoted by φ and θ. These variables are sufficient to identify a rigid body dynamics of a parametric fully-coupled 6 DOF models. The heading angle ψ is dropped because it does not influence the helicopter dynamic response.

The *input* variables u are the pilot controls, which are the collective, the longitudinal cyclic, the lateral cyclic, and the pedals commands (see Section 1.2.1).

Two methods are used to determine the A and B matrices. First, a *recursive* method is set up to define the matrices in open-loop i.e. the estimated states variables are obtained from the measured states variables and the pilot commands. Then, an *optimization* procedure is considered in order to obtain the optimal matrices in closed-loop i.e. the estimated states variables are obtained only from the measured pilot commands.

2.2 Identification of the open-loop state equation by use of recursive method

2.2.1 Theory

For a discrete-time study, the states at a sample time k depend on the states and the input controls at the previous sample time k - 1 so that a recursive method can be used to estimate the A and B matrices (Jategaonkar, 2006). As mentioned in the Section 2.1, the state vector x and the controls vector u are defined as follows:

$$x = \begin{bmatrix} u & v & w & p & q & r & \phi & \theta \end{bmatrix}^T \tag{2.2}$$

$$u = \begin{bmatrix} coll & long & lat & ped \end{bmatrix}^T \tag{2.3}$$

Then, s equations are considered with $(n + m)$ parameters to be estimated. We denote by n the number of states variables and m the number of pilot commands. The k^{th} equation defining the state element x_i at step time $k + 1$ is:

$\forall k \in [\![1;s]\!], \forall i \in [\![1;n]\!],$

$$x_i(k+1) = \sum_{j=1}^{n} a_{ij} x_j(k) + \sum_{j=1}^{m} b_{ij} u_j(k) \tag{2.4}$$

The input vector is defined as follows:

$\forall k \in [\![1;s]\!],$

$$io(k) = \begin{bmatrix} x(k) & u(k) \end{bmatrix}^T \tag{2.5}$$

where io is a vector of dimensions $[(n + m) \times 1)]$.

The parameter vector gathers the elements of A and B matrices and is formed by the parameters to be estimated. The elements of the i^{th} line of A and B matrices at the k^{th} step time are denoted as follows:

$\forall k \in [\![1;s]\!], \forall i \in [\![1;n]\!],$

$$ab_i(k) = [a_{i1} \quad a_{i2} \quad a_{i3} \quad a_{i4} \quad a_{i5} \quad a_{i6} \quad a_{i7} \quad a_{i8} \quad b_{i1} \quad b_{i2} \quad b_{i3} \quad b_{i4}] \tag{2.6}$$

The i^{th} state at the k^{th} sample time can be written, based on Eqs. (2.4) and (2.5):

$\forall k \in [\![1;s]\!], \forall i \in [\![1;n]\!],$

$$\begin{aligned} x_{i\,des}(k+1) &= x_{i\,est}(k+1) + e_i(k) \\ x_{i\,des}(k+1) &= ab_i(k)\,io(k) + e_i(k) \end{aligned} \tag{2.7}$$

where e is the error to be minimized.

For s measurements:

$$\begin{cases} x_i(2) = ab_i(1)\,io(1) + e_i(1) \\ \ldots \\ x_i(s) = ab_i(s-1)\,io(s-1) + e_i(s-1) \end{cases} \tag{2.8}$$

In Eq. (2.8), the initial conditions are represented by the first sample time of *io* vector.

These s equations are summarized with the following formulation:

$$X_{des} = AB.IO + e \tag{2.9}$$

Where:

$$\begin{cases} X_{des}(s) = [x_{des}(2) \quad \ldots \quad x_{des}(s)]^T \\ IO(s-1) = [io(1) \quad \ldots \quad io(s-1)]^T \\ e(s-1) = [e(1) \quad \ldots \quad e(s-1)]^T \end{cases} \tag{2.10}$$

In order to minimize the error, the following cost function J is defined:

$$J(AB) = e^T We \tag{2.11}$$

where W is a weighted matrix, which is diagonal and where the non diagonal element are set to zero. One of the properties of this type of matrix is that its transpose is equal to itself i.e. $W^T = W$. If the diagonal terms $w(k)$ are equal to one, then the error coefficients have the same weight. If not, then it means that the error coefficients are different. Derusso *et al.* (1998) have chosen this weighted matrix so that the k^{th} element is defined as follows:

$$w(k) = \gamma^{s-k} \tag{2.12}$$

where $\gamma < 1$ is a forgetting factor and s is the number of measurements.

Equation (2.11) becomes:

$$J(AB) = \sum_{k=1}^{s-1} \gamma^{s-k} e^2(k) \tag{2.13}$$

As s-k gets larger and γ does not equal to one, the weighting factor approaches zero, therefore older points receive little weight. As s-k goes to zero, the weighting factor approaches one and the most recent data are favoured. The smaller γ is, the faster the algorithm can track, but the more the estimates vary, even the true parameters are time-invariant. By developing the cost function equation, we obtain:

$$\begin{aligned} J(AB) = e^T We &= (X_{des} - AB.IO)^T W (X_{des} - AB.IO) \\ &= X_{des}^T W X_{des} - X_{des}^T WAB.IO - (AB.IO)^T W X_{des} + (AB.IO)^T WAB.IO \\ &= X_{des}^T W X_{des} - 2X_{des}^T WAB.IO + IO^T.AB^T W\ AB.IO \end{aligned} \tag{2.14}$$

The A and B matrices which minimize the cost function J are found by equalizing the cost function derivative with respect to AB to zero:

$$\begin{aligned} & \frac{\partial J(AB)}{\partial AB} = 0 \\ \Leftrightarrow\ & -2X'_{des} W\, IO + (IO.AB)^T W.IO + (IO.AB)^T W.IO = 0 \\ \Leftrightarrow\ & -2X'_{des} W.IO + 2IO^T W.IO.AB = 0 \end{aligned} \tag{2.15}$$

and next equation is obtained:

$$X_{des}^{T} W .IO = IO^{T} W .IO.AB \tag{2.16}$$

The parameter vector is estimated as follows:

$$A\hat{B} = \left(IO^{T} W .IO \right)^{-1} IO^{T} W .X_{des} \tag{2.17}$$

The matrix $IO(N+1)$ can be written as:

$$IO(N+1) = \left[io(1) \quad io(2) \quad \ldots \quad io(N+1) \right]^{T} \tag{2.18}$$

The term $IO^{T} W.IO$ of the expression of $A\hat{B}$ (see Eq. (2.17)) is developed by introducing a recurrence relationship:

$$\begin{aligned} IO(N+1)^{T} W(N+1) IO(N+1) &= \sum_{k=1}^{N+1} io(k) w(k) io^{T}(k) \\ &= \sum_{k=1}^{N+1} io(k) \gamma^{N+1-k} io^{T}(k) \\ &= \sum_{k=n}^{N} io(k) \gamma\gamma^{N-k} io^{T}(k) + io(N+1) \gamma^{(N+1)-(N+1)} io^{T}(N+1) \\ &= \gamma IO^{T}(N) W(N) IO(N) + io(N+1) io^{T}(N+1) \end{aligned} \tag{2.19}$$

The matrix P is defined as follows:

$$P^{-1}(k) = IO^{T}(k) W(k) IO(k) \tag{2.20}$$

By substituting the expression of P^{-1} in Eq. (2.19) for $k = N$, the expression of P^{-1} becomes:

$$P^{-1}(N+1) = \gamma P^{-1}(N) + io(N+1) io^{T}(N+1) \tag{2.21}$$

so that:

$$P(N+1) = \left[\gamma P^{-1}(N) + io(N+1) io^{T}(N+1) \right]^{-1} \tag{2.22}$$

In order to obtain a recurrence expression of P, the following analytical formula is used:

$$\left(A+BCD\right)^{-1}=A^{-1}-A^{-1}B\left(C^{-1}+DA^{-1}B\right)^{-1}DA^{-1} \tag{2.23}$$

By denoting $A=\gamma P^{-1}(N), B=io(N+1), C=1, D=io^{T}(N+1)$, the expression of P $(N+1)$ is rewritten as follows:

$$\begin{aligned} P(N+1)=&\frac{P(N)}{\gamma} \\ &-\frac{P(N)}{\gamma}io(N+1)\left[1+io^{T}(N+1)\frac{P(N)}{\gamma}io(N+1)\right]^{-1}io^{T}(N+1)\frac{P(N)}{\gamma} \end{aligned} \tag{2.24}$$

With the same reasoning, the term $IO^{T}W.X_{des}$ given by Eq. (2.17) is defined as follows:

$$IO^{T}(N+1)W(N+1)X_{des}(N+1)=\gamma IO^{T}(N)W(N)X_{des}(N)+io(N+1)x_{des}(N+1) \tag{2.25}$$

A new expression of $A\hat{B}$ is reformulated by use of Eqs. (2.17) and (2.25):

$$\begin{aligned} A\hat{B}(N+1)=&\left[\frac{P(N)}{\gamma}-\frac{P(N)}{\gamma}io(N+1)\left[1+io^{T}(N+1)\frac{P(N)}{\gamma}io(N+1)\right]^{-1}io^{T}(N+1)\frac{P(N)}{\gamma}\right] \\ &.\left[\gamma IO^{T}(N)W(N)X_{des}(N)+io(N+1)x_{des}(N+1)\right] \end{aligned} \tag{2.26}$$

Hence, by substituting Eq. (2.24) into Eq. (2.26), the estimated parameter matrix is defined as:

$$A\hat{B}(N)=P(N)IO^{T}(N)W(N)X_{des}(N) \tag{2.27}$$

By developing Eq. (2.26) and by replacing the $A\hat{B}(N)$ formulation given by Eq. (2.27), Eq. (2.26) becomes:

$$\begin{aligned} A\hat{B}(N+1) &= A\hat{B}(N) + \frac{P(N)}{\gamma} io(N+1) x_{des}(N+1) \\ &- \frac{P(N)}{\gamma} io(N+1)\left[1 + io^T(N+1)\frac{P(N)}{\gamma} io(N+1)\right]^{-1} io^T(N+1) A\hat{B}(N) \\ &- \frac{P(N)}{\gamma} io(N+1)\left[1 + io^T(N+1)\frac{P(N)}{\gamma} io(N+1)\right]^{-1} io^T(N+1)\frac{P(N)}{\gamma} io(N+1) x_{des}(N+1) \end{aligned} \tag{2.28}$$

The K matrix is defined as:

$$K(N+1) = \frac{P(N)}{\gamma} io(N+1)\left[1 + io^T(N+1)\frac{P(N)}{\gamma} io(N+1)\right]^{-1} \tag{2.29}$$

Then, Eq. (2.28) becomes:

$$\begin{aligned} A\hat{B}(N+1) &= A\hat{B}(N) + \frac{P(N)}{\gamma} io(N+1) x_{des}(N+1) - K(N+1) io^T(N+1) A\hat{B}(N) \\ &- K(N+1) io^T(N+1)\frac{P(N)}{\gamma} io(N+1) x_{des}(N+1) \end{aligned} \tag{2.30}$$

By isolating $\frac{P(N)}{\gamma} io(N+1)$ of Eq. (2.29), and replacing its expression in Eq. (2.30), the concise form of $A\hat{B}(N+1)$ is obtained:

$$A\hat{B}(N+1) = A\hat{B}(N) + K(N+1)\left[x_{des}(N+1) - io^T A\hat{B}(N)\right] \tag{2.31}$$

By substituting Eq. (2.29) into Eq. (2.24), the new expression of P is written as:

$$P(N+1) = \frac{1}{\gamma}\left[I_d - K(N+1) io^T(N+1)\right] P(N) \tag{2.32}$$

The algorithm of method implementation in MATLAB / SIMULINK is the following:

(1) Initialization of the forgetting factor γ so that $0 < \gamma \leq 1$, which corresponds to an exponential weighting. We observe that the computation time decreases by keeping γ constant and equal to one and does not significantly affect the results.

(2) Initialization of the matrices P and $A\hat{B}$ where P is chosen as a diagonal matrix so that its diagonal terms have high values and the $A\hat{B}$ elements are set to zero.

(3) Calculation of the K matrix:

$$K(k+1) = P(k)io(k+1)\left[\gamma + io^T(k+1)P(k)io(k+1)\right]^{-1}$$

(4) Calculation of the $A\hat{B}$ matrix:

$$A\hat{B}(k+1) = A\hat{B}(k) + K(k+1)\left[x_{des}(k+1) - io^T(k+1)A\hat{B}(k)\right]$$

(5) Calculation of the P matrix:

$$P(k+1) = \frac{1}{\gamma}\left[I_d - K(k+1)io^T(k+1)\right]P(k)$$

(6) Computation of the third step and repetition of the iteration procedure for each sample time.

2.2.2 Practical application

For the open-loop system, no state feedback is used. The state variables are functions of the real state variables and pilot controls at the previous step time. The regression method described in the previous section is used to estimate the state variables. For each time step, the A and B matrices parameters are obtained. Each couple of matrices is tested for the entire signals, and the couples giving the best results are selected. An index was defined to characterize the matrix performance, and is used in the fuzzy logic method to find the potential value of each point:

$\forall k \in \mathbb{R}^s, \forall i \in \mathbb{R}^n,$

$$pot(k) = \sum_{j=1}^{s} e^{-4\sum_{i=1}^{n}\left[X_{des}(i) - A\hat{B}_i(k)IO(k)\right]^2} \tag{2.33}$$

The highest this index is, the better is the performance of the selected matrix. Since the matrix value was determined, the state space dynamics is time-invariant:

$$x(k+1) = Ax(k) + Bu(k) \tag{2.34}$$

and can be written under the following form:

$$x(k+1) = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \tag{2.35}$$

as shown in the following figure:

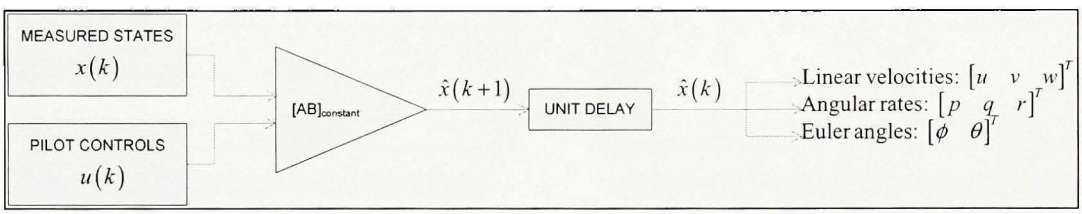

Figure 2.1 Simulation of the open-loop state equation.

2.2.3 Results for the flight condition HA6000ft-50kts

In this section, the plots of the states variables for one case study defined as HA6000ft-50kts are presented. The signals used to perform the model identification are set up with four 2-3-1-1 concatenated command pilot. The three validations used to validate the model are: (1) a 2-3-1-1 longitudinal command in level flight, (2) no command in a -1,000 ft/min flight, and (3) a 2-3-1-1 lateral command in level flight.

Figure 2.2 to Figure 2.9 show the pilot commands used to identify and validate the model for this flight condition (left column) and the state variables evolution (right column). For the figures on the left side, the full blue line are the pilot commands used for the identification and the dashed green lines are the pilot commands used to validate the model. For the figures on the right side, the full red lines are the estimated states signals and the dashed black lines are the tolerance margins.

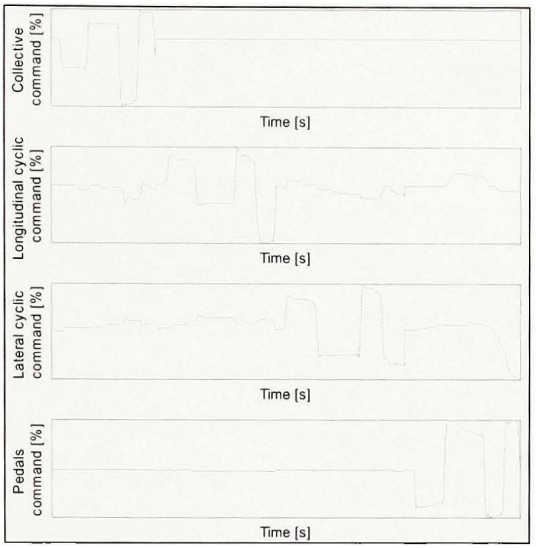

Figure 2.2 Pilot inputs for the model identification.

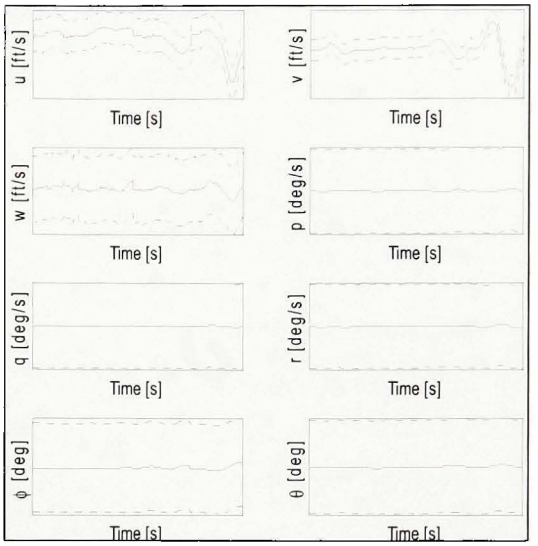

Figure 2.3 State variables evolution in open-loop for the identification.

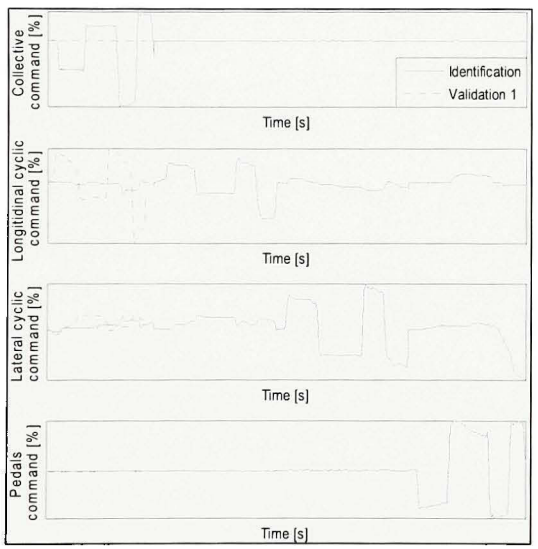

Figure 2.4 Pilot command for the validation 1.

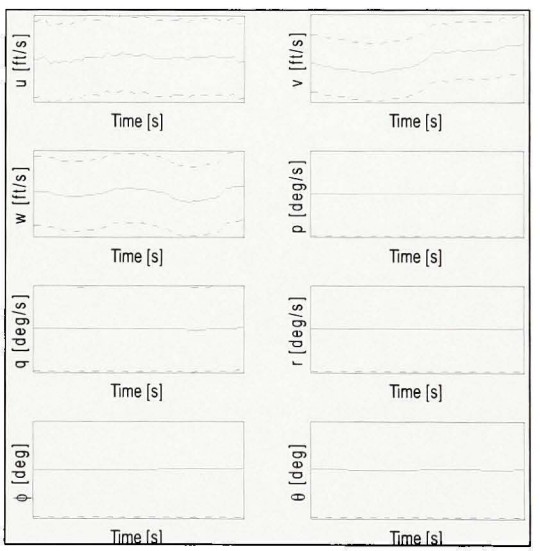

Figure 2.5 State evolution in open-loop for the validation 1.

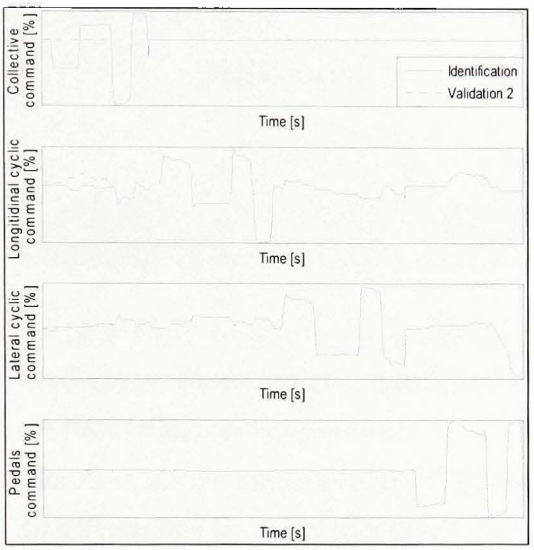

Figure 2.6 Pilot command for the validation 2.

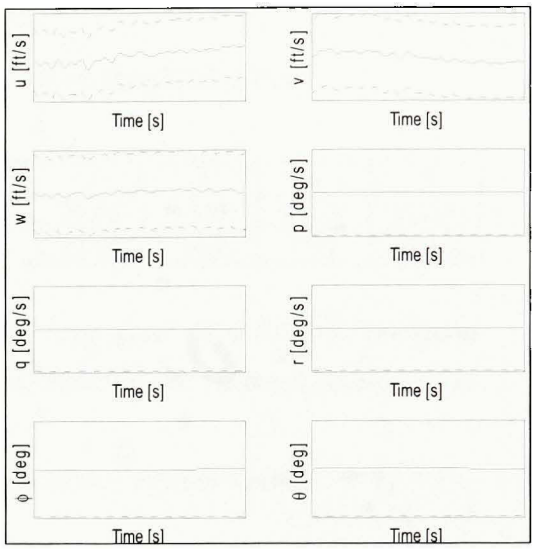

Figure 2.7 State evolution in open-loop for the validation 2.

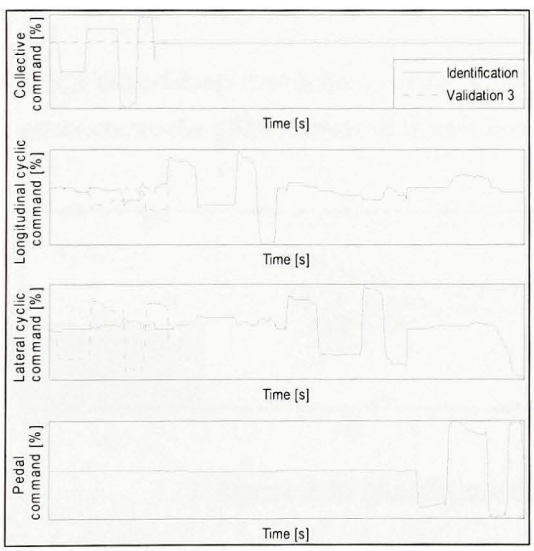

Figure 2.8 Pilot command for the validation 3.

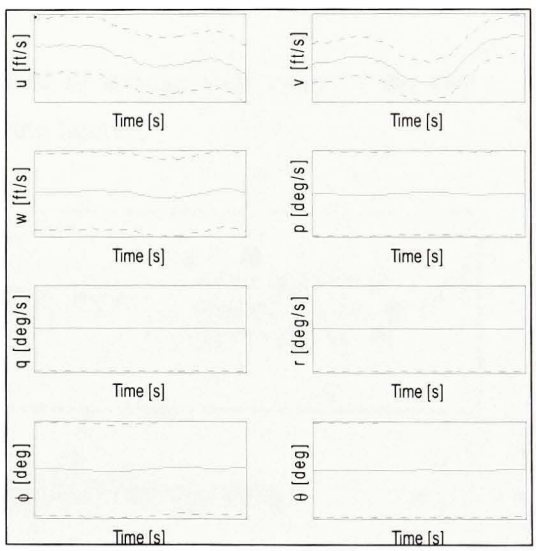

Figure 2.9 State evolution in open-loop for the validation 3.

Graphically, the estimated signals in open-loop study respect the FAA tolerance margins during all time histories. The fit and correlation coefficients for this flight condition are shown in the next table.

Table 2.1

Model performances for the flight condition HA6000ft-50kts

	u	v	w	p	q	r	φ	θ
Fit [%]	99.21	99.51	97.17	99.42	99.19	99.55	99.49	99.38
Correlation [%]	99.99	100	99.84	99.99	99.99	100	100	99.99

For this flight condition, the numerical results are very good. The fit and correlation coefficients values are higher than 97%. This model is identified and validated accordingly to the FAA tolerances margins rules.

Moreover, all the estimated signals stay within the tolerance margins defined by the FAA, which increases the model goodness.

2.3 Identification of the closed-loop state equation by use of an optimization procedure

For a closed-loop simulation, a state feedback is used so that the only inputs of the state equation are the pilot controls as shown in the following figure:

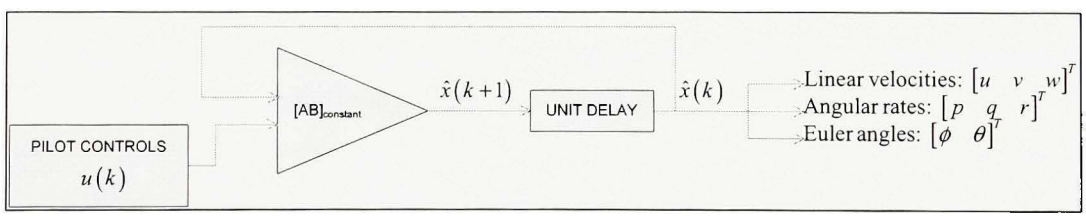

Figure 2.10 Simulation of the closed-loop state equation.

The A and B matrices found in open-loop do not guarantee good results in closed-loop. Indeed, a system is stable if and even if the eigenvalues are negative if the system is continuous or if and even if the eigenvalues are inside an unitary circle centered on the plan origin if the system is discrete. When the eigenvalues do not satisfy these constraints, then the model is not stable, the responses tend to diverge. An optimization procedure is necessary

to obtain good matrices in closed-loop. The open-loop system matrices are used as initial guesses for the optimization.

The procedure is based on the neural networks theory. A model can be identified by providing a set of examples, i.e. input/target pairs of proper system behaviour (Hagan *et al.*, 1996). A neural network is composed of elements operating in parallel. The values of the connections between them are adjusted based on a comparison between the network output and the target (desired) output. This method is called *backpropagation* and the training is based on the Levenberg-Marquadt algorithm.

A second method used to satisfy the project constraints (see Section 1.6) is the model tuning by adjusting the initial conditions of the system. This technique is also called a "proof-of-match" of the model.

Thus, the neural network optimization enables the system to be stable and the "proof-of-match" enables the estimated state parameters to match with the measured ones.

2.3.1 Neural network theory

A neural network is composed of an input p, a weight W, a bias b and a transfer function f called *activation function* as shown below:

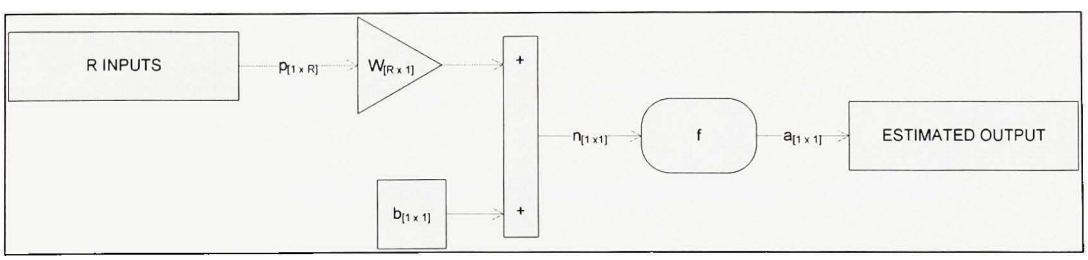

Figure 2.11 Neural network architecture.

The mathematical relationship between all these parameters can be formulated as in Eq. (2.36):

$$a = f(Wp + b) \tag{2.36}$$

In order to obtain an output a with Q elements i.e. a dimension $[Q \times 1]$, Q neurons should be arranged in parallel. Therefore, the network can be composed by several layers in series, with different activation functions. The only constraint for these functions is that they are to be *differentiable*. For a multilayer network, Eq. (2.36) must be adapted in order to consider all activation functions. For example, the equation relating the input and output of a two-layer network ($m = 2$) with only one neuron ($Q = 1$) is:

$$a_2 = f_2\left(W^2 p^2 + b^2\right) = f_2\left(W^2 \underbrace{\left[f_1\left(W^1 p^1 + b^1\right)\right]}_{p^2} + b^2\right) \tag{2.37}$$

A neural network with several outputs and several layers is presented in Figure 2.12 :

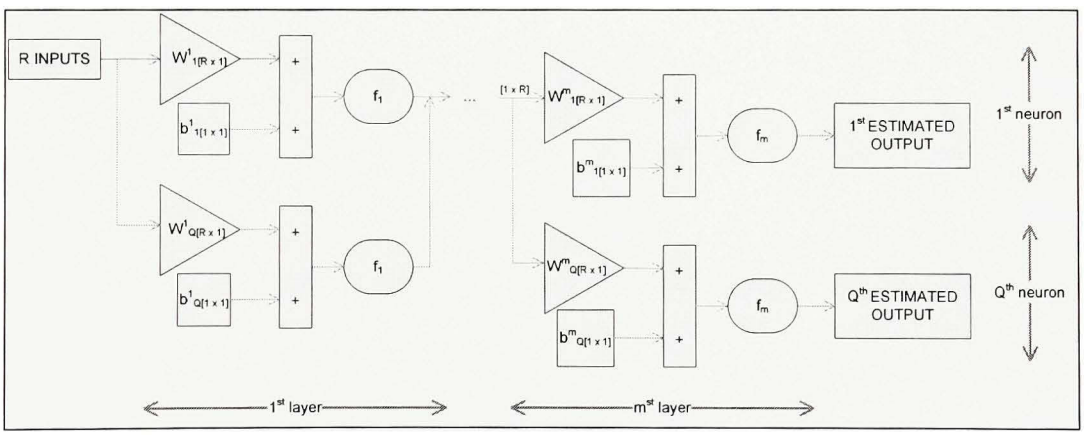

Figure 2.12 Multi-output and multilayer neural network architecture.

The backpropagation algorithm (Hagan *et al.*, 1996) is a gradient descent optimization procedure in which a mean square error performance index is minimized by adjusting the network parameters (weights and biases). The performance index at iteration k is defined as follows:

$$F(w,b)=\frac{1}{2}e^{T}(k)e(k)=\frac{1}{2}\left[y_{des}(k)-a(k)\right]^{T}\left[y_{des}(k)-a(k)\right] \tag{2.38}$$

where y_{des} is the target output and a is the neural network output which is compared to the desired output. The steepest descent algorithm considered for the approximate mean square error modeling is:

$$w_{i,j}^{m}(k+1)=w_{i,j}^{m}(k)-\eta\frac{\partial F}{\partial w_{i,j}^{m}} \tag{2.39a}$$

$$b_{i}^{m}(k+1)=b_{i}^{m}(k)-\eta\frac{\partial F}{\partial b_{i}^{m}} \tag{2.39b}$$

where m is the number of the considered layer and η is the learning rate called the network training speed. For a multilayer network, the error is an indirect function of the weights in the hidden layers, so that these derivatives are calculated differently:

$$\frac{\partial F}{\partial w_{i,j}^{m}}=\frac{\partial F}{\partial n_{i}^{m}}\cdot\frac{\partial n_{i}^{m}}{\partial w_{i,j}^{m}} \tag{2.40a}$$

$$\frac{\partial F}{\partial b_{i}^{m}}=\frac{\partial F}{\partial n_{i}^{m}}\cdot\frac{\partial n_{i}^{m}}{\partial b_{i}^{m}} \tag{2.40b}$$

It is known that:

$$n_{i}^{m}=\sum_{j=1}^{S^{m-1}}w_{i,j}^{m}\,a_{j}^{m-1}+b_{i}^{m} \tag{2.41}$$

so that the two derivatives $\frac{\partial n_{i}^{m}}{\partial w_{i,j}^{m}}$ and $\frac{\partial n_{i}^{m}}{\partial b_{i}^{m}}$ are directly defined since the net input of the m^{th} layer is an explicit function of the weights and biases in that layer:

$$\frac{\partial n_{i}^{m}}{\partial w_{i,j}^{m}}=a_{j}^{m-1} \tag{2.42a}$$

$$\frac{\partial n_i^m}{\partial b_i^m} = 1 \tag{2.42b}$$

The F sensitivity is defined as any effect of the system function or any other system characteristic caused by a change in one or more system parameters. The larger the system sensitivity is, the stronger is the effect of small changes in the system performances. We define:

$$s_i^m = \frac{\partial F}{\partial n_i^m} \tag{2.43}$$

Then, Eqs. (2.40a) and (2.40b) become:

$$\frac{\partial F}{\partial w_{i,j}^m} = s_i^m a_j^{m-1} \tag{2.43a}$$

$$\frac{\partial F}{\partial b_i^m} = s_i^m \tag{2.43b}$$

Thus, the weight and bias expressions defined in Eqs. (2.39a) and (2.39b) are rewritten as follows:

$$w_{i,j}^m (k+1) = w_{i,j}^m (k) - \eta\, s_i^m\, a_j^{m-1} \tag{2.45a}$$

$$b_i^m (k+1) = b_i^m (k) - \eta\, s_i^m \tag{2.45b}$$

The Jacobian matrix is next defined:

$$\frac{\partial n_i^{m+1}}{\partial n_m} = \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_1^{m+1}}{\partial n_{S^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_2^{m+1}}{\partial n_{S^m}^m} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_{S^m}^m} \end{bmatrix} \tag{2.46}$$

For one of the elements of this above matrix, we can write:

$$\frac{\partial n_i^{m+1}}{\partial n_j^m} = \frac{\partial \left(\sum_{l=1}^{R} w_{il}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial n_j^m} \tag{2.47a}$$

$$\frac{\partial n_i^{m+1}}{\partial n_j^m} = w_{il}^{m+1} \frac{\partial a_l^m}{\partial n_j^m} \tag{2.47b}$$

By according to Figure 2.11, $a = f(n)$. Eq. (2.47 b) can be written in the following form:

$$\frac{\partial n_i^{m+1}}{\partial n_j^m} = w_{il}^{m+1} \frac{\partial f^m \left(n_j^m \right)}{\partial n_j^m} \tag{2.47c}$$

$$\frac{\partial n_i^{m+1}}{\partial n_j^m} = w_{il}^{m+1} \dot{f}^m \left(n_j^m \right) \tag{2.47d}$$

By generalization of these terms:

$$\frac{\partial n^{m+1}}{\partial n^m} = W^{m+1} \dot{F}^m \left(n^m \right) \tag{2.48}$$

where:

$$\dot{F}^m = \begin{bmatrix} \dot{f}^m \left(n_1^m \right) & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \ldots \\ \ldots & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \dot{f}^m \left(n_Q^m \right) \end{bmatrix} \tag{2.49}$$

The recurrence relationship between sensitivities at different layers is written by using the chain rule:

$$s^m = \frac{\partial F}{\partial n^m} = \frac{\partial F}{\partial n^{m+1}} \frac{\partial n^{m+1}}{\partial n^m}$$
$$s^m = s^{m+1} W^{m+1} \dot{F}^m \left(n^m \right) \tag{2.50}$$

where $m = M - 1, \ldots, 2, 1$ and M is the number of layers.

The starting point s^M (where M is the output layer) is defined with the recurrence relationship between sensitivities:

$$s_i^M = \frac{\partial F}{\partial n_i^M} = \frac{\partial \left[\frac{1}{2} \left(t_i - a_i \right)^T \left(t_i - a_i \right) \right]}{\partial a_i} \cdot \frac{\partial a_i}{\partial n_i^M}$$
$$s_i^M = -\left(t_i - a_i \right) \frac{\partial a_i}{\partial n_i^M} \tag{2.51}$$

Since:

$$\frac{\partial a_i}{\partial n_i^M} = \frac{\partial a_i^M}{\partial n_i^M} = \frac{\partial f^M \left(n_j^M \right)}{\partial n_i^M} = \dot{f}^M \left(n_j^M \right) \tag{2.52}$$

Next equation is written:

$$s_i^M = -\left(t_i - a_i \right) \dot{f}^M \left(n_j^M \right) \tag{2.53}$$

By summarizing, during the first step, the input is propagated forward through the network. Then, during the second step, the sensitivities are propagated back through the network. Finally, during the third step, the biases and weights are updated.

The weights and biases update can be powered by adding a momentum coefficient μ. It allows the change of the weight and bias at iteration k has an influence on the change of the weight and bias at iteration k+1:

$$\Delta w_{i,j}^{m}(k+1) = -\eta s_i^m a_j^{m-1} - \mu \Delta w_{i,j}^{m}(k) \tag{2.54a}$$

$$\Delta b_i^m(k+1) = -\eta s_i^m - \mu \Delta b_i^m(k) \tag{2.54b}$$

2.3.2 Algorithm application in the project

The algorithm used in the neural network theory to identify system behaviour is here applied. Indeed, the system has eight states variables and four control inputs, therefore a total of 12 inputs (R = 12). Due to the fact that A matrix has dimensions [8 × 8], the network must have eight (8) neurons arranged in parallel (Q = 8). This configuration can be synthesized in the simplest architecture (see Figure 2.11) such that the weight is a matrix with dimension [8 × 8] and the biases a vector with dimension [8 × 1].

In order to get the same equation with neural network and state space formulations (comparison between Eqs. (2.35) and (2.36)), the neural network architecture must be defined as follows:

(1) The biases must be null;

(2) The activation function must be linear in order to ensure the relation $f(a) = a$ and;

(3) The network must have only one layer (simplest architecture).

The adjustment of the weights values i.e. the A and B matrices elements can be done with an optimization on-line algorithm in which the weight values are updated. By focusing on this particular case of the neural network theory, the A and B matrices must be updated as follows:

$$A_{new} = A_{old} - dA_{new} - \mu dA_{old} \tag{2.55a}$$

$$B_{new} = B_{old} - dB_{new} - \mu dB_{old} \tag{2.55b}$$

Where:

$$dA_{new} = -\eta\left(x_{des}(k+1) - \hat{x}(k+1)\right)\hat{x}(k) \tag{2.56a}$$

$$dB_{new} = -\eta\left(x_{des}(k+1) - \hat{x}(k+1)\right)u(k) \tag{2.56b}$$

In order to increase the on-line optimization power, the pilot controls and desired states are concatenated. After several tests, it has been observed that five cycles are sufficient to obtain good results. Then, at each sequence, the algorithm parameters (learning rate and momentum) are divided by two in order to ensure the convergence of the optimization algorithm.

2.3.3 Initial conditions

As previously said, a recurrence relationship exists between the state variables of a system. For $k = 0$:

$$x(1) = Ax(0) + Bu(0) \tag{2.57}$$

Then, for $k = 1$:

$$\begin{aligned} x(2) &= Ax(1) + Bu(1) \\ x(2) &= A\left[Ax(0) + Bu(0)\right] + Bu(1) = A^2x(0) + ABu(0) + Bu(1) \end{aligned} \tag{2.58}$$

For $k = 2$:

$$\begin{aligned} x(3) &= Ax(2) + Bu(2) \\ x(3) &= A\left[Ax(1) + Bu(1)\right] + Bu(2) \\ x(3) &= A\left[A\left[Ax(0) + Bu(0)\right] + Bu(1)\right] + Bu(2) \\ &= A^3x(0) + A^2Bu(0) + ABu(1) + Bu(2) \end{aligned} \tag{2.59}$$

Thus, for $k = n$:

$$x(n+1) = A^{n+1}x(0) + A^{n}Bu(0) + \sum_{j=1}^{n} A^{n-j}Bu(j) \tag{2.60}$$

By analysing this equation, is noticed that there are two ways to influence the state evolution:

(1) The first way is the tuning of the initial state values $x(0)$ and;

(2) The second way is to modify slightly the initial command inputs $u(0)$.

For system identification, is more acceptable to tune the initial state variables than the command inputs. For validation, both tuning parameters can be used.

Two methods are used to set up the initial conditions:

(1) A manually tuning and;

(2) An automatic tuning of the initial state variables based on an optimization.

Both methods aim to the minimization of the points outside the tolerance margins.

2.3.3.1 Manual tuning

Manual tuning is done when initial conditions are null. Empirical results are found for a majority of flight conditions: a slightly change of the Euler angles initial conditions has a great influence on the linear velocities, which can be explained by the equations of motion in which the u, v, w evolution is influenced by θ and φ (Prouty, 2002).

Table 2.2 shows the reduction of the points outside the tolerance margins when a manual tuning is done. The initial guesses are the initial conditions set to zero.

Table 2.2

Influence of the initial state variables on the number of point outside the tolerance margins when a manual tuning is done (%)

Flight condition	HA3000ft	HA6000ft	HF3000ft	HF6000ft
Reduction of the points out of the tolerance margins when initial conditions are optimized (%)	20.51	20.13	60.96	62.20

Table 2.2 shows that the initial conditions of the state equation have a high influence in the percentage of points found outside the tolerance margins. For the flight conditions HF, the reduction percentage is higher than 60%.

2.3.3.2 Automatic tuning

The second step is to set up the initial conditions automatically with an optimization algorithm which finds a minimum of a constrained multivariable function. The output of the cost function to be minimized is the number of points outside the tolerance margins defined for the studied flight test. A lower and upper bounds on the design variables of the initial states conditions $x(0)$ are fixed, so that the solution should be always in the defined range. This range is the tolerances margins.

A necessary condition for a point x_0 (initial states conditions) to be a global minimum is for it to satisfy the Karush-Kunh-Tucker (KKT) conditions. Suppose the cost function to be minimized is J: $\mathbb{R}^n \rightarrow \mathbb{R}$ (n = 8) and the constraint function is G: $\mathbb{R}^n \rightarrow \mathbb{R}$. The constraint function G is a linear function such that $G(x_0) = x_0 \pm tol$. We suppose that they are continuously differentiable at a point x_0. If x_0 is a minimum, then there exists a constant λ such that:

(1) $\nabla J(x_0) + \sum_{i=1}^{m} \lambda_i . \nabla G_i(x_0) = 0$,

(2) $G_i(x_0) \le 0$, for all $i = 1, ..., m$, where m is the number of inequalities. Because there are lower and upper bounds for each state variable, there are 8×2=16 inequalities.

(3) $\lambda_i \ge 0$ for all $i = 1, ..., m$,

(4) $\lambda_i G_i(x_0) = 0$, for all $i = 1, ..., m$.

The KKT equations describe the gradients cancellation between the objective function and the constraints (the solution range) at the solution point (see condition (1)). For the gradient to be cancelled, Lagrange multipliers are necessary to balance the deviations in magnitude of the objective function, which are constraints gradients (see condition (4)). If J is convex, then the KKT conditions are also sufficient.

The second order approximation is used to find the minimum of the cost function J. the Taylor series of $J(x)$ is:

$$J(x + \Delta x) = J(x) + \nabla J(x)^T \Delta x + \frac{1}{2} \Delta x^T H \Delta x \tag{2.61}$$

where ∇J is the gradient of the cost function and H is the Hessian matrix of the cost function.

The Taylor series of the gradient of the cost function is:

$$\nabla J(x + s_k) = \nabla J(x) + H s_k \tag{2.61}$$

The minimum of the cost function is found when the gradient of the cost function is null i.e. $\nabla J(x_k + s_k) = 0$.

Then the direction step s_k is calculated

$$\begin{aligned} & \nabla J(x_k) + H s_k = 0 \\ \Leftrightarrow \quad & s_k = -H^{-1} \nabla J(x_k) \end{aligned} \tag{2.62}$$

If H is a positive semi-definite matrix, then J is a convex function. In this case, the optimization problem has a global minimum if there exists at least one vector x_0 satisfying the constraints $G(x_0)$ and $J(x_0)$ which is bounded below the feasible region, which corresponds to the tolerance margins. If H is a positive definite matrix, then the global minimum is unique.

As a starting point, H_0 can be set to any symmetric positive definite matrix, for example to the identity matrix.

A line search is performed in order to find the optimal α_k, which satisfies the Wolfe conditions:

(1) $J(x_k + \alpha_k s_k) \leq J(x_k) + c_1 \alpha_k s_k^T \nabla J(x_k)$

(2) $s_k^T \left| \nabla J(x_k + \alpha_k s_k) \right| \geq c_2 s_k^T \left| \nabla J(x_k) \right|$ with $0 < c_1 < c_2 < 1$

The condition (1) ensures that α_k decreases J sufficiently, and condition (2) ensures that the slope of the function $\varsigma(\alpha_k) = J(x_k + \alpha_k s_k)$ at α_k is higher than c_2 times the function slope at $\alpha_k = 0$.

The update of x is then computed:

$$x_{k+1} = x_k + \alpha_k s_k \tag{2.63}$$

The evolution of the cost function gradient is then defined:

$$q_k = \nabla J(x_{k+1}) - \nabla J(x_k) \tag{2.64}$$

Because it is time consuming to define the Hessian matrix of the cost function at each iteration, a quasi-Newton updating method is used to compute an approximate matrix of H. This quasi-Newton method approximates the Hessian matrix and uses the observed behavior of $J(x_0)$ and $\nabla J(x_0)$ to build up the curvature information to make an approximation to H

using an appropriate technique. In the used optimization algorithm, the Hessian matrix update is done using the BFGS formula from the Broyden, Fletcher, Goldfard, and Shanno studies about Newton's optimization method:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T s_k^T s_k H_k}{s_k^T H_k s_k} \tag{2.65}$$

To define the direction step s_k, it is necessary to compute the Hessian matrix approximate inverse. An approximate inverse of the matrix H_k is usually obtained by applying the Sherman-Morrison formula:

$$H_{k+1}^{-1} = H_k^{-1} + \frac{\left(s_k s_k^T\right)\left(s_k^T q_k + q_k^T H_k^{-1} q_k\right)}{\left(s_k^T q_k\right)^2} - \frac{H_k^{-1} q_k s_k^T + s_k q_k^T H_k^{-1}}{s_k^T q_k} \tag{2.66}$$

This procedure is summarized in the following figure:

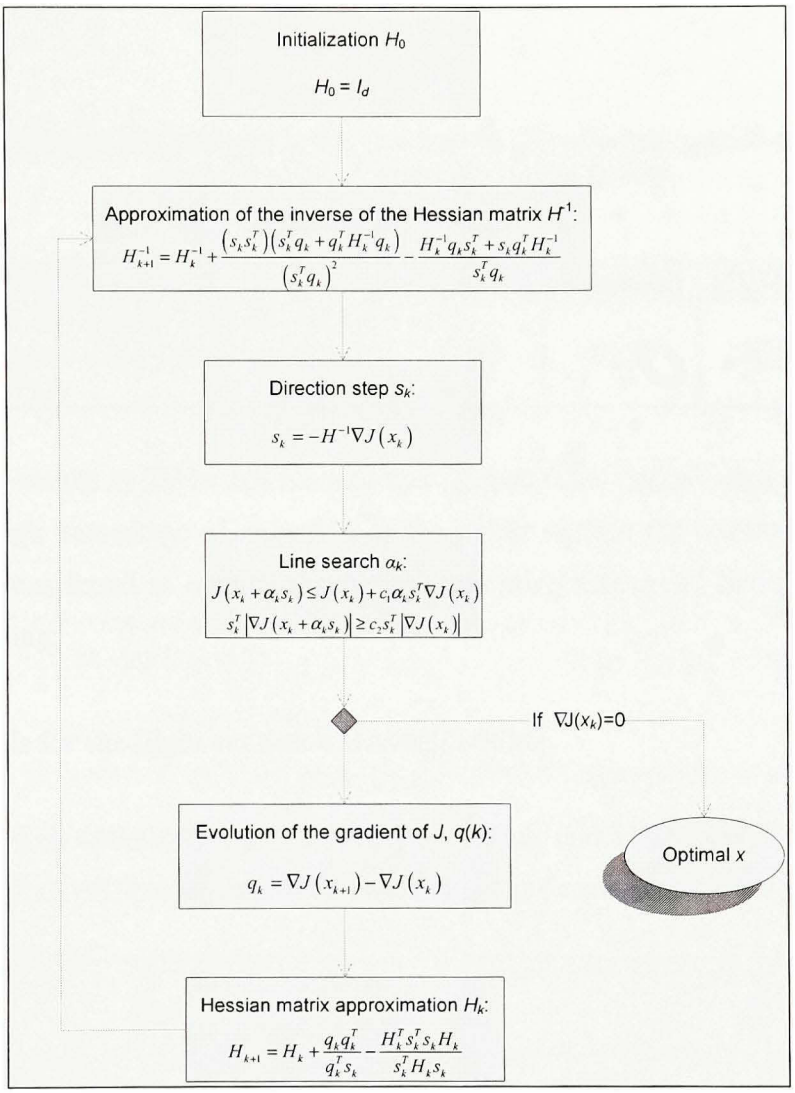

Figure 2.13 Initial state conditions optimization procedure.

The following table shows the reduction of the points outside the tolerance margins when an automatic tuning is done. The initial guesses used are the initial state variables found in the manually tuning:

Table 2.3

Influence of the initial state variables on the number (%) of points outside of the tolerance margins when an automatic tuning is done

Flight condition	HA3000ft	HA6000ft	HF3000ft	HF6000ft
Reduction of the points outside of the tolerance margins when initial conditions are optimized (%)	47.87	14.29	70.03	75.42

By observing results in Table 2.3, we note that the automatic optimization is very powerful due to the high percentage of reduction of the points outside the tolerance margins. The initial conditions found as optimal for the manual tuning were good initial guesses for the automatic tuning.

2.3.4 Results for the flight condition HA6000ft-50kts

After applying an optimization to the open-loop models and adjustment of the initial states values, the state evolutions in closed-loop were found and represented in the next figures.

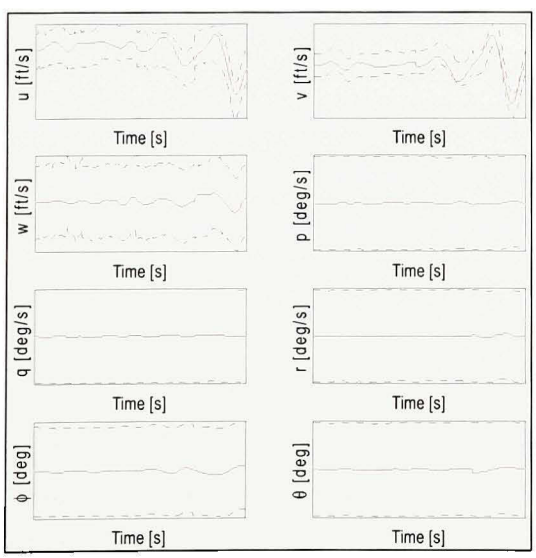

Figure 2.14 State evolution in closed-loop identification.

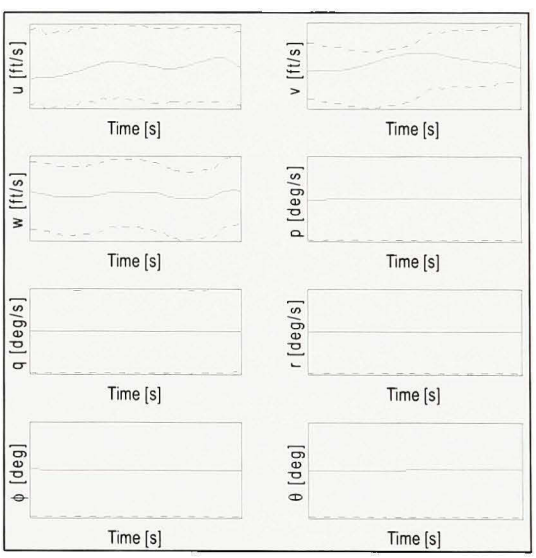

Figure 2.15 State evolution in closed-loop for the validation 1.

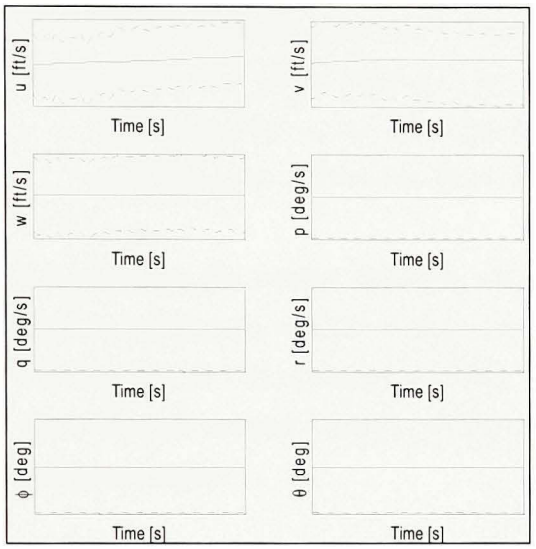

Figure 2.16 State evolution in closed-loop for the validation 2.

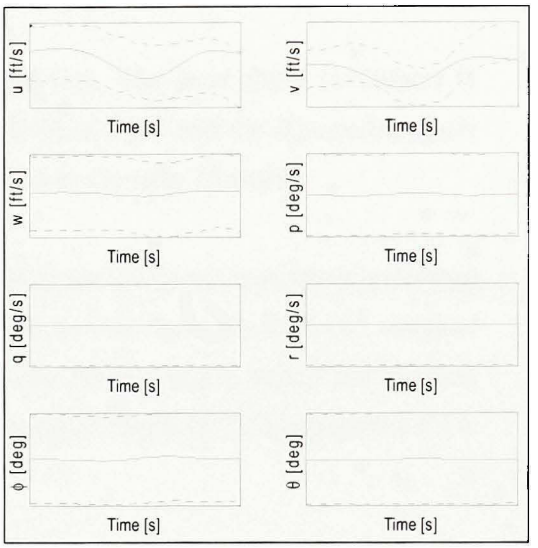

Figure 2.17 State evolution in closed-loop for the validation 3.

By observing the above figures, we note that the estimated signals stay within the tolerance margins during all time histories. The identified model is validated with three different flight tests for this flight condition.

2.4 All flight conditions synthesis

In order to synthesize the obtained results for all flight conditions, we decide to focus the analysis on the tolerance margins to be respected during the manoeuvres. Results are shown for the identification and the three validations in Appendix I, Table 1.1.

The first column presents the flight condition dependent on the helicopter loading, its altitude and speed. The second column shows the studied parameters which are the mission type, the control lengths in seconds and the tolerance margins values to be respected in open-loop (OL) and in closed-loop (CL). The other columns present the plots summary obtained in OL and in CL.

For the flight mission type, two parameters are given:

(1) The command excited by the pilot, that can be the collective, the longitudinal cyclic, the lateral cyclic or the pedals command, and;

(2) The arrows which symbolize the type of mission. The *level flight* (±500fpm) is represented by →, the *ascending* flight (+1000fpm) by ↑ and the *descending flight* (-1000fpm) by ↓. The autorotation is represented by the term "Autorot".

For each case, the number of signals which are within the tolerance margins is presented between brackets [ ] with the instant when the signals go outside of the tolerance margins. For example, [8 / 8] represents a perfect match for the model outputs versus the required FAA tolerance margins while [7 / 8] shows that one signal output (8-7) is outside the FAA tolerance margins.

By analyzing Table 1.1 in Appendix I, it can be observed that in open-loop, the estimated signals remain within the tolerance margins during all the controls duration. For each studied case, in open-loop, the model identified with 2-3-1-1 command inputs is validated three times and the estimated signals are closed to those measured more than three seconds. The FAA rules are respected in open-loop.

In closed-loop, for most of the flight conditions, all estimated signals are found within the tolerances margins. For the cases when signals are outside the tolerance margins, only one signal over eight does not stay within the tolerance margins. Yet, for all cases, the estimated signals remain within the tolerance margins at least during the first three seconds. The FAA rules are also respected in closed-loop.

Thus, with the obtained results, it was found that the dynamic behaviour of the helicopter can be described by a linear state space model.

CHAPTER 3

OUTPUT EQUATION DETERMINATION

3.1 Motivation

Generally, in order to obtain the linear accelerations from the state parameters, the following classical equations of motion are used:

$$\begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} + \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \times \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{3.1}$$

When Eq. (3.1) is applied, we did not obtain very good results. Below are the plots of the linear accelerations for the flight condition: heavy aft (HA) flight, altitude at 6,000 ft and speed of 50 knots. In each plot, the full green line corresponds to the classical method results where the dashed black lines symbolize the tolerances margins:

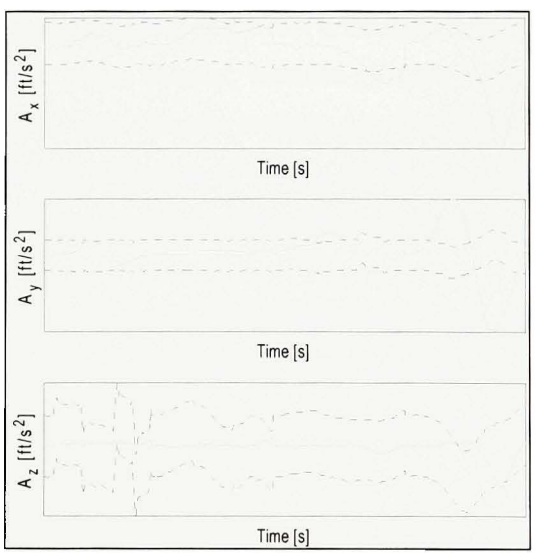

Figure 3.1 Outputs evolution with classical equations.

Graphically, we observe that the signals from the classical method do not remain within the tolerance margins during all time histories. The following Table 3.1 presents the average of the percentage of points outside the tolerance margins for all flight conditions, when the linear accelerations are calculated with classical equations.

Table 3.1

Percentage (%) of points outside the tolerance margins for all flight conditions when classical equations are used to observe the linear accelerations

HA3000ft			HA6000ft			HF3000ft			HF6000ft		
A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z
29.39	63.41	45.24	12.11	39.39	22.96	24.52	47.45	18.73	20.34	43.94	13.31

By analysis of results shown in this table, is found that the classical method is not appropriate to observe the linear accelerations calculated from the state variables. The worse results are obtained when observing the linear accelerations A_y. Thus, the theoretical model given by classical equations is limited and does not give satisfactory results. This can be explained by the fact that the instruments used to do the measure give a systematic error, a random error due to the test environment. Then, because the instruments (accelerometers for example) were not located at the helicopter center of gravity, a data pre-processing was done in order to get all data at the same point. Finally, because all data were not measured by the instruments, numerical calculi were done in order to get all necessary data, which generate error. Therefore, the white box definition of Bohlin (2006) is confirmed as the white box methods do not consider the random effects of the data environment.

Thus, other methods are set up to observe the desired outputs. For this reason, two different methods are chosen: a *nonlinear* method, and a *linear* method. In both methods, the same inputs are used as the ones in the classical equations. In a discrete study, the states variables at sample times k and $k+1$ were considered.

3.2 Nonlinear method theory

Fuzzy logic method is used to identify the MISO open loop models. Used to represent highly nonlinear relationships between inputs and outputs data, this algorithm can be split into two parts:

(1) Numerical data clustering, and;

(2) Equation coefficients relating the inputs to the outputs data.

3.2.1 Subtractive clusters

The purpose of the numerical data clustering is to define subsets so that data in the same subset are as similar as possible. Each point has a degree of belonging to clusters. Data clustering is used for statistical data analysis. This process summarizes data in order to produce a concise system behaviour representation. Two parameters are used: the *clusters centers* and the *standard deviation* of the inputs and outputs signals.

3.2.1.1 Parameters definition

In order to define the clusters, we need all inputs and outputs signals. The inputs and outputs data are gathered in a unique variable called X:

$$X = \left[X_{in} \quad X_{out} \right]_{[s \times (n+m)]} \tag{3.2}$$

where s is the points number of the set of signals, $(n + m)$ is the number of inputs and outputs, and where X_{in} and X_{out} are the inputs and outputs signals.

3.2.1.2 Standard deviation

The standard deviation σ is defined as follows:

$$\sigma_{[s\times 1]} = \frac{radii(maxX - minX)}{2\sqrt{2}} \tag{3.3}$$

where *radii* specifies the cluster size in each data dimension, and where *minX* and *maxX* are the minima and maxima of the inputs and outputs signals. Because all signals have the same power, the parameter *radii* is fixed as the same constant for all data. Thus, a value of the standard deviation is obtained for all inputs and outputs data.

3.2.1.3 Cluster centers

The algorithm steps are the following:

(1) The signals normalization;

(2) The potential value calculation for each sample time;

(3) The selection of the first highest potential called the reference potential and;

(4) The comparison of the highest potential value with the reference value.

Depending on this ratio value, we conclude on the validity of this point as a cluster. Finally, we calculate the new potential values and perform the same steps as previously until all potential values decrease to zero.

3.2.1.3.1 Data normalization

In order to define the true relationships between the inputs and outputs data, these data are normalized in the same range of values i.e. the signals are bounded between the two values. Due to the fact that no specific range is specified, we use the minima and maxima values to normalize data:

$$X_{normalized} = \frac{X_{real} - minX}{maxX - minX} \tag{3.4}$$

If the minima and maxima values have a zero range, we calculate a small range relative to these data, which is different from zero:

$$maxX_{new} = maxX_{old} - 0.0001 \times \left(1 + \left|maxX_{old}\right|\right) \tag{3.5a}$$

$$minX_{new} = minX_{old} - 0.0001 \times \left(1 + \left|minX_{old}\right|\right) \tag{3.5b}$$

The normalization is validated by use of the following three "if-then" rules:

$$if \quad X_{old} < 0, \; then \quad X_{new} = 0 \tag{3.6a}$$

$$if \quad X_{old} > 1, \; then \quad X_{new} = 1 \tag{3.6b}$$

$$if \quad 0 < X_{old} < 1, then \quad X_{new} = X_{old} \tag{3.6c}$$

3.2.1.3.2 Potential value

Each data point could be a cluster center (Chui, 1994), therefore the potential of each data point is calculated with a likelihood measure that each data point has with its surrounding data points. Researching of a cluster center by considering that all data point could be a cluster center is named *subtractive clustering*. The potential values for each sample time are calculated with following formulas:

$\forall i \in [\![1; s]\!]$,

$$potVals(i) = \sum_{i=1}^{s} e^{-4 \sum_{j=1}^{(n+m)} (dx_{i,j})^2} \tag{3.7}$$

where $dx = \begin{bmatrix} X(i,1)-X(1,1) & \ldots & X(i,(n+m))-X(1,(n+m)) \\ \ldots & \ldots & \ldots \\ X(i,1)-X(s,1) & \ldots & X(i,(n+m))-X(s,(n+m)) \end{bmatrix}$.

Thus, because of the negative exponential, the higher the potential value is, the closest the point selected at the i^{th} sample time is near all points of the signals X.

3.2.1.3.3 Maximum potential value

After calculation of the all potentials values, the maximum potential value is selected; its value is defined by *maxPotVal* and its index *maxPotIndex*. For the first cluster, the highest potential value is called the reference potential, *refPotVal*. Then, we associate the maximum potential value *maxPotVal* and its index *maxPotIndex* to the signal points which have the same indices:

$$maxPoint = X(maxPotIndex,:) \tag{3.8}$$

The highest potential value is associated to the following ratio:

$$maxPotRatio = \frac{maxPotVal}{refPotVal} \tag{3.9}$$

For the first cluster, the highest potential value is assigned to the reference potential value so that the ratio is equal to one. The other potentials are lower than the reference potential value so that the ratio decreases to zero and another cluster could be found.

3.2.1.3.4 Conditions to accept or reject a point as a cluster

Due to the fact that the ratio decreases and trends to zero, the limits are fixed in order to accept or reject a point as a cluster. Two thresholds are then introduced:

(1) The *accept ratio* for the above potential, which will be definitively accepted by the data point as a cluster center and;

(2) The *reject ratio* for the below potential, which will be definitively removed by the data point from the cluster centers list.

If the ratio falls in the region between the two thresholds, a new minimum distance *minDist* is defined in order to check if the data point has a high potential enough and far enough from the previous cluster centers:

$$minDist = \sqrt{dxSq} \tag{3.10}$$

where:

$$dxSq_{[1\times 1]} = \left(\left(maxPoint - centers\left(i,:\right)\right).\times accumMultp\right)^2 \tag{3.11}$$

If *maxPotRatio* + *minDist* < 1, then the point is not a cluster and the potential value is set to zero. Otherwise, this point is added to the clusters list.

3.2.1.3.5 New potentials

The new potential values are functions of the previous potential values as shown in the following equation:

$$potVals_{new} = potVals_{old} - maxPotVal_{old} \times e^{-4\sum_{i=1}^{(n+m)}\left(ddx_{i,j}\right)^2} \tag{3.12}$$

Where:

$\forall l \in [\![1; \text{numPoints}]\!]$,

$$\begin{cases} maxP\left(l\right) = maxPoint \\ ddx = \left(maxP - X\right).new\_sqshMultp \end{cases} \tag{3.13}$$

A part of the previous maximum potential value is subtracted from each data point as a function of its distance from the previous cluster center. With the use of the negative exponential, the point near the previous cluster center will have greatly reduced potential and will not be used as a cluster center. The factor *new_sqshMultp* will represent the distance from a center and will allow a high potential reduction.

When the potential of all data points has been updated with Eq. (3.12), all negative potential values are arranged to zero and the data point is selected with the highest remaining potential values. The loop is realized until all potential values become equal to zero. This algorithm part is summarized on the next figure:

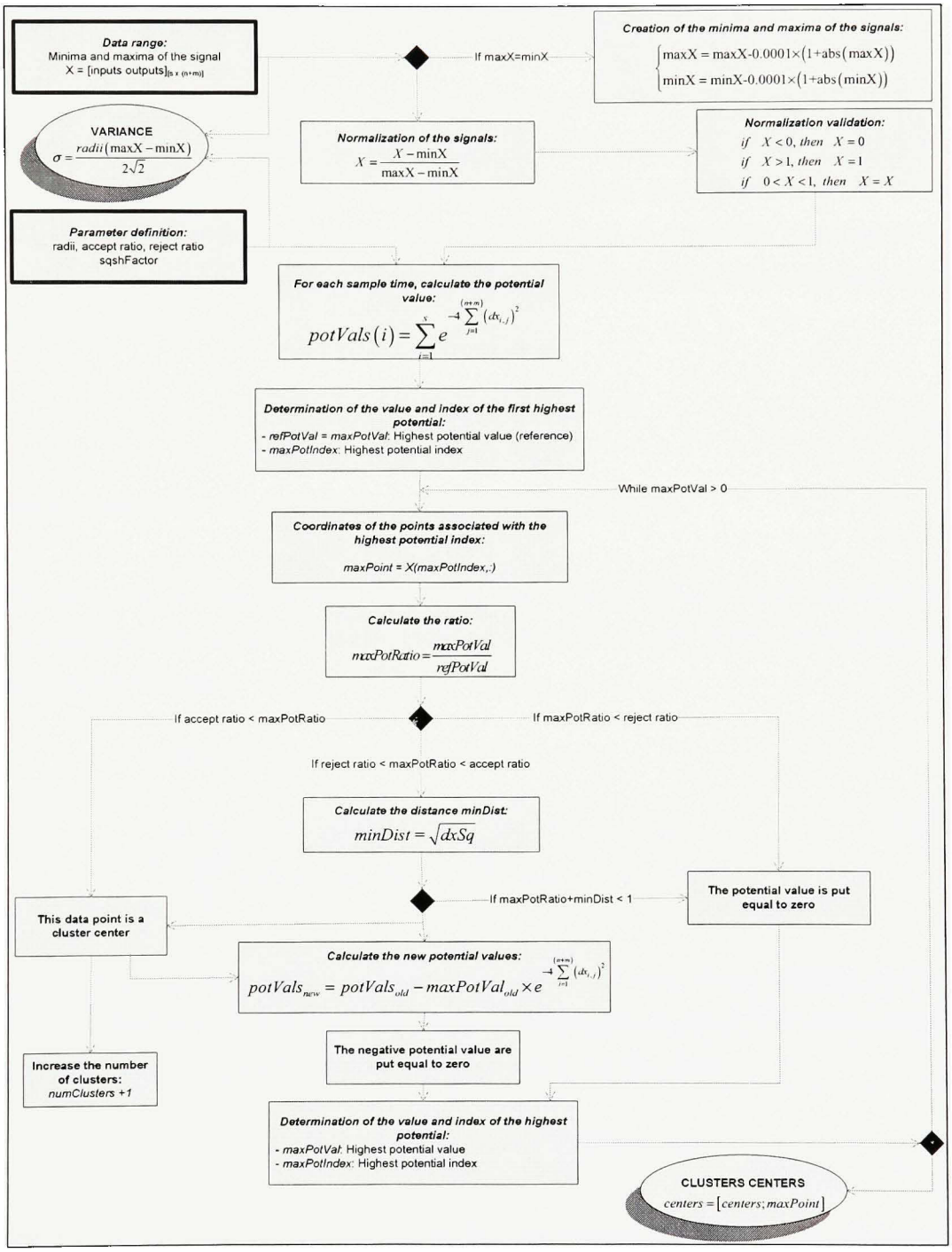

Figure 3.2 Subtractive clustering estimation.

3.2.2 Linear function between the inputs and outputs data

Each cluster center is considered as a fuzzy rule that describes the system behaviour. In classical form, rules or membership functions take only the value of 1 (member) or 0 (non-member). Yet, with fuzzy algorithms, membership functions take infinity of values which indicate the membership of an element to a subset. Based on a negative exponential, the Gaussian membership functions give more accurate optimized models. Thus, the fuzzy rules are defined as follows:

$$muVals_{[s\times 1]} = e^{-\left(\frac{Xin-centers}{\sqrt{2}\sigma}\right)^2} \tag{3.14}$$

where *centers* and σ are estimated in the first part of the algorithm.

The fuzzy rules are used to define the normalized *muMatrix* matrix which represents the concordance degree between all data points and data used as center clusters for representing the system behaviour.

$$muMatrix\left(i,(j-1)(n+1)+1:(n+1)j\right) = \left[\frac{Xin(i,:)\times muVals\left(i,(n+1)j\right)}{\sum_{k=1}^{numRule} muVals\left(i,(n+1)k\right)} \quad \frac{muVals\left(i,(n+1)j\right)}{\sum_{k=1}^{numRule} muVals\left(i,(n+1)k\right)}\right]_{[s\times(numRule.(n+1))]} \tag{3.15}$$

By use of this matrix, is possible to calculate the coefficients of the linear algebraic equation relating the inputs to outputs data. The following equation has been used in order to obtain the coefficients, due to the fact that the *muMatrix* matrix is not square:

$$outEqns = \left(muMatrix^T muMatrix\right)^{-1} muMatrix^T .Xout \tag{3.16}$$

These coefficients can also be obtained by the least squares method (Tagaki and Sugeno, 1985). The second part of the algorithm can be summarized in the following figure:

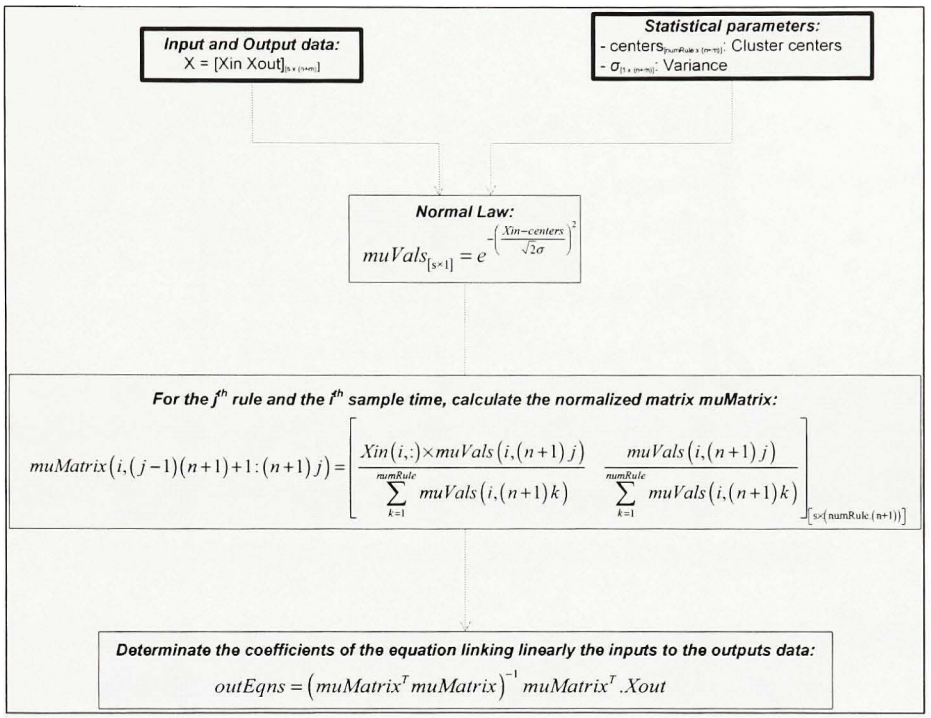

Figure 3.3 Linear equation estimation.

3.2.3 Fuzzy system training

In order to build a fuzzy system, for each input, the correct number of membership function has to be found so that the estimated output fit the measured output signals. This process is achieving by a trial-and-error iterative algorithm, which is a combination of the least-square method and the back propagation gradient descent methods. The cost function used to define the output error is expressed as root mean squared errors between the estimated and the measured signals. The following figure shows the cost function evolution according to the iterations for the flight condition HA6000ft-50kts.

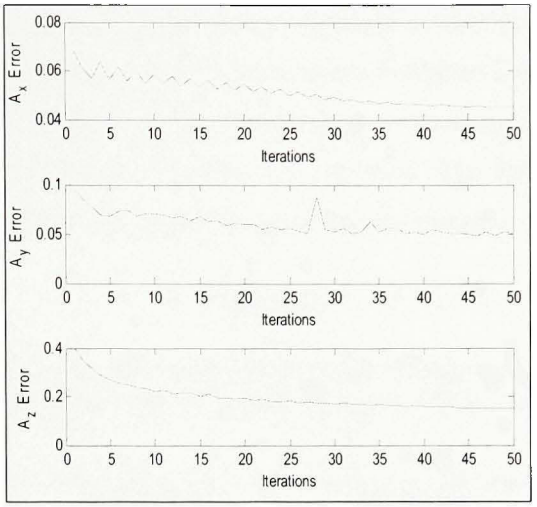

Figure 3.4 Evolution of the output error during the training.

By observing Figure 3.4, the fuzzy system training allows the achievement of better results by minimizing the output error.

3.3 Linear method theory

The second method used to observe the output evolution from the state variables is a linear method. Because there is a recurrence relationship between the inputs (state variables at sample time k and k+1) and the outputs, we use the same method as the one explained in Section 2.2. The following table presents the percentage of points outside the tolerances margins.

Table 3.2

Percentage (%) of points outside the tolerance margins for all flight conditions when a linear block is used to observe the linear accelerations

HA3000ft			**HA6000ft**			**HF3000ft**			**HF6000ft**		
A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z
3.28	0.58	11.82	0	0	1.68	0.75	0.13	4.81	0.77	0.07	4.75

Although the averages expressed in percentage of points outside the tolerances margins are lower than the percentage when classical equations are used (see Table 3.1), an optimization based on neural network theory is applied in order to decrease these results. The same kind of optimization as the one detailed in Section 2.3.2 is used. The following table shows the evolution of the percentage of points outside the tolerances margins:

Table 3.3

Reduction (%) of the percentage of points out of the tolerance margins for all flight conditions when linear method is used to observe the linear accelerations

HA3000ft			**HA6000ft**			**HF3000ft**			**HF6000ft**		
A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z
-14.8	+37.4	-18.7	0	0	-23.5	-6.36	-7.27	-53.4	0	0	-49.6

In the above table, the minus sign (-) shows that the optimization gives the better results and the plus sign (+) shows that the optimization does not improve the results.

No improvement is found for the A_y signals for the flight condition HA3000ft. Yet, graphically, the estimated signals come closer to the tolerance margins, which mean that the fit coefficient becomes higher. For an altitude of 6,000 ft, there is no improvement, which can be explained by the fact that the results were already very good and the optimization did not give significant improvements because the solution was optimal.

For the A_z signals, for the four flight conditions, a significant improvement is shown.
In the next Sections, the linear method optimized with a Neural Network (NN) method will be denoted by "Linear NN".

3.4 Practical application

Independently of the state equation, we use the fuzzy logic and linear methods to estimate the output variables by use of the measured states $x(k)$ as shown in the following figure:

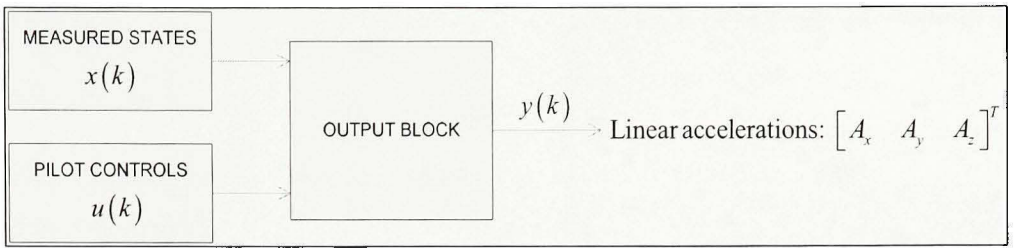

Figure 3.5 Simulation of the output equation.

We observe that linear accelerations do not depend on the Euler angles so that we do not consider the Euler angles as inputs of the fuzzy and linear blocks.

3.5 Results

The next figures show the estimated outputs calculated from the three following methods:

(1) In full green lines, the classical equations;

(2) In full blue lines, the linear method and;

(3) In full red lines, the nonlinear method.

The flight condition used is the same as in the other figures i.e. HA6000ft-50kts.

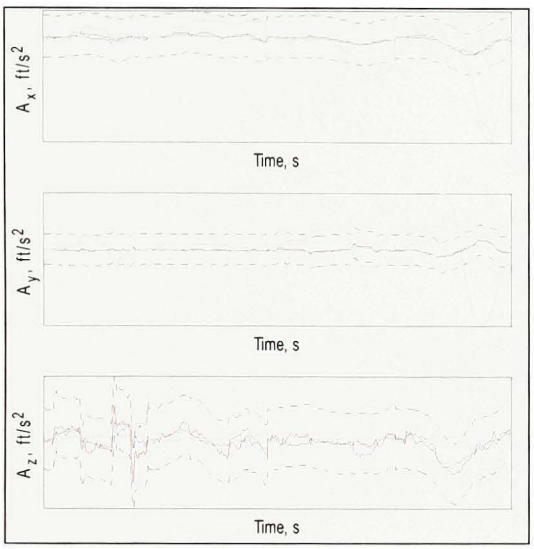

Figure 3.6 Outputs evolution for the model identification.

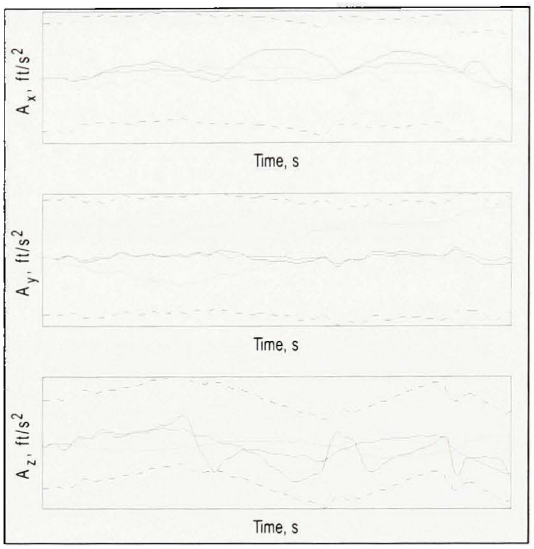

Figure 3.7 Outputs evolution for the model validation 1.

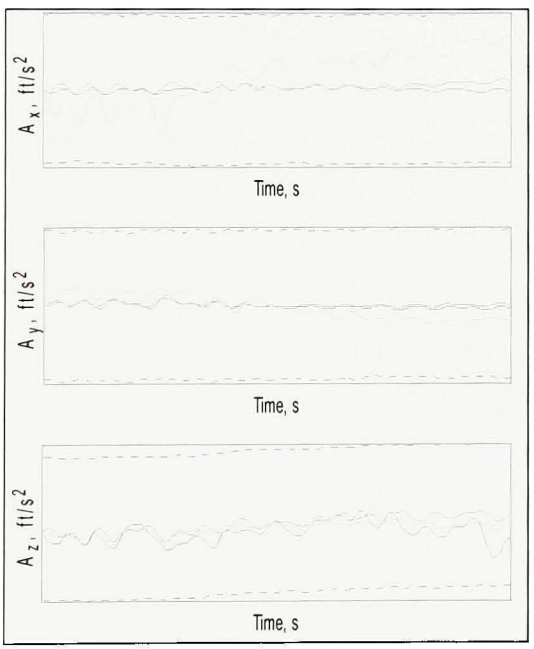

Figure 3.8 Outputs evolution for the model validation 2.

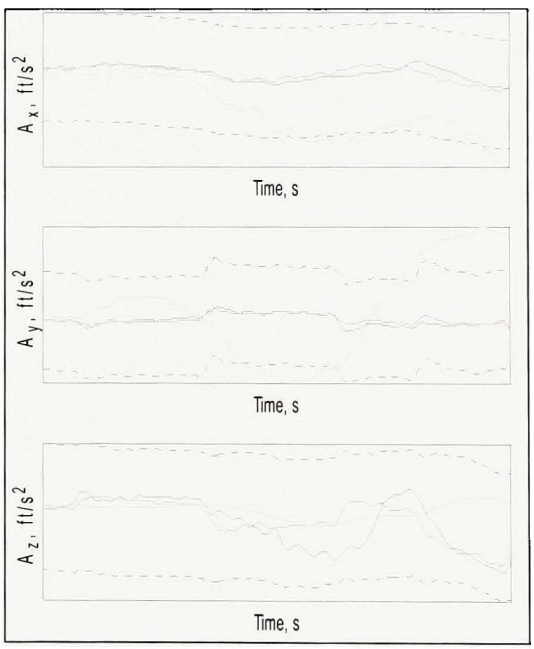

Figure 3.9 Outputs evolution for the model validation 3.

The following table presents the average in percentage of the points outside the tolerance margins for the three methods.

Table 3.4

Percentage average of the number of points outside the tolerance margins for the output equation (%)

	HA3000ft			**HA6000ft**			**HF3000ft**			**HF6000ft**		
	A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z
Classic	29.39	63.41	45.24	12.11	39.39	22.96	24.52	47.45	18.73	20.34	43.94	13.31
Fuzzy	0.09	0.23	2.48	0	0	0	0	0	0	0	0	0
Linear	3.28	0.58	11.82	0	0	1.68	0.75	0.13	4.81	0.77	0.07	4.78

The classical equations give worst results in comparison with the fuzzy logic and the linear methods. With a high percentage of points outside the tolerance margins, we can also note that using the classical equations lead up to a high negative correlation coefficient which means that the estimated signals have the opposite trends of the measured signals. If we take into account the pilot feeling, this reaction will not be appreciated by the pilot. When observed outputs are estimated from the *measured* states signals, the fuzzy logic method seems to give the best results. The percentage of points outside the tolerance margins is close to zero. In order to choose the best method, a comparison between the three methods is done. This comparison will attribute a score for each of the three methods in order to get an objective conclusion.

3.6 Comparison between the three methods

3.6.1 Methodology

Because this model could be implemented in a simulator for the pilot training for example, the comparison between the three methods to observe the system outputs from the system

states is based on the pilot feedback. Thus, the comparison concentrates on the plots particularities, which can be felt by the pilot. Several criteria are considered:

(1) The *graphs aspects*: if there is a sudden change of the signal evolution (whereas there is no high modification of the measured signal slope), the pilot may feel it. Two parameters are studied: the *speed* of the error, and the *acceleration* of the error. If the speed of the error between the estimated and measured signals is not null and increases, then it seems that the slope of the error curve is increasing, which means that the estimated signals move away from the tolerance margins. Therefore, if the acceleration of the error is not null and trends to high value, it means that the estimated signals have a peak which can be felt by the pilot. If one of these methods generates such characteristics, the method must be penalized because it does not exist in reality.

(2) The *number of points outside the tolerance margins*: the FAA defines tolerance margins such that the pilot does not feel difference with the reality if the signals remain within the tolerance margins;

(3) The *fit* coefficient: This parameter is a function of the error position. The higher is this coefficient, the better is the method and the model better represents the reality;

(4) The *correlation* coefficient: This parameter gives information about the signal trends. The higher is the correlation fit, the better is the method. However, this coefficient is not a sufficient criterion to conclude on the method goodness. In order to obtain a score in percentage, the negative correlation coefficients are set to zero.

(5) The amount of time MATLAB/SIMULINK takes to complete the *output block generation*: To run the simulations, SIMULINK is used. The three methods are arranged in different schemes in order to know the time taken to compute the simulation. Thus, this criterion is not a critical characteristic of the method but it can be a critical criterion in order to choose the best method among the three proposed in the case the other parameters are nearly the same for the three methods.

(6) The *sensitivity of the initial state conditions*: The first fifty points (i.e. 1 s) are considered. In the open-loop study, this criterion is not significant because the initial state conditions are set to zero. In the closed-loop study, as said in section 2.3.3, the

tuning of the initial state conditions is explained. Optimal initial state conditions are obtained with a manual and an automatic tuning. This procedure has a direct consequence of the output results (observed linear accelerations) because the linear accelerations are functions of the state variables. If a model is highly sensitive to a slightly change of the initial state conditions, then it means that the model has a weakness in robustness.

Because these criteria do not have the same importance, weighting coefficients are associated to each criterion, and are gathered in the following table:

Table 3.5

List of the criteria weights for the comparison between the three methods in order to observe the outputs variables from the state variables

	Criteria	Weight [%]
1	Sudden change of the signals evolution: Acceleration of the error between the measured and estimated signals	**35**
2	Number of points outside the tolerance margins	**25**
3	Fit coefficient	**15**
4	Correlation coefficient	**10**
5	Output block generation	**2.5**
6	Graphs aspects: Velocity of the error	**7.5**
7	Sensitivity to the initial state conditions	**5**

A high importance is set to the error acceleration between the measured and estimated signals and the number of points outside the tolerance margins, which can be explained by the FAA tolerance margins. The pilot does not feel changes with reality if the signals remain within the tolerances margins.

The fit and correlation coefficients, and the graphs aspects are then the most important criteria. Indeed, the better are these coefficients then the better is the method. By empirical analysis, the correlation coefficient has a lower weight than the fit coefficient because an

estimated signal may have a correlation coefficient of 100% and be totally outside the tolerance margins contrarily to the fit coefficient. Then, good fit coefficient and plots aspect imply that the estimated signals are in the tolerance margins.

The sensitivity of the initial state conditions is a criterion which shows the robustness of the method to slightly changes of the state variables. Its weight represents 5% of the final score because the initial states conditions only have a consequence on the first sample times. Indeed, when the matrix A is stable (i.e. its eigenvalues are in the unitary circle for a discrete-time system), then the state at the n^{th} sample time is influenced by the initial state by a coefficient A^n, which is a decreasing coefficient with sample time (see Section 2.3.3).

The final criterion defined in this comparison is the amount of time to compute the output block with the three methods. It has a poor weight because it has an importance if the same score is obtained by the three methods.

In the next Section, the detailed results obtained for the flight condition HA6000ft-50kts are presented. Then, a summary of all flight conditions is shown.

3.6.2 Results for the flight condition HA6000ft-50kts

By applying these criteria to the graphs obtained in Figure 3.6 to Figure 3.9, the following tables show the detailed score for each of the three methods for the identification and validations parts.

Table 3.6

Score (%) obtained by the three methods for the identification of the flight condition HA6000ft-50kts in the output equation

	Methods	Criteria							Total
		1	2	3	4	5	6	7	
A_x	Fuzzy	100	100	91.10	98.29	0	99.26	100	95.94
	Linear NN	100	100	80.68	92.18	100	98.15	100	**96.18**
	Classic	25	83.68	24.91	63.43	100	0	100	47.25
A_y	Fuzzy	100	100	93.62	99.19	0	100	100	96.46
	Linear NN	100	100	87.06	96.95	100	100	100	**97.75**
	Classic	25	72.16	1.34	0	100	0	100	34.49
A_z	Fuzzy	100	100	88.16	97.03	0	47.56	100	**91.49**
	Linear NN	50	99.70	65.95	78.44	100	32.05	100	70.07
	Classic	0	94.24	31.58	5.16	100	0	100	36.31

By observing the above table, we note that the *classical equations* obtain the worst results with final scores smaller than 50%. This method is highly penalized because of the error acceleration (criterion 1) i.e. the "peaks", which can be felt by the pilot outside the tolerance margins. Then, the global aspects of the plots, which are the error speed, the fit and correlation coefficients, have decrease the final score of the method. Indeed, this is because of the high number of points outside the tolerance margins, and the high amplitude of the A_x and A_y plots (see Figure 3.6) comparing to the measured signals. The good scores on the sensitivity of the initial conditions and the output block generation do not significantly improve the goodness of the method because of the small weights according to these criteria.

The *fuzzy logic* and *linear* methods give better results than the classical equations. These two methods give similar results for A_x and A_y plots with a final score higher than 95%. The linear method is penalized because of the general aspects of the A_z plot. Indeed, comparing to the fuzzy logic method, the linear method has not a good fit and correlation coefficients. Then, the error speed is not high because of the small oscillations observed although these oscillations are within the tolerance margins.

Finally, the linear method is highly penalized by the error acceleration, which corresponds to the peak but also to the "anti-peaks". An "anti-peak" is called so when the measured signal has a high slot but the estimated signal does not so that numerically, there is a peak felt by the pilot referring to the measured signal.

The following table summarizes the scores obtained by the three methods for the validation 1:

Table 3.7

Score (%) obtained by the three methods for the validation 1 of the flight condition HA6000ft-50kts in the output equation

	Methods	Criteria							Total
		1	2	3	4	5	6	7	
A_x	Fuzzy	100	100.00	39.35	12.31	0.00	94.62	100	81.73
	Linear NN	100	100.00	60.57	57.82	100.00	94.62	100	**89.46**
	Classic	100	100	39.01	3.02	100	89.25	100	80.35
A_y	Fuzzy	100	100	47.86	46.86	0	100	100	86.87
	Linear NN	100	100	54.80	76.45	100	100	100	**90.87**
	Classic	100	100	4.48	0	100	87.46	100	74.73
A_z	Fuzzy	75	98.92	43.62	61.96	0	13.98	100	72.27
	Linear NN	100	100	63.47	78.80	100	96.42	100	**92.13**
	Classic	100	100	16.13	0	100	83.87	100	76.21

As identification, *classical equations* obtain the worst results even if the estimated signals remain within the tolerances margins during all time histories. The method is penalized by the fit and correlation coefficients. The correlation coefficients are set to zero for A_y and A_z in order to symbolize that the coefficients were negative, which means that the trends of the estimated signals from classical equations were in the opposite side of the measured ones.

The *fuzzy logic* and *linear* methods obtain better scores than the classical equations. The fuzzy logic method is penalized by the fit and correlation coefficients for the three variables

and by the error velocity for the A_z plot. Indeed, in the last plot, high difference of the slope is observed in the tolerance margins.

The following table summarizes the scores obtained by the three methods for the validation 2:

Table 3.8

Score (%) obtained by the three methods for the validation 2 of the flight condition HA6000ft-50kts in the output equation

	Methods	Criteria							Total
		1	2	3	4	5	6	7	
A_x	Fuzzy	100	100	16.17	0	0	100	100	77.43
	Linear NN	100	100	20.40	8.85	100	84.69	100	**77.80**
	Classic	100	100	4.60	10.68	100	0	100	69.26
A_y	Fuzzy	100	100	16.79	0	0	97.45	100	77.33
	Linear NN	100	100	17.53	0	100	100	100	**77.63**
	Classic	100	100	4.17	0	100	71.94	100	73.52
A_z	Fuzzy	100	100	25.40	38.02	0	31.12	100	77.45
	Linear NN	100	100	69.77	87.63	100	74.49	100	**92.32**
	Classic	100	100	41.04	76.20	100	100	100	88.78

The validation 2 is a test of no-command of the pilot. The method results are more similar. However, the linear method obtains the best final scores.

The following table summarizes the scores obtained by the three methods for the validation 3:

Table 3.9

Score (%) obtained by the three methods for the validation 3 of the flight condition HA6000ft-50kts in the output equation

	Methods	Criteria							Total
		1	2	3	4	5	6	7	
A_x	Fuzzy	100	100	67.26	78.32	0	100	100	92.92
	Linear NN	100	100	71.86	86.22	100	100	100	**94.40**
	Classic	50	79.24	39.84	88.48	100	94.15	100	66.70
A_y	Fuzzy	100	100	61.01	74.01	0	100	100	91.55
	Linear NN	100	100	58.82	81.76	100	100	100	**92.00**
	Classic	50	57.02	7.85	0	100	28.36	100	42.56
A_z	Fuzzy	100	100	50.48	81.82	0	79.53	100	89.22
	Linear NN	100	100	59.00	81.04	100	98.54	100	**91.84**
	Classic	100	100	42.52	0	100	100	100	81.38

The *classical equations* obtain the worst results mainly because of three criteria: the error acceleration, the number of points outside the tolerance margins and the negative correlation coefficients. The two other methods have approximately the same high scores.

By gathering the results found for this flight condition, the *linear* method is the best method. The comparison is done for all flight conditions. Thus, four tables like them are filled out for each flight condition. In order to summarize the global results, an average of the results is done for each method and for each output.

3.6.3 Results for all flight conditions

For the A_x, A_y and A_z variables, the average of the score obtained by each of the three methods and the means for the three variables are summarized on the following table:

Table 3.10

Summary of the three method goodness for the output equation (%)

	A_x	A_y	A_z	Total
Fuzzy	83.04	89.30	66.35	79.56
Linear NN	85.46	90.30	70.16	**81.97**
Classic	63.97	40.61	56.28	53.62

By analyzing the above table, we observe that the fuzzy logic and linear methods are more powerful than the classical equations of motion for the output equation. Therefore, the linear method stands out.

The main goal of this project is to identify and validate a global model for which the estimated states variables are function of the measured pilot commands only and the estimated outputs are function of the estimated states variables only. The methods comparison for this model allows us to conclude about the real best method to observe the outputs with the estimated signals.

CHAPTER 4

GLOBAL MODEL

4.1 Formulation

In this section, we group the two blocks presented in Chapter 2 and Chapter 3 in order to create a global model in which the observed outputs are computed from the estimated states and the measured pilot commands. The following scheme is then simulated:

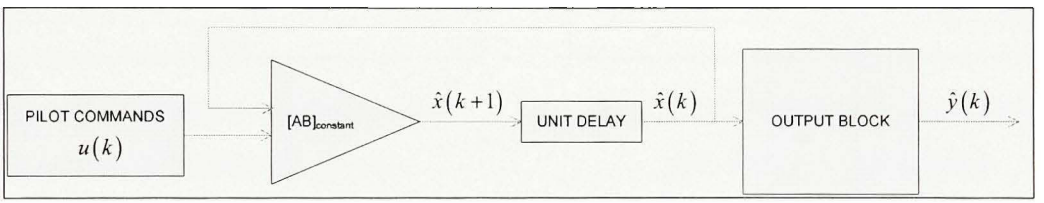

Figure 4.1 Global model simulation.

By use of this simulation, we are able to test the robustness of the output blocks found in Chapter 3 to observe the linear accelerations.

4.2 Results for the flight condition HA6000ft-50kts

Figure 4.2 to Figure 4.5 show the accelerations results when the output block inputs are the estimated states. In full green lines are represented the classical equations of motion, in full red lines the fuzzy logic method and in full blue lines the linear Neural Network method. The dashed black lines represent the tolerance margins.

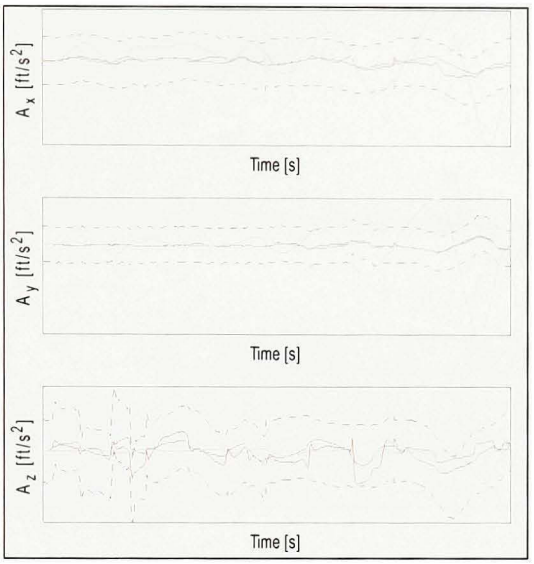

Figure 4.2 Output evolution for the global model identification.

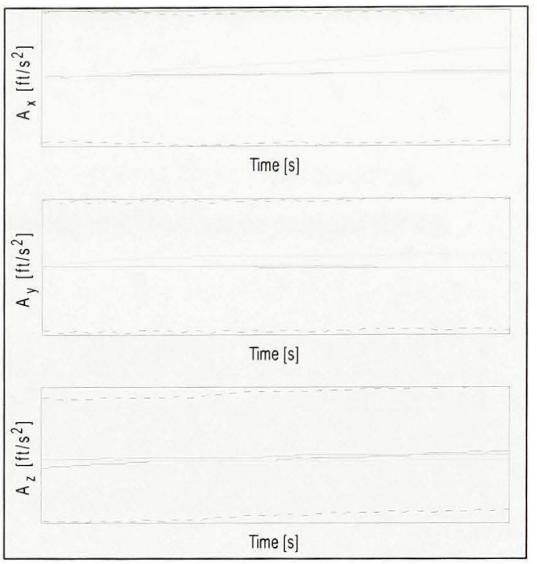

Figure 4.3 Output evolutions for the global model validation 1.

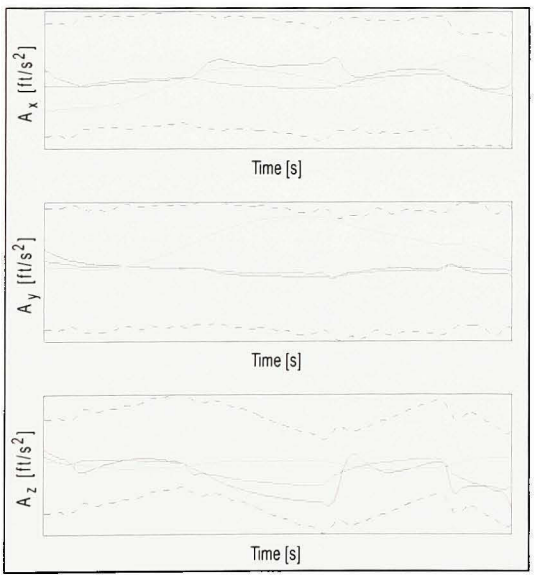

Figure 4.4 Output evolution for the global model validation 2.

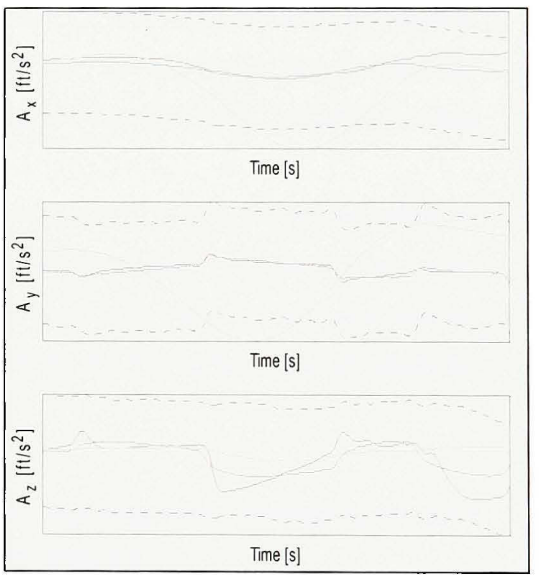

Figure 4.5 Output evolution for the global model validation 3.

The following table summarizes the percentage of points outside the tolerance margins for all studied flight conditions.

Table 4.1

Percentage (%) averages of the number of points outside of the tolerance margins for the global model

	HA3000ft			**HA6000ft**			**HF3000ft**			**HF6000ft**		
	A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z	A_x	A_y	A_z
Classic	25.36	53.02	43.30	11.26	38.88	18.42	18.7	40.75	17.55	17.04	45.1	12.22
Fuzzy	7.01	2.79	31.04	0.92	0.37	12.51	0.79	0.08	4.81	1.75	0	8.60
Linear NN	3.54	0.78	22	0.04	0	3.59	0.61	0.13	4.17	0.01	0.77	6.1

This table shows the consequences of considering the estimated state variables instead of the measured ones. If results obtained in Table 4.1 and Table 3.4 are compared, then we can observe that the number of points outside the tolerance margins changed. For the *classical equations*, we note that the percentage of points outside the tolerance margins slightly decrease when the global model is compiled. This fact can be explained by the fact that the estimated state signals given as input of the output equation are less noisy, smoother, so that the classical block does not cope little oscillations.

For the *fuzzy* block, there is a high increase of the percentage of points outside the tolerance margins, which shows that the fuzzy block is not robust enough to overcome the state changes. For the *linear* block, the percentage of points outside the tolerance margins remains the same, which shows that the linear block surmounts the random effect of the environment when the data are gathered.

By applying the same reasoning as in Chapter 3, the following tables summarize the final score of the three methods for the flight condition HA6000ft-50kts:

Table 4.2

Score (%) obtained by the three methods for the identification of the flight condition HA6000ft-50kts for the global model

	Methods	Criteria							Total
		1	2	3	4	5	6	7	
A_x	Fuzzy	100	100	66,08	77,36	0	90,77	100	89,46
	Linear NN	100	100	66,58	81,29	100	99,63	100	**93,09**
	Classic	50	77,99	22,18	48,59	100	42,39	100	55,86
A_y	Fuzzy	100	100	79,58	92,11	0	94,09	100	93,20
	Linear NN	100	100	81,87	94,24	100	95,20	100	**96,34**
	Classic	25	73,63	1,35	0	100	0	94,67	34,59
A_z	Fuzzy	75	99,41	60,66	67,02	0	0	89,33	**71,37**
	Linear NN	0	99,48	58,92	70,85	100	0	100	48,29
	Classic	0	96,23	32,12	6,75	100	0	100	37,05

Table 4.3

Score (%) obtained by the three methods for the validation 1 of the flight condition HA6000ft-50kts for the global model

	Methods	Criteria							Total
		1	2	3	4	5	6	7	
A_x	Fuzzy	100	100	46,34	23,58	0	94,62	100	81,41
	Linear NN	100	100	39,02	61,20	100	100	100	**86,97**
	Classic	100	100	17,43	0	100	87,46	100	76,67
A_y	Fuzzy	100	100	36,70	20,95	0	100	100	80,10
	Linear NN	100	100	49,53	47,17	100	100	100	**87,15**
	Classic	100	100	7,15	0	100	100	100	76,07
A_z	Fuzzy	100	100	36,49	54,65	0	15,77	100	77,12
	Linear NN	100	100	42,91	55,99	100	94,62	100	**86,63**
	Classic	100	100	15,79	0	100	85,66	100	76,29

Table 4.4

Score (%) obtained by the three methods for the validation 2 of the flight condition HA6000ft-50kts for the global model

	Methods	Criteria							Total
		1	2	3	4	5	6	7	
A_x	Fuzzy	100	100	16,76	8,38	0	100	100	75,85
	Linear NN	100	100	20,39	15,40	100	100	100	**79,60**
	Classic	100	100	6,11	14,36	100	100	100	77,35
A_y	Fuzzy	100	100	29,36	0	0	100	100	76,90
	Linear NN	100	100	29,32	0	100	100	100	79,40
	Classic	100	100	11,83	51,15	100	100	100	**81,89**
A_z	Fuzzy	100	100	58,85	92,53	0	100	100	90,58
	Linear NN	100	100	73,06	95,78	100	100	100	**95,54**
	Classic	100	100	4,17	24,70	100	100	100	78,10

Table 4.5

Score (%) obtained by the three methods for the validation 3 of the flight condition HA6000ft-50kts for the global model

	Methods	Criteria							Total
		1	2	3	4	5	6	7	
A_x	Fuzzy	100	100	32,93	13,01	0	100	100	78,74
	Linear NN	100	100	62,91	66,30	100	100	100	**91,07**
	Classic	100	78,36	35,88	38,13	100	100	100	78,79
A_y	Fuzzy	100	100	55,80	70,93	0	100	100	87,96
	Linear NN	100	100	63,46	81,06	100	100	100	**92,63**
	Classic	50	78,65	7,46	0	100	83,92	100	52,08
A_z	Fuzzy	100	100	49,35	68,85	0	89,77	100	86,02
	Linear NN	100	100	60,39	68,69	100	100	100	**90,93**
	Classic	100	100	28,81	11,03	100	100	100	80,42

In general, the scores are lower for the global model than for the output model, which is because the estimated states are not exactly the same as the measured ones even if they are inside the tolerance margins. Then, we observe that the results are smoother, which can be explained by the less noisy estimated states than the measured ones.

The linear method obtains the best final scores with a higher difference with respect to the fuzzy logic method.

4.3 Results for all flight conditions

The following table summarizes the results for the three methods for the global model.

Table 4.6

Summary of the three method goodness for the global model (%)

	A_x	A_y	A_z	Total
Fuzzy	67,52	75,95	49,57	64,35
Linear NN	83	85,05	67,18	**78,41**
Classic	67,98	42,36	55	55,11

As in the output equation study, the *classical equations* give the worst results in comparison with the fuzzy logic and the linear methods in the global model. When the behaviour of the *fuzzy logic* block is studied, in which the inputs are the *estimated* states signals, we notice that the estimated outputs are not good enough. Sudden jumps are observed in the estimated outputs which can be felt by the pilot in a simulator.

Contrarily to the fuzzy logic method, the *linear* method is more stable when changing the measured states by the estimated states. It is a good argument because each flight test is different even if is realized at the same flight condition. The linear block absorbs the small

differences in the signals. Thus, in order to observe properly the linear accelerations of the system, the linear block seems to be more adequate because its robustness.

4.4 Model interpolation

Development of a global model that is valid over the complete operational envelope requires flight testing for different flight conditions. The common approach is to perform dynamic maneuvers at each flight condition and identify models for each configuration separately. Interpolation over the flight condition is done in order to know the system dynamics at any flight condition.

In this thesis, only the speed interpolation is performed. Thus, four types of interpolation are done: HA3000ft, HA6000ft, HF3000ft and HF6000ft.

By empirical analysis, a *cubic* interpolation is sufficient to generalize the model for each configuration.

In Chapter 3, three methods for observing the system outputs from the state variables are presented: the *classical equations* of motion, the *fuzzy logic*, and the *linear* methods. It has been shown that the classical equations of motion are not realistic enough to describe the system outputs evolution from the state variables, which is verified in Chapter 4 when the global model is set up. Then, in this chapter, we show that the fuzzy logic block is not robust enough to cope with the random effect of the environment, so that it is not used for the model interpolation. Thus, the linear block seems to be the best method to observe the linear accelerations from the estimated states.

The following figures show an example of the interpolation of the HA3000ft configuration. The pilot command is a 2-3-1-1 collective input for a level flight mission at the speed of 89 knots of speed. In Figure 1.5, all test points are shown. No model is identified around the speed of 90 knots.

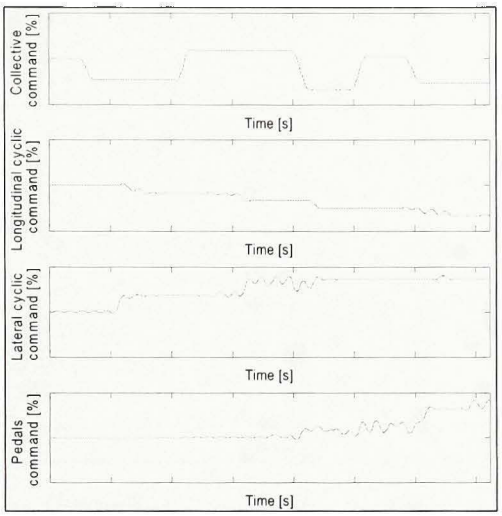

Figure 4.6 Input command used for the interpolation.

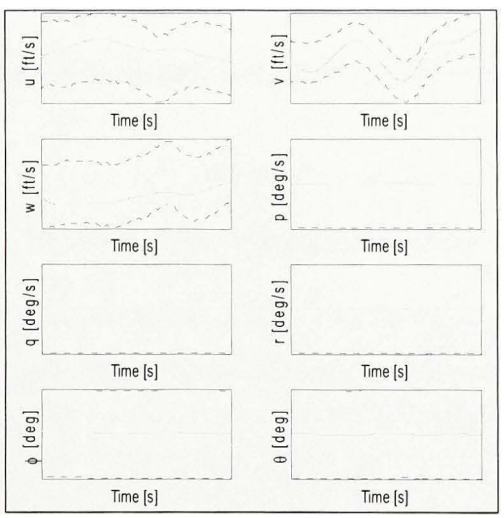

Figure 4.7 State evolution for the interpolation.

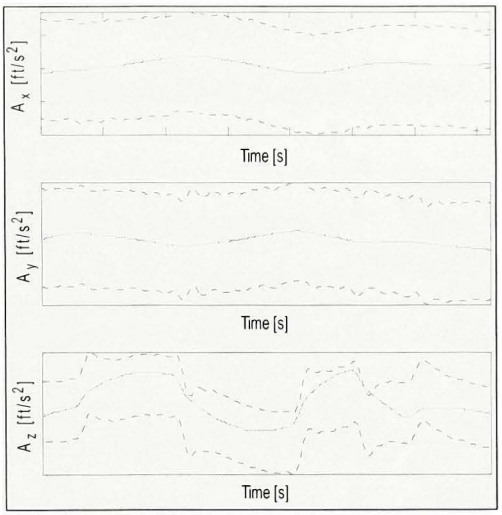

Figure 4.8 Output evolution for the interpolation.

The estimated state signals remain within the tolerance margins during all time histories.

The following table summarizes the interpolation goodness for certain flight conditions:

Table 4.7

Summary of the interpolation goodness

Flight mission	**Speed [kts]**	**Command input**	***STATES*** **Tolerance margins respect**	***OUTPUTS*** **Linear method score [%]**
HA3000ft autorot	41	2311 lat	Yes	79.57
HA3000ft autorot	41	2311 coll	Yes	69.15
HA3000ft autorot	42	2311 long	Yes	64.60
HA3000ft autorot	42	2311 ped	Yes	73.52
HA3000ft →	60	Void	Yes	84.34
HA3000ft autorot	61	2311 long	Yes	72.48
HA3000ft →	69	2311 coll	[7 / 8] 7.44s	73.31
HA3000ft autorot	87	Void	Yes	64.28

Table 4.7 (suite)

Summary of the interpolation goodness

Flight mission	**Speed [kts]**	**Command input**	***STATES*** **Tolerance margins respect**	***OUTPUTS*** **Linear method score [%]**
HA3000ft →	89	2311 long	Yes	70.69
HA3000ft →	89	2311 coll	Yes	82.81
HA3000ft →	89	Void	Yes	89.34
HA3000ft →	89	Void	Yes	82.58
HA3000ft ↑	100	2311 coll	Yes	86.14
HA6000ft →	49	2311 long	[7 / 8] 13.34s	75.86
HA6000ft ↑	49	2311 lat	Yes	79.51
HA6000ft →	50	2311 ped	Yes	91.73
HA6000ft →	50	2311 lat	Yes	93.50
HA6000ft →	50	2311 lat	Yes	88.94
HA6000ft →	50	2311 coll	Yes	73.19
HA6000ft →	50	2311 ped	Yes	82.34
HA6000ft →	50	2311 lat	Yes	87.08
HA6000ft ↑	50	2311 coll	Yes	84.51
HA6000ft →	50	2311 long	Yes	89.73
HA6000ft →	51	2311 ped	Yes	87.32
HA6000ft ↓	60	Void	Yes	83.21
HA6000ft →	60	Void	Yes	85.65
HA6000ft ↑	70	2311 lat	Yes	91.75
HA6000ft ↓	70	Void	Yes	88.23
HA6000ft →	71	2311 coll	Yes	72.76
HA6000ft ↑	89	2311 lat	[7 / 8] 7.14s	89.35
HA6000ft →	90	2311 ped	Yes	86.67
HA6000ft ↑	90	2311 lat	Yes	91.01
HF3000ft →	79	Void	Yes	82.00
HF3000ft autorot	83	Void	Yes	76.61
HF3000ft →	99	Void	Yes	81.83
HF3000ft →	102	Void	Yes	76.44
HF3000ft →	108	Void	Yes	83.87
HF6000ft ↑	58	Void	Yes	81.70

By observing the above table, it can be said that the model interpolation is validated by many flight tests not used to identify the model at the different flight condition. For only three flight tests, the estimated state signals do not remain within the tolerance margins during all time histories. Yet, the estimated states are outside the tolerance margins long afterwards three seconds. Then, the estimated states, which are outside the tolerance margins do not diverge, they remain near the tolerance margins. Thus, the state evolution for the interpolated flight conditions is defined correctly.

For the outputs observation, the linear method gives good results with a mean higher than 80% of goodness.

CONCLUSION

In this thesis, a Bell-427 helicopter model is identified for 22 flight conditions. Each flight condition is defined by an altitude (from 3,000 ft to 6,000 ft), a speed (from 30 knots to 115 knots), a helicopter loading (heavy) and a position of the helicopter CG (aft or forward). To identify the models, 2-3-1-1 multistep control inputs are performed by the pilot to excite all helicopter modes. Four types of pilot controls are used: the collective, the longitudinal cyclic, the lateral cyclic and the pedals controls. In order to identify the longitudinal and lateral motions of the helicopter, a new data set is constructed by concatenating the data related to each of the four control input.

This thesis can be split into two parts, each of them concerning a particular problem. The first one is about determining the state evolution from the pilot controls. The state variables are the linear and angular speeds and the x- and y- axis Euler angles. This problem is treated in open-loop (the estimated state signals are defined by the pilot controls and the measured state signals) and in closed-loop (the estimated state signals are defined only by the pilot controls). A recursive method is used to define the relationship between the states and the pilot controls. An optimization based on neural network theory is then applied in order to increase the goodness of the model in closed-loop. Then, a tuning of the initial state conditions is implemented in order to satisfy the FAA rules. Indeed, to guarantee the goodness of a model, the FAA has defined several rules to be satisfied by the model. The first rule is about tolerance margins the estimated signals have to remain within. These tolerance margins are defined for each variable and for each helicopter mission (level flight, ascending flight, descending flight and autorotation). The second rule is about the model validation. All identified models must be validated by at least three different flight tests not used for the models identification and the estimated signals must remain in the tolerance margins at least during three seconds. For the 22 flight conditions, the state evolution in open-loop and in closed-loop satisfies the FAA rules so that the method to generate the state equation is validated. The first conclusion of this thesis is that the dynamical behaviour of the Bell-427 helicopter would be defined by a six DOF linear model.

The second problem is about the outputs observation from the state variables. In this thesis, the outputs variables are the linear accelerations. It has been shown that the classical equations of motion do not give good results, which is explained by the fact that the theoretical equations do not cope with the random effects of the environment when data are measured. Thus, two methods are defined and a comparison between the three methods is done in order to find a powerful method to observe the system outputs from the system states: (1) a fuzzy logic method and (2) a linear method optimized with a neural network algorithm (linear NN). In order to compare the results obtained with the three methods, all blocks have the same inputs and the same outputs. Then, because the model could be implemented in a simulator for the pilot training, the pilot feedback is very useful in order to compare the reality with the results of the mathematical model. Thus, the comparison among the three methods is based on the plots particularities. A list of criteria is defined with weighted coefficients. When the outputs are obtained from the measured state variables, the linear NN and fuzzy logic methods give noticeably the same results but the linear method obtains a better score.

A clear conclusion about this second problem can be given when the global model is set up. Indeed, the global model is the most realistic and could be implemented in a simulator to reproduce the reality. The outputs must be defined by the estimated state variables and not by the measured ones. The estimated states variables depend only on the pilot controls (closed-loop study). With this model, the results are clearer. The fuzzy logic method is not robust enough so that the outputs plots shows peaks, which can be felt by the pilot. A better final score is obtained by the linear method more than the fuzzy logic method.

The last step of this thesis is to use the conclusions of both problems in order to generate the dynamical behaviour of the Bell-427 helicopter for any flight condition. For that, a cubic interpolation is done for the state and output equations. Three matrices are generated: (1) the *state matrix*, A, (2) the *input matrix*, B, and (3) the *output matrix* given by the linear NN method. The interpolation is validated by many flight tests not used to identify the models.

Thus, the methodology presented in this thesis has shown its robustness for the identification and the validation of the Bell-427 helicopter model and has yielded a global up-and-away model.

APPENDIX I

STATE EQUATION RESULTS

Table 1.1

Synthesis of all flight conditions studied

Flight condition	Studied parameters		Identification	Validation1	Validation2	Validation3
HA3000ft 30kts	Flight mission		2311 →	2311 coll →	Void →	2311 long →
	Controls duration, s		25.1	7.54	5.36	7.5
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[7 / 8] 23.36s	[7 / 8] 5.66s	[8 / 8]	[8 / 8]
HA3000ft 40kts	Flight mission		2311 →	2311 lat ↓	Void →	2311 coll ↑
	Controls duration, s		28.68	5.54	26.16	6.26
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HA3000ft 50kts	Flight mission		2311 →	2311 coll →	Void →	Void →
	Controls duration, s		23.74	5.6	23.3	8.74
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[6 / 8] 21.86s	[8 / 8]	[8 / 8]	[8 / 8]
HA3000ft 70kts	Flight mission		2311 →	2311 long ↑	Void →	2311 ped ↓
	Controls duration, s		26.76	5.66	12	5.72
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HA3000ft 100kts	Flight mission		2311 →	2311 lat ↑	Void →	Step long ↓
	Controls duration, s		27.54	5.82	22.12	4.78
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]

Table 1.1 (suite)

Synthesis of all flight conditions studied

HA3000ft 110kts	Flight mission		2311 →	Step lat →	Step coll →	Step long ↓
	Controls duration, s		24.24	6.68	6.7	4.38
	Tolerance margins	OL				
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HA6000ft 50kts	Flight mission		2311 →	2311 long →	Void ↓	2311 lat →
	Controls duration, s		27.16	5.58	3.92	6.84
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HA6000ft 70kts	Flight mission		2311 →	2311 long →	2311 ped →	2311 lat →
	Controls duration, s		25.24	7.7	5.98	8.58
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HA6000ft 80kts	Flight mission		2311 →	2311 lat ↓	2311 coll →	2311 ped ↓
	Controls duration, s		29.7	7.1	6.6	7.12
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[7 / 8] 27.14s	[8 / 8]	[8 / 8]	[8 / 8]
HA6000ft 90kts	Flight mission		2311 →	2311 coll →	2311 long ↑	2311 lat ↑
	Controls duration, s		27.92	9.1	6.42	7.82
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HF3000ft 30kts	Flight mission		2311 →	2311 long →	Void →	2311 lat →
	Controls duration, s		25.42	8	7.1	6.72
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]

Table 1.1 (suite)

Synthesis of all flight conditions studied

HF3000ft 40kts	Flight mission		2311 →	2311 lat →	Void →	2311 ped →
	Controls duration, s		28.82	7.04	7.74	7.16
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[7 / 8] 26.96s	[8 / 8]	[8 / 8]	[8 / 8]
HF3000ft 50kts	Flight mission		2311 →	2311 lat →	Void →	2311 long →
	Controls duration, s		27.4	6.02	7.36	6.14
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HF3000ft 60kts	Flight mission		2311 →	Step coll ↓	Doub. Long →	Step coll →
	Controls duration, s		25.56	3.82	4.32	4.32
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HF3000ft 70kts	Flight mission		2311 →	2311 lat →	Void →	2311 coll →
	Controls duration, s		28.68	7.08	5.62	8.42
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HF3000ft 90kts	Flight mission		2311 →	2311 coll →	Void →	2311 long →
	Controls duration, s		27.54	10.5	14.06	6.5
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HF3000ft 115kts	Flight mission		2311 →	Void →	Doub. Long →	Step lat ↓
	Controls duration, s		26.6	4.64	4.48	5.36
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HF6000ft 40kts	Flight mission		2311 →	Void →	2311 ped Autorot.	2311 lat Autorot.
	Controls duration, s		30.2	3.82	8.26	8.28
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[7 / 8] 5.42s	[8 / 8]

Table 1.1 (suite)

Synthesis of all flight conditions studied

HF6000ft 50kts	Flight mission		2311 →	2311 coll →	Void ↓	2311 long →
	Controls duration, s		32.18	7.74	6.76	10.66
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HF6000ft 70kts	Flight mission		2311 →	2311 coll →	Void ↓	2311 coll →
	Controls duration, s		32.26	9.36	4.84	10.08
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HF6000ft 80kts	Flight mission		2311 →	2311 ped Autorot.	2311 ped ↓	2311 lat ↓
	Controls duration, s		35	7.68	7.06	7.52
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
HF6000ft 90kts	Flight mission		2311 →	2311 long →	2311 lat ↑	Void →
	Controls duration, s		33.64	8.5	7.74	4.1
	Tolerance margins	OL	[8 / 8]	[8 / 8]	[8 / 8]	[8 / 8]
		CL	[8 / 8]	[8 / 8]	[7 / 8] 7.02s	[7 / 8] 6.72s

APPENDIX II

STATE AND OUTPUTS EVOLUTION IN OPEN AND CLOSED-LOOP

The states variables in open-loop for all flight conditions are shown, followed by the output equation graphs. Finally, the states and outputs evolution of the global model are shown. All these results are given on the CD.

For confidential reasons, charts dissemination is not authorized.

APPENDIX III

COMPARISON AMONG THE THREE METHODS FOR THE OUTPUT EQUATION

All Tables of the scores needed for the three methods comparison are shown in the joined CD.

APPENDIX IV

COMPARISON AMONG THE THREE METHODS FOR THE GLOBAL MODEL

All Tables of the scores needed for the three methods comparison are shown in the joined CD.

APPENDIX V

STATES AND OUTPUTS EVOLUTION FOR THE MODEL INTERPOLATION

All plots obtained for all flight conditions which are not used for the model identification and validation are shown in the joined CD.

For confidential reasons, charts dissemination is not authorized.

LIST OF REFERENCES

Bohlin, T., 2006, *Practical Grey-box Process Identification: Theory and Applications*, London: Springer-Verlag London Limited (http://dx.doi.org/10.1007/1-84628-403-1)

Chui, S., 1994, *A Cluster Estimation Method with Extension to Fuzzy Model Identification*, IEEExplore (http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00343644)

Crisan, E. G., 2005, *Validation of a mathematical model for Bell 427 helicopter using parameter estimation techniques and flight test data*, M. Eng thesis, Ecole de technologie supérieure, Montréal.

Derusso, P, Roy, R.J., Close, C.M., Desrochers, A.A, 1998, *State variables for engineers*, 2nd edition, New York, N.Y., J. Wiley and Sons.

Hagan, M., Demuth, H., Beale, M., 1996, *Neural network design*, Boston, MA: PWS Publishing.

Hamel, P. G., Kaletka J., 1997, *Advances in rotorcraft system identification*, Prog. Aerospace Sci., Vol. 33, pp. 259-284.

Jategaonkar, R. V., 2006, *Flight vehicle system identification: a time domain methodology*, Progress in Astronautics and Aeronautics, Vol. 216. Arlington: Lu, 535 p.

Milliken, W. F., 1951, *Dynamic stability and control research*, Proceedings of the third Anglo-American Aeronautical Conference", Brighton, pp. 447-524.

Popov, A., 2005, *Proof-of-match technique for Bell 427 helicopter level D simulator*, M. Ing thesis, Ecole de technologie supérieure, Montréal.

Powell, M. J. D, 1978, *A fast algorithm for nonlinearly constrained optimization calculations*, Springer Berlin / Heidelberg, Vol. 630, pp.144-157.

Prouty, R.W., 2002, *Helicopter performance, stability and control*, Malabar, Flor., R. E. Krieger.

Takagi, T., Sugeno, M., 1985, *Fuzzy identification of systems and its applications to modeling and control*, IEEE Transactions on Systems, Man and Cybernetics, Vol.15, no.1, pp. 116-132.

Tischler, M., B., Remple, R. K.. 2006, *Aircraft and rotorcraft system identification: Engineering methods with flight test examples*, AIAA Education Series, Virginia: Schetz, 523 p.

Tischler, M. B., Kaletka, J., 1987, *Modeling XV-15 tilt-rotor aircraft dynamics by frequency and time-domain identification techniques*, AGARD: Rotorcraft Design for Operations, 20 p.