Data Analysis In Flight Testing

A.K. Karwal, J.A. Mulder, H. Muhammad
Delft University of Technology, The Netherlands

Data analysis and model identification play an ever increasing important role in the design and execution of flight test programs. In the experiment, however, huge amounts of digital data is posing the flight test engineer with serious data management problems. In the data analysis phase a good user-interface to the identification software is essential in order to enable the flight test engineer to intelligently monitor algorithm performance and -results. The present paper discusses the design and implementaion of a user-friendly modular program package 'Interactive Flight Data Analysis' (IFDA) covering all phases of flight test. IFDA takes advantage of the computing power and graphical capabilities of a workstation, enabling the flight test engineer to do all analysis behind-the-desk rather than resorting to mainframe operations. The program package can also be used as a research tool as it makes an easy comparison of different identification algorithms possible. In addition simulation experiments of a-priori models enable the flight test engineer to optimize experiment design with respect to costs and information contents. Examples will be given where the modular program package has been used in successfully managing computational burdens: an aerodynamic model identification of a Jetstream Series 100 of Cranfield Institute of Technology and a turbo propeller thrust identification of a Swearingen Metro II of the Dutch National Aerospace Laboratory.

Introduction

Dynamic flight test technology is now acknowledged as a valuable tool for the assessment of performance- and stability and control characteristics. A number of system- and parameter estimation techniques can be applied to the problem of estimating the aircraft trajectory, instrument errors and aerodynamic model parameters from onboard measurements in non-steady flight. All these techniques have in common that -using modern data logging equipment- processing of large amounts of digital data is involved, posing the flight test engineer with data management problems. In the data analysis phase a good user-interface to the identification software combined with high-quality graphical output is essential for monitoring algorithm performance and intelligent judging of the results.
In this paper the design of a user-friendly modular program package 'Interactive Flight Data Analysis' (IFDA) covering all phases of flight test data analysis is discussed. The program package takes advantage of the computing power and graphical capabilities of a workstation, enabling the flight test engineer to do all analysis behind-the-desk rather than having to resort to cumbersome mainframe operations. In the paper the different phases of flight test data analysis is discussed and an overview is given of the identification algorithms incorporated in the the program package. Finally two applications will be given where the program package has been successfully used in managing computational burdens.

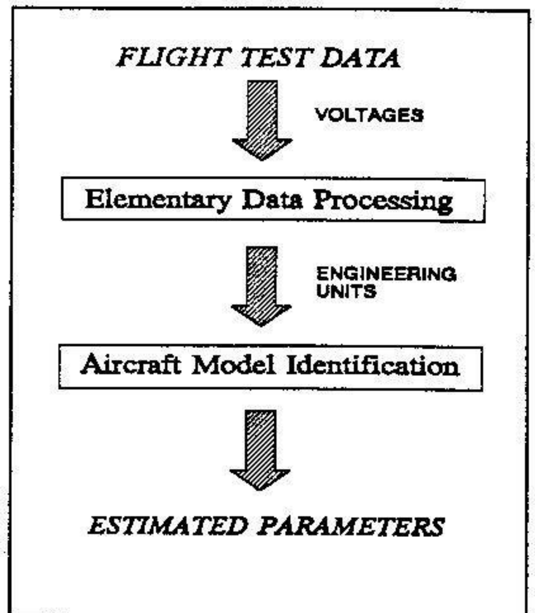

Figure 1 : Phases in flight test data analysis.

Elementary Data Processing

In the analysis of flight test data two distinctive phases may be discerned, see Figure 1. The first step of data reduction, i.e. the 'translation' of the recorded voltages into physical values to be used in the identification process is refered to as 'Elementary Data Processing'. The output of the Elementary Data Processing serves as a data base for the Model Identification phase where identification techniques are applied to develope the mathematical models used for calculation of the desired performance- and stability characteristics. Accurate flight test data can be obtained by using high accuracy instrumentation[1]. In general, high quality sensors generate accurate analog DC voltage output signals that are passed through low-pass filters, fed to a multiplexer, serially converted to digital readings by an Analog-to-Digital Converter (ADC) and recorded on disk or tape together with some administrative data, see Figure 2.
Elementary Data Processing comprises several steps[2]: