DEVELOPMENT OF A FLEXIBLE FLIGHT TEST INSTRUMENTATION SYSTEM

ing. A.Muis, ir. J.Oliveira, prof. J.A. Mulder

Delft University of Technology, Faculty of Aerospace Engineering,
Aerospace Design, Integration and Operations Group,
Control and Simulation Division
1059 CM Delft, The Netherlands
E-mail: a.muis@tudelft.nl

Abstract: This article poses some questions about what should be expected from an instrumentation system and evolves to a description of the actual flight test instrumentation system developed for a Cessna Citation 550. Reasons are presented which support the statement that the main concern in building an instrumentation system is to achieve an easy maintainable system with flexible configuration of sensors.

Keywords: Instrumentation, data acquisition

1 INTRODUCTION

The Faculty of Aerospace Engineering of the Technical University Delft in The Netherlands maintains a Cessna Citation 550 aircraft in cooperation with the National Aerospace Laboratory (NLR). This aircraft is used for several experiments and student flights where students experience basic flight dynamic manoeuvres of the aircraft.

Figure 1: Cessna Citation 550

The Cessna Citation 550 is a small business jet aircraft, which is instrumented to serve as a flying classroom and flight research aircraft. The instrumentation system, including sensors, transducers, filters and a dSPACE computer is called the Flight Test Instrumentation System (FTIS) [1, 2, 3, and 6]. This instrumented flying platform serves multiple investigation pur-

poses: the multitude of applications for which the instrumentation system was initially thought, require it to be very flexible, very accurate and very comprehensive in terms of the sensorial data being logged.
The FTIS also serves as a flying classroom. The students of the Aerospace faculty take part on a demonstration flight as part of a practical assignment. Each seat is equipped with an observation screen. The students record the data that is displayed on the screens to compile the results of their reports. For these assignments the FTIS must be able to communicate with multiple observation stations to deliver the data.

Figure 2: Flying classroom

It also serves as a Flight Test Identification platform, providing highly accurate flight test data for aerodynamic model identification purposes. The field of research includes both online and offline data processing. Online data processing requires fast computing power and the offline data processing requires large data storage capability for logging data. The FTIS must be able to satisfy both requirements.
The instrumentation system is currently being developed into a platform that will give the aircraft fly by wire capability. This capability is expected to be functional sometime during 2007. In order to achieve the fly by wire requirements, the FTIS must be equipped with highly accurate sensors, perform online computations and deliver data input signals that establish the desired deflections of the control surfaces.

2 OVERVIEW

In order to cope with the new challenges set by the fly by wire requirements, the existing FTIS was evaluated upon its interaction with third party experiments and the student flights. A series of interviews were carried out among the users that interact with the FTIS or that work with the data that it provides. Based on the collected information, a list of requirements was gathered which should be satisfied in a second generation of the FTIS. These conclusions are presented in section 3. The system hardware and software were updated. This enabled a new programming environment which allowed an easy and fast development of the features required by the users. Section 4 describes the first generation hardware and software setup. Section 5 describes the new environment provided by the second generation hardware and software setup. The results obtained with the new FTIS are discussed in section 6 and the conclusions are presented in section 7.

3 USER'S POINT OF VIEW

According to its users, the FTIS should be mainly 1) a data acquisition system (sampling rates >100 Hz) with 2) an accurate Flight Path Reconstruction (FPR) module and 3) provided with a protocol that allows bi-directional communication with other working stations, allowing the FTIS to deliver data in real-time. A working station should include real-time data visualization tools and should be designed to allow easy implementation of algorithms based on real-time data processing. This will allow several simultaneous projects to be running together in real-time on the same flight, each running on an independent station communicating with the FTIS.
Several applications are already in mind: innovative FPR methods and models [3], Attitude and Heading Reference Systems, Aerodynamic Model Identification [10], Human Factors experiments, Control Input generation and advanced reconfigurable flight control systems ([7], [8] and [9]). To reinforce the use of DUECA (Delft University Environment for Communication and Activation [4] as a standard software package used inside the Control and Simulation Division, FTIS should be designed to optimally communicate with DUECA. The computer stations should use DUECA to communicate with the FTIS. Nevertheless, this should not be a limiting factor for the FTIS, i.e. other software packages should be able to connect to FTIS and retrieve data from it, such as Matlab working on a real-time environment.
The technicians agreed the FTIS should have a flexible architecture to allow easy and fast system updates. It should have a good documentation base for each sensor and a well defined protocol for storing data, to be used as a standard inside the Control and Simulation Division. This would allow the data from several facilities to be readily available for any project that might require it for some purpose. There should be propagation of sensor and system information to the logged data in the form of an ASCII header concatenated to the logged data. The files should contain ASCII information inside them explaining the contents of the file, so that an unacquainted user may still be able to use it. Additionally, software modules should be developed to help the technicians in solving hardware problems, specifically designed to run in any station.
The management staff focused on two factors, 1) a (software) working environment common or at least fully compliant with the other projects running inside the Control and Simulation Division (standardization), which should 2) allow a small team of technicians to assure the correct maintenance of all these systems (affordability).

4 FIRST GENERATION HARDWARE AND SOFTWARE SETUP

The processing core of the first generation Flight Test Instrumentation System (FTIS) consists of the combination of two dSPACE processor boards. One of these boards is connected to the dSPACE data acquisition boards by a high speed bus and is designed to handle a large amount of input/output (I/O) signals at high sampling rates. The other processor board is used to run Flight Path Reconstruction algorithms and processing data at a high rate. The system houses analog, digital, serial and synchro boards which are connected to several sensors placed inside the aircraft. These boards acquire information from the air data computer, flight management system (FMS), flight controls, engine parameters, gps receiver, an inertial measurement unit, angle-of-attack and angle-of-sideslip vanes and some other quantities of interest. Adjacent to the processor boards is an Intel processor board running a in-house developed program Codex [5] which can access the memory space of the processor boards through the ISA bus.

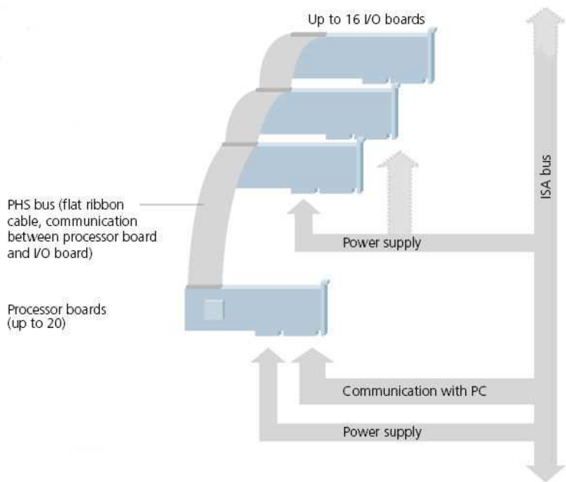

Figure 3: Hardware layout dSPACE boards

The Codex is scheduling the activation of tasks for data acquisition and publishing the information on the Intel network card. It enables the creation of a flexible data acquisition system by using object-oriented programming. Every sensor, transducer, filter or publisher is programmed into a class which is controlled by the scheduler routine. The use of object-oriented programming helps the user or developer to easily adapt a small section of the data acquisition system and therefore it's recommended to use this structure inside a flexible environment. However, creating a home made program requires a high level of programming skills and the effort to maintain the whole program well documented. These two factors are the main disadvantages of the Codex. As time evolved and the new users came into contact with the system, it showed to require a user with good programming skills and it failed to provide a good documentation, which acted as a brake in understanding the complex structure of the total program. Moreover, to overcome the complexity of updating the Codex program, third party projects frequently opted instead to modify their systems already tested while running stand-alone. This made the software development a slow, expensive, error-prone and cumbersome task.

5 SECOND GENERATION HARDWARE AND SOFTWARE SETUP

Following the completion of the first generation FTIS, based on in-house developed software, a series of projects were conducted and valuable knowledge was acquired through time and experience in dealing with the system. This knowledge led to a new generation of FTIS. The overview of the first generation system has shown that use of an object-oriented environment is recommended when developing a flexible system. However, the custom made software used in combination with third party projects did not work as expected. Therefore, a new environment has been created which continues to benefit from the object-oriented programming flexibility and also interacts better with the users.

Hardware

The Flight Test Instrumentation computer is a PowerPC 750GX (1GHz). This processor is connected to several data acquisition boards by a high speed bus and is designed to handle a large amount of I/O signals at high sampling rates (the system has been tested successfully running at 1kHz sampling rate). At this moment the system houses analog, digital, serial, arinc, csdb and synchro boards which are connected to several sensors placed inside the aircraft. These sensors acquire information from the air data computer, flight management system (FMS), flight controls, engine parameters, gps receiver and an inertial measurement unit. The information gathered from the sensors can be stored as raw data or transformed into meaningful values that can be stored on disk or published to a client listening on the Ethernet network. The update rate of the data from each sensor is customizable. Sensors can be interrupt handled or sampled via the polling mode. It requires little effort to program an I/O board, such as the serial board, to publish data to a client instead of using the Ethernet network.

Software

Most of the software developed for the flight test instrumentation system is created with the use of Matlab Simulink. dSPACE provides the user with different libraries which can be installed in Simulink and therefore simplify the creation of the application. Data acquisition boards are displayed as blocks which can interact with the sensors. These blocks can be connected to other standard Simulink blocks.

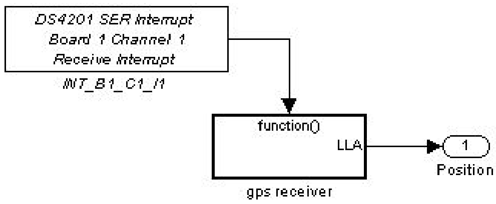

Figure 4: Serial interrupt connection

Figure 4 shows an interrupt block that is connected to a subsystem which holds the routines to compute the gps position gathered from the serial board. The interrupt block is creating a function call that is activating the sub-system. Inside the sub-system the developer places a block that retrieves data from the serial board and runs the data through a function which will transform the raw data into meaningful values. The function inside the sub-system has a direct relation with the sensor connected to the data acquisition block and therefore a sort of object-oriented environment is created. Several of these blocks can be created inside a Simulink model. Each block will be triggered by the scheduler according to its sampling mode (interrupt handled or polling mode).

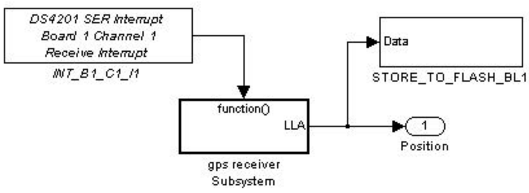

Figure 5: Data storage

The data is stored on a memory space in a sample-and-hold fashion. Every sensor block tags the data with the time of the system internal clock immediately after receiving it. Known time delays are accounted for. Figure 5 displays a layout with a connection to the flash memory space of the processor board. Another possibility is to send this data through an Ethernet network, a feature which can be programmed as easily as the flash-memory block.

The new FTIS is more than a data acquisition system. Using standard Matlab Simulink blocks, computationally heavy data processing routines can be implemented in the software. Figure 6 shows an example of a general Kalman filter implementation. The whole model represented here is a single sub-system which can be dragged and dropped like a regular Simulink block. The user is merely required to adapt a series of S-functions and declare a few matrices which are directly related to the driving equations of the state vector and to the measurement equations.

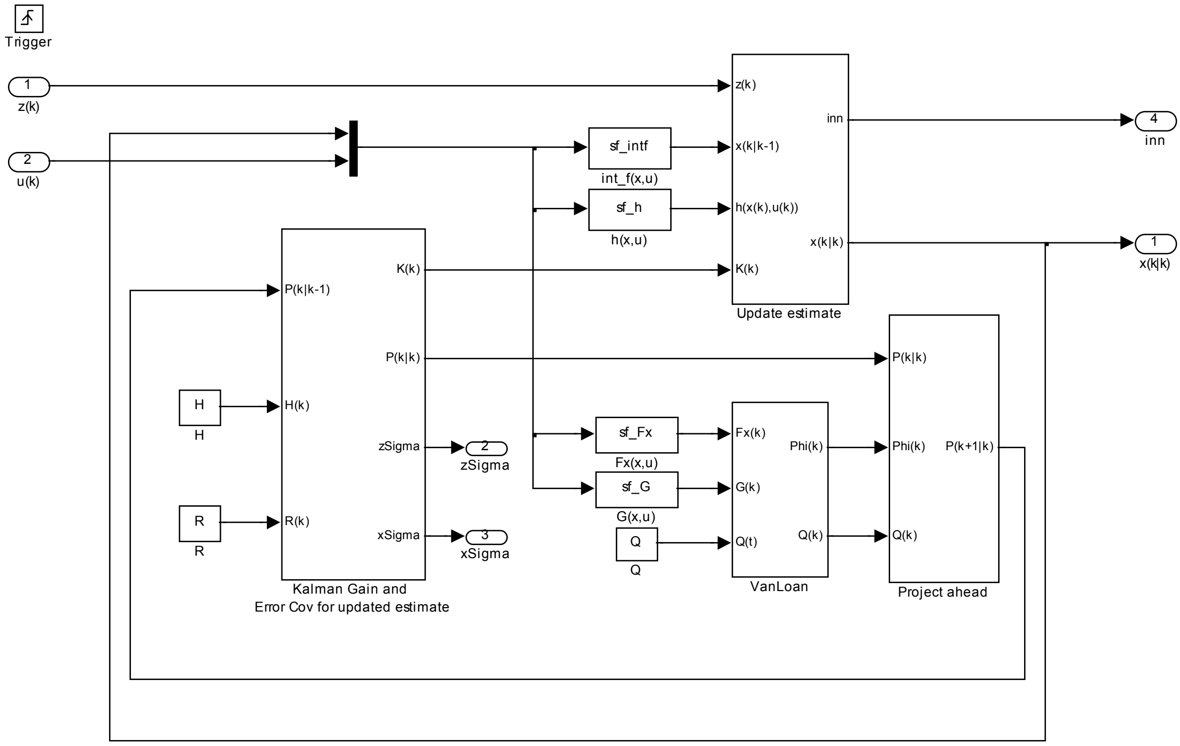

Figure 6

6 RESULTS

To test the flexibility of the data acquisition system two system setups were build. One setup was created to fulfil the requirements of the student flights where students need to observe in real-time the manoeuvres of the airplane and log the data to include in their reports. The second setup consisted of an interaction with a third party data acquisition system. The objective was to verify the quality of an attitude and heading reference system (AHRS).

Student flight setup

The student flight setup was the first setup using DUECA for the data communication and using the Matlab/Simulink environment to acquire the information from the sensors.

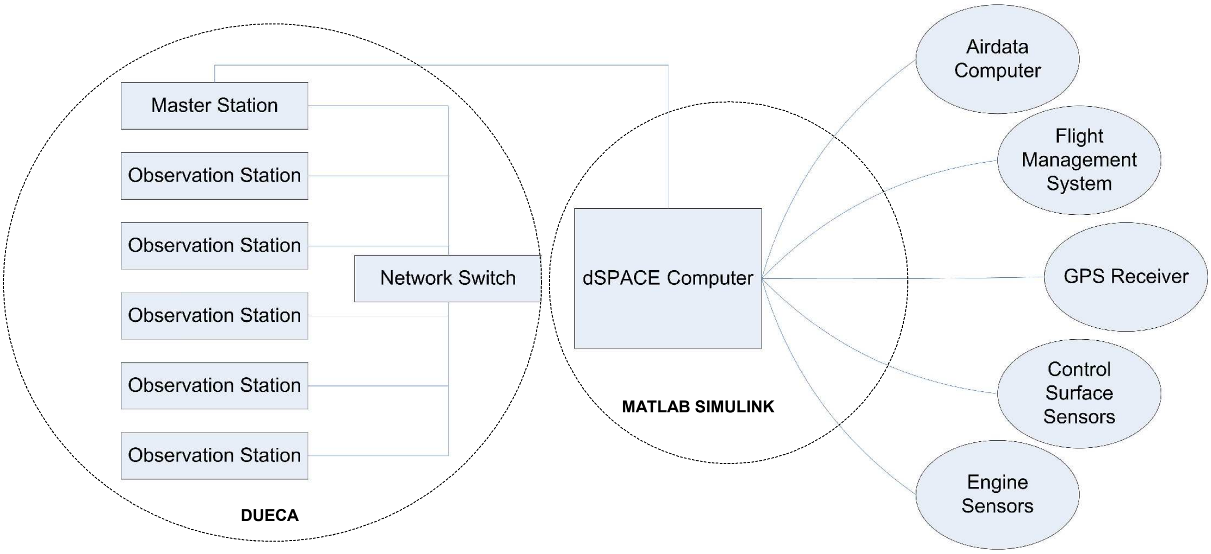

Figure 7: Flying classroom schematic

The FTIS broadcasted the data to a master station via the Ethernet network using UDP packets. The master station had two Ethernet cards. The second Ethernet network was controlled by DUECA and connected the master station to the other observation stations. The data was logged on by the master station. Figure 8 shows a plot of a dutch roll manoeuvre. The students were watching this kind of plots being drawn in real-time while experiencing the manoeuvre.

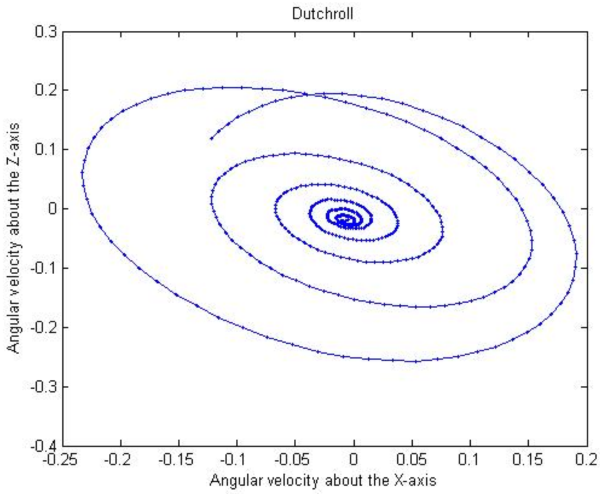

Figure 8: Dutchroll

Third party experiment setup

This setup was made for a company which wanted to verify the accuracy of their own attitude and heading reference systems (AHRS) in a dynamic environment. To accomplish this, the FTIS was connected via the ARINC board to an Inertial Reference System (IRS) unit, which provided a reference for the attitude and heading measurements. All standard sensors belonging to the FTIS remaining connected to the system and only a simple adaptation was needed to update the software inside the Matlab Simulink environment. An indication of the system's flexibility was demonstrated by the fact that the IRS was installed and fully readable in less than a week. In this setup the DUECA environment was not used. The third party data acquisition system was listening to the FTIS like if it was another sensor, as shown in figure 9.

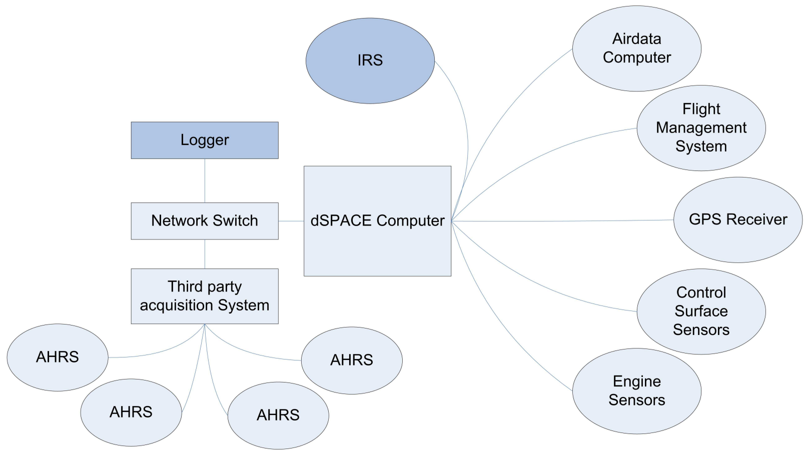

Figure 9: Third party schematic

Figure 10 shows a comparison plot between the attitude states measured by one AHRS and the IRS reference. The plots in the first row show a series of roll manoeuvres, while the plots in the second row refer to the pitch angle during a parabola manoeuvre. The residual plots on the right show the deviation of the AHRS measurements from the IRS reference. These plots were created after the flight. However, the flight engineer aboard the flight had access to similar plots in real time.

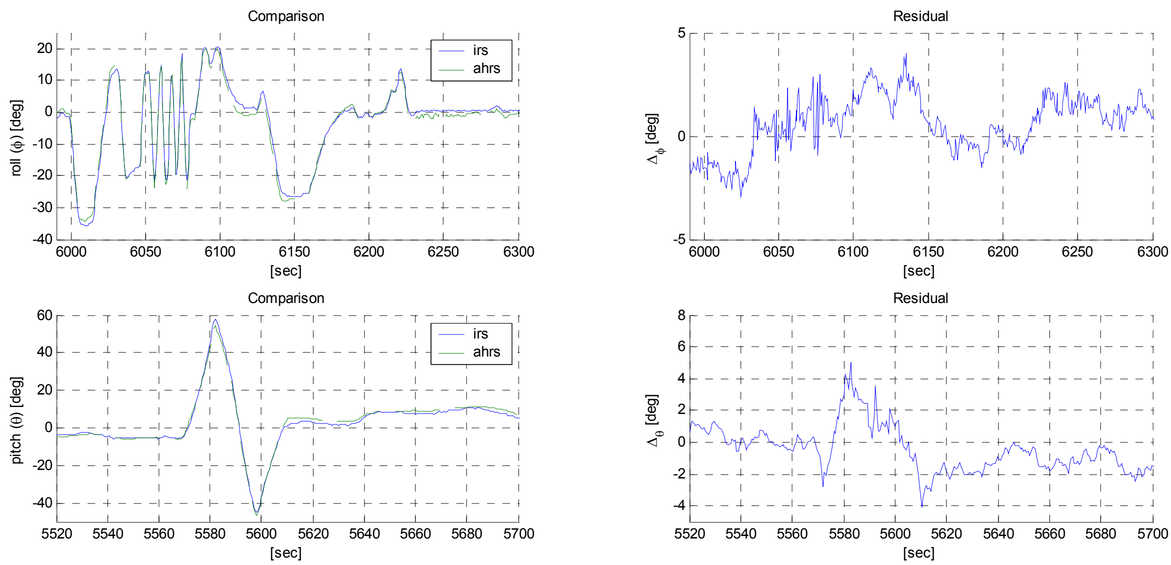

Figure 10: Comparison between the attitude states measured by one AHRS and the IRS reference

7 CONCLUSIONS

The results and the experience obtained from the different setups have shown that the system is flexible enough to fulfil the requirements of a variety of applications and users. Additional sensors can be connected to the system and the software is easily adaptable by any user who is familiar with the Matlab Simulink environment. The integration with the DUECA environment allows users to test their projects (software modules) in different facilities inside the Control and Simulation Division. As an example, projects can be tested, debugged and improved on the flight simulator before performing the actual flight tests. The communication with third party systems not running the DUECA environment is also possible. The FTIS software can be easily adapted according to the project requirements. To establish a data link with a third party system, the FTIS can accomplish this via: Ethernet, serial, digital and analog communication. The new FTIS integrated with DUECA now allows for the fly-by-wire to be implemented.

8 FUTURE DEVELOPMENTS

The future developments of the FTIS system will focus on the data logging. The current configuration does not perform data logging. While a particular project may require no more than the data from a particular sensor, there is no reason to discard the rest of the data being acquired. Therefore, future developments will connect the FTIS to an independent data logging system that will store all the data gathered from the FTIS through the use of a separate link. This will separate the logging of data from the different layouts of each project. The data logger will have direct access to the system ISA bus, allowing communication speeds higher than those achieved through the Ethernet network. This solution will allow data logging at rates higher than 1kHz without affecting the bandwidth of the available data link protocols which are used to communicate with the other stations. The data logger will use a standard for storing data which includes an ASCII header containing relevant information about the data and the instructions on how to read the file, allowing future projects to easily use the data from flight tests carried out in the past.

9 REFERENCES

[1] M.W.Soijer. Data acquisition and analysis software as real-time processes. 31st Annual Symposium Society of Flight Test Engineers, 2000

[2] G.W.Milne, J.A.Mulder, M.W.Soijer, S.Juliana, and Hermansyah. Maximum likelihood stability & control derivative identification of a Cessna Citation II. AIAA 2001-4031, August 6-9 2001. Montreal, Canada.

[3] Oliveira J., Mulder J.A., Chu G.P. and Vos W.G.M. Developing a navigation system for aerodynamic model identification using two-step method. EUCASS European Conference on Aerospace Sciences, Moscow, july 2005.

[4] M.M.Paassen, Olaf Stroosma and J.Delatour. DUECA – Data-Driven Activation in Distributed Real-Time Computation, AIAA-2000-4503, AIAA Modelling and Simulation Technologies Conference, Denver, CO, Aug. 14-17, 2000.

[5] M.W.Soijer. Software-Enabled Modular Instrumentation Systems. ISBN: 90-9017445-1

[6] http://www.dspaceinc.com/ww/en/inc/home.cfm valid URL on 15 February 2006.

[7] L. Sonneveldt, Q.P. Chu, J.A. Mulder, Constrained adaptive backstepping flight control: application to a nonlinear F-16/MATV model, accepted to AIAA Guidance, Navigation, and Control Conference and Exhibit in August 2006, Colorado.

[8] T.J.J. Lombaerts, D.A. Joosten†, J. Breeman, T.J.J. van den Boom, Q.P. Chu, J.A. Mulder, M. Verhaegen, Assessment criteria as specifications for reconfiguring control, accepted to AIAA Guidance, Navigation, and Control Conference and Exhibit in August 2006, Colorado.

[9] E.R. van Oort, Q.P. Chu and, J.A. Mulder, Robust Model Predictive Control of a Feedback Linearized Nonlinear F-16/MATV Aircraft Model, accepted to AIAA Guidance, Navigation, and Control Conference and Exhibit in August 2006, Colorado.

[10] J. Oliveira, Q.P. Chu, J.A. Mulder, H.M.N.K. Balini and W.G.M. Vos, Output Error Method and Two Step Method for Aerodynamic Model Identification, AIAA Guidance, Navigation, and Control Conference and Exhibit, 15-18 August 2005, San Francisco, California.