FAULT TOLERANT AIR DATA SYSTEM FOR PITOT TUBE FAILURE

OMAR HAZBON ALVAREZ

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
DOCTORADO EN INGENIERÍA
MEDELÍN
2020

FAULT TOLERANT AIR DATA SYSTEM

FOR PITOT TUBE FAILURE

OMAR HAZBON ALVAREZ

A Dissertation submitted to
the School of Engineering in partial fulfilment of the requirements
for the degree of Ph.D. in Engineering (Doctor en ingeniería)

Supervisor(s)

Luis Benigno Gutiérrez Zea, Ph.D.
Marcello Rosario Napolitano, Ph.D.
Cornelis Bil, Ph.D.

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
DOCTORADO EN INGENIERÍA
MEDELÍN
2020

Noviembre 30, 2020

Omar Hazbon Alvarez

“Declaro que este trabajo de grado no ha sido presentado con anterioridad para optar a un título, ya sea en igual forma o con variaciones, en esta o en cualquiera otra universidad”.
Art. 92, parágrafo, Régimen Estudiantil de Formación Avanzada.

Firma: Omar Hazbon Alvarez.
_____

To my wife Angelica for her constant comprehension and support and my dad who made possible to materialize my dreams.

ACKNOWLEDGEMENTS

I would like to thank Dr. Luís Benigno Gutiérrez, my advisor, for his encouragement and support during my PhD studies and professional life at Universidad Pontificia Bolivariana. I would like to thank Dr. Marcello Rosario Napolitano for introducing me to his field of research and suggesting the research topic on the aircraft airspeed sensor failure detection, identification and accommodation, his research material and constant advice was fundamental for my research. Special thanks to Dr. Cees Bil for his constant mentoring in my professional life and his invaluable support for this thesis, especially during the research internship at RMIT University. Finally, I would like to acknowledge the support of Colciencias as sponsor institution and to the Universidad Pontificia Bolivariana, institutions which made possible to materialize the Ph.D.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

An integrated airspeed and angle of attack sensor failure detection, identification, isolation and accommodation scheme is presented. The proposed system uses information from the inertial measurement unit, available air data sensors, and an aircraft digital twin that provides virtual measurements of the aircraft's aerodynamic and propulsion forces to feed a nonlinear estimator capable of detecting air sensor failure and suppress its effect on the aircraft air data prediction. The novelty of the proposed approach is that sensor fault detection, identification, isolation, and accommodation are integrated into a feedback scheme where the information produced by fault detection is used to modulate the noise covariance of faulty sensors so that the nonlinear estimator is able to maintain the air data estimate with a small error despite the presence of various failures in the air data sensors.

The system was developed and tested in simulation. A Matlab/Simulink Ryan Navion aircraft simulation model was developed using flight test and wind tunnel data from Princeton University Flight Research Laboratory. Matlab/Simulink sensor models where developed using actual measured sensor data. A Dryden wind turbulence model was used to test the system against atmospheric perturbations. Flight simulations included climb, cruise, turns and descent maneuvers.

Independent and joint Pitot tube and angle of attack vane sensor failures where simulated. Simulation results showed that the fault tolerant estimation air data scheme is very accurate and eliminates undetected or false alarm failures.

KEYWORDS: EXTENDED KALMAN FILTER; AIR DATA SYSTEM; PITOT TUBE FAILURE; ANGE OF ATTACK VANE FAILURE, AIR DATA SYSTEM FAILURE; UNRELIABLE AIRSPEED; COVARIANCE MODULATION; AIRCRAFT DYNAMIC MODEL; AIRCRAFT SIMULATION; SENSOR FAILURE DETECTION; SENSOR FAILURE ISOLATION; SENSOR FAILURE IDENTIFICATION; SENSOR FAILURE ACCOMODATION; NAVIGATION SYSTEM, AIRCRAFT EQUATIONS OF MOTION; VIRTUAL SENSOR, TAKAGI-SUGENO; FUZZY INFERENCE SYSTEM; NEURO-FUZZY NETWORK; LOW PASS FILTER; SIGNAL PROCESSING; AVIONICS; ANGLE OF SIDESLIP VANE; SYNTHETIC AIR DATA SYSTEM; ADS; IMU; SENSOR SIMULATION MODELS; PROPULSION SYSTEM MODEL; AIRCRAFT DYNAMICS; FLIGHT TESTING; AIR DATA ESTIMATOR; PITOT TUBE ICING; ADS SIMULTION MODEL; AIRCRAFT DIGITAL TWIN.

CHAPTER 1. INTRODUCTION

The air data system (ADS) plays a vital role in the aircraft operation. The information provided by this device is used by the pilot and other aircraft subsystems for maneuvering and navigating between some safe performance boundaries. There is a history of aircraft accidents caused by the failure of one ADS sensor, the Pitot tube, for example Air France flight 447 in June 2009 and Saratov Airlines flight 6W703. However recent air accidents like Lion Air flight 610 in October 29 2018 and Ethiopian airlines flight 302 in March 10 2019 were attributed to a malfunction of an important component of the ADS, the angle of attack vane sensor. Table 1 shows a brief description of aircraft accidents caused by the ADS sensors failure since 1973.

Table 1. Civil aircraft accidents caused by ADS sensor failure.

Date	Model	Damage	Cause	Casualties	Source
01/30/1973	DC-9	Total loss	Ice in Pitot tubes	0	[1]
12/01/1974	Boeing 727	Total loss	Ice in Pitot tubes	3	[2]
07/28/1984	Learjet 25B	Total loss	Pitot tube covers	0	[3]
03/21/1986	Tupolev 154B	Total loss	Ice on Pitot tubes	0	[4]
03/02/1994	MD-82	Total loss	Ice on Pitot tubes	0	[5]
02/06/1996	Boeing 757	Total loss	Debris Pitot tubes	189	[6]
10/02/1996	Boeing 757	Total loss	Tape static port	70	[7]
10/10/1997	DC-9-32	Total loss	Ice on Pitot tubes	74	[8]
04/7/1999	Boeing 737	Total loss	Ice on Pitot tubes	6	[9]
10/17/1999	MD-11F	Total loss	Ice on Pitot drain	0	[10]
06/3/2006	Dornier 328	Total loss	Obstruction Pitot tube	0	[11]
02/25/2009	Boeing 737	Total loss	Radio Altimeter failure	9	[12]
06/01/2009	Airbus A330	Total loss	Obstruction Pitot tube	228	[13]
02/11/2018	Antonov 148	Total loss	Ice on Pitot Tubes	71	[14]
10/29/2018	Boeing 737	Total loss	Angle of attack vane	189	[15]

The Pitot tube sensor is responsible for the airspeed estimation. Under certain atmospheric conditions the sensor becomes covered or blocked with ice, dirt or even ground protection devices and, as a result, the computed airspeed becomes erratic or unreliable. The sensor that measures the aircraft's angle of attack, the angle of attack vane, can fail in a similar way as the Pitot tube. Pilots may not be able to identify the failure and become confused due to unreliable and conflicting warnings.

Aircraft manufacturers have implemented sensor redundancy with a voting scheme to detect and isolate faulty air data sensor signals. However, under certain atmospheric circumstances all sensors might fail at the same time, which is called a "common mode failure". Common mode failures can be caused by very cold and humid atmospheric air where icing can block the dynamic and static orifices and drain holes. Water can also be accumulated in probe or pressure lines when drain hole is obstructed by ice or a foreign object. Airbus equipped the A320, A330 and A340 with a Backup Speed Scale (BUSS) [16]; this system is based on the use of a theoretical airspeed which is estimated from pitch and thrust tables, however the system failed on Air France 447 [17]. Airbus is planning to incorporate an extra airspeed estimation sensor based on engine nacelle pressure on the A350 [18], [19]. On the other hand, Boeing's approach to this problem was to equip the 787 Dreamliner with an airspeed estimation calculated from angle of attack and inertial data which they call "Synthetic Air Speed"; however even this system has been known to fail on Jetstar flight JQ 07 [20]. Apart from sensor redundancy no backup system has been developed and incorporated by aircraft manufacturers in the case of an angle of attack vane sensor failure.

This research presents a solution to the ADS sensor failure by the development of an integrated airspeed and angle of attack sensor failure detection, identification and accommodation (SFDIA) scheme. The proposed system uses information from the inertial measurement unit, available air data sensors, and an aircraft digital twin which provides information of the aircraft's aerodynamic and propulsion forces to feed a nonlinear estimator capable of detecting the airspeed and angle of attack sensor failure to suppress its effect on the aircraft state prediction. The system is capable of providing the control system and the pilot with reliable data of airspeed, angle of attack, angle of sideslip and Euler angles after losing airspeed and angle of attack sensor measurements.

The system was developed and tested in simulation. A Matlab/Simulink Ryan Navion aircraft simulation model was developed using flight test and wind tunnel dynamic data from Princeton University Flight Research Laboratory and other verified sources. Matlab/Simulink sensor models were developed from actual sensor measurements. A Dryden wind turbulence model was used to test the fault tolerant estimation system against atmospheric perturbations. Aircraft flight simulations included cruise, climb, turns, descent, acceleration and deceleration maneuvers.

1.1 Objectives

- General Objective:
    - To estimate the aircraft airspeed analytically by means of a model-based virtual sensor so unreliable Pitot tube airspeed signals can be detected and replaced by a thrusted estimate.

- Specific Objectives:
    - To develop an aircraft air data estimator algorithm capable of modulating the sensors covariance automatically with the aim of reducing the effect of sensor failures in the estimation accuracy.
    - To assess the effect of an aircraft digital twin that will act as a virtual sensor of aerodynamic and propulsion forces to help in the aircraft air data estimation during the Pitot tube and angle of attack sensor failures.
    - To develop and assess a sensor failure detection system that will feedback the aircraft air data estimator so the covariance modulation can take place.
    - To assess if the aircraft air data estimator tolerant to Pitot sensor failure works accurately at every point in the flight envelope assisting the pilot to land safely.

1.2 Research Questions

The following research questions were addressed in this work:

- How can a Pitot sensor failure lead to unsafe flight conditions and accidents?
- How can an aircraft digital twin concept detect instrument failure and assist the flight crew in maintaining situational awareness and safe flight?
- Is it possible to automatically obtain a reliable estimation of the air data system variables based on the information of all the available sensors, including those that work nominally and those that do not work correctly, by using the aircraft dynamic behavior model as a virtual sensor which will be based on previous and current information from the actual sensors?

- Can this air data estimation be reliable at every single point in the aircraft flight envelope?

1.3 Methodology

To come up with a solution to the problems caused by the Pitot and angle of attack vane failure in the accuracy of the computed variables in the ADS, it was necessary to setup a flight simulation in which a particular aircraft and its sensor models flew through a predetermined trajectory which included the common maneuvers encountered during the operation of a commercial or general aviation aircraft. The aircraft flight simulation model was constructed from Princeton university flight research laboratory Ryan Navion experimental aircraft public domain documents that contains aircraft geometry, mass, propulsion system, performance properties and dynamic data measured in flight test and wind tunnel. The sensors statistical characteristics used for the development of the simulation models were measured from actual sensor test. Matlab® and Simulink® software was used as the development tool and simulation environment.

The sensor models have the capability of generating two possible type of sensor faults, the stuck and additive failures, to replicate the behavior of the unhealthy Pitot signals. Stuck failures model obstruction of the dynamic pressure ports of the Pitot tube. The additive failure models electronic hardware failure.

The aircraft simulation model generated true state data signals that were converted in to measurement signals by the addition of bias and noise in the sensor simulation models. In addition to the Pitot tube, angle of attack vane, angle of sideslip vane sensor models, it was necessary to model additional sensors that allowed a navigation estimator to compute

the aircraft acceleration, angular rates, position and attitude without using the Pitot sensor. The navigation estimator outputs allowed the computation of aerodynamic and propulsion forces by means of the force equation of motion. The necessary additional sensors used were the inertial measurement unit (IMU) containing a three-axis accelerometer and a gyroscope, a magnetometer and a global positioning system (GPS).

A second estimator performed the task of computing the airspeed, angle of attack and angle of sideslip estimated signals. To compute the residual signals used for failure detection, the measurement airspeed, angle of attack and angle of sideslip signals were subtracted from the estimated airspeed, angle of attack and angle of sideslip signals respectively. The residual signal statistics were used to calculate thresholds that defines a limit for a sensor signal to be healthy.

In the case of a sensor failure detection a flag will be sent to the second estimator to modulate its sensor covariance so the damaged sensor measurements influence the estimation in less extent. To help in the accuracy of the estimation process during a sensor failure, a gray box model of aerodynamic and propulsion forces was developed to air in the accuracy of the air data estimation.

The performance of the system was measured by means of the estimation root mean squared error (RMSE), the time to detect the faults and the number of undetected or false sensor failures. The air data estimation and fault detection scheme were extended successfully the angle of attack vane failure. Joint Pitot tube and angle of attack failures were simulated to assess the accuracy of the air data estimation in an extreme case of an

ADS failure. The air data estimation system was able to restore itself after sensor failures end.

1.4 Thesis Contribution

A successful fault tolerant model-based air data estimator was designed and tested in simulation This research opens a new research line in fault tolerant aircraft systems in the aerospace and engineering research group at Universidad Pontificia Bolivariana and has the potential to complement the postgraduate courses. The developed system is a promising solution to the problem of air data system failure which represents a significant contribution to aircraft safety: The developed system can be integrated to current air data system without mayor modifications. The technological novelties can be described as follows:

- A feedback from the sensor failure detection and identification module to the aircraft state estimator, so the covariance of the noise of the faulty sensors and the covariance of the process noise in the estimation filter is modulated in such a way that these erroneous measurements are isolated and do not affect the accuracy of the aircraft state estimation process. The system is capable of returning to normal operation if the faulty sensors becomes healthy again as it will be the case if ice covering the pitot or angle of attack melts.
- The modular design of the SFDIA algorithm allows extra Pitot, angle of attack vane and angle of sideslip vane sensors to be added easily to obtain sensor redundancy and increase the system reliability. This is possible because the neuro-fuzzy model used in the digital twin is only employed to model the aircraft dynamics and acts as

an extra “virtual sensor”. Fault detection and accommodation can easily be scaled to other sensors as it was done to the angle of attack vane sensor failure.

- The state estimation algorithm process uses aerodynamic and thrust forces computed during flight as measurements to increase the estimation accuracy. A grey-box adaptive neuro-fuzzy neural network model based on Takagi-Sugeno-Kang method of Adaptive Neuro-Fuzzy Inference System, acting as an aircraft digital twin that provides virtual aerodynamic and propulsion forces as measurements to help during the estimation process when the Pitot and Angle of Attack sensors fail independently and together. The digital twin gives a reliable prediction in the complete range of aircraft speeds and maneuver types allowing the estimation to be precise in all stages of flight.
- Faults in airspeed sensors were modelled based on the physical phenomena of the failures. Dynamic pressure orifices blockage, was modelled allowing the airspeed readings to vary with the change in static pressure. Stuck Pitot and angle of attack vane failures were also modelled, where the sensor signals remain constant independently of aircraft maneuvers.
- The fault detection is accomplished by analyzing the airspeed and angle of attack residual signals against a threshold computed from the mean standard deviation, but removing carefully its peaks using low-pass filters. The filter to compute the mean residual signal is tuned with a higher band width than the standard deviation filter which is used to compute threshold but both covering the spectrum presented during aircraft maneuvers. As a result, the residual signal mean changes faster that

the computed thresholds assuring fault detection with minimum undetected or false faults.

- A novelty introduced by this research is that the faulty sensors are always used in the estimation process independently of the occurrence of the failure and that there is always a permanent closed loop feedback between the air data estimator and the fault detection module. Apart from being computationally simple, the fault tolerant air data estimation scheme eliminates false alarms or undetected failures and handles all uncertainty of the model giving very accurate results.

1.5 Thesis Organization

This dissertation is divided into eight chapters. After this brief introduction, Chapter 2 presents background and related work including a discussion of the state of the art in Non-Traditional Air Data Systems. Chapter 3 gives a brief review of the foundations of aircraft flight dynamics. A deep discussion about the proposed air data system tolerant to sensor failure is presented in Chapter 4. In Chapter 5, the Matlab/Simulink® implementation of the simulation model is discussed extensively. Chapter 6 presents the simulation results. Chapter 7 presents the conclusions of the research findings. Finally, Chapter 8 presents the future work that this research opens.

CHAPTER 2. THE AIR DATA SYSTEM

2.1 Traditional Air Data Systems

The ADS measures the aircraft speed relative to the air mass in which it is flying by means of its sensors that are traditionally the Pitot tube, static pressure port, angle of attack vane, angle of side slip vane and a temperature probe as shown below in Figure 1. Pneumatic lines connect the pressure taps on the Pitot tube and static ports with pressure transducers that, in turn send signals to an air data computer.

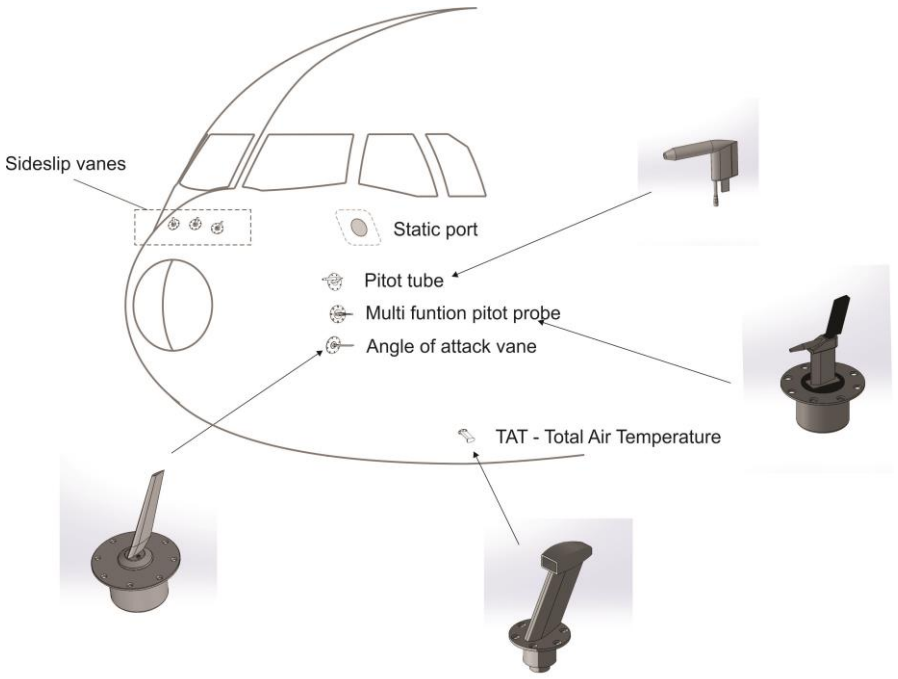

Figure 1. ADS sensors on Airbus A350XWB.

The computation of airspeed from the Pitot tube, static port and total temperature probe sensor measurements is performed as shown below in the block diagram of Figure 2. The Mach number is computed using:

$$M = \sqrt{5\left[\left(\frac{p_t - p_0}{p_0}\right)^{\frac{2}{7}} - 1\right]} \quad (1)$$

where p_t and p_0 are the pitot (or total) and static pressure, respectively. The Pitot and static pressure measurements must be compensated to account for deviations due to the local flow disturbance in their respective installation locations. The compensations depend on the Mach number and the position of the sensor on the fuselage and its calibration is achieved through extensive wind tunnel and flight tests [21]. Performance standards of an ADS to be certified are described in [22].

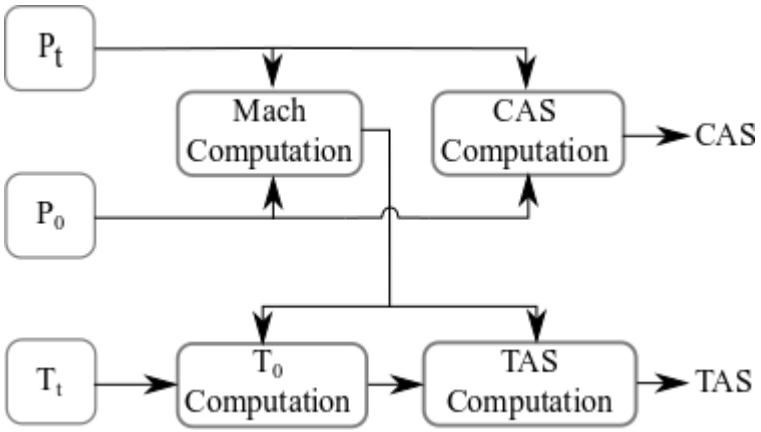

Figure 2. Airspeed computation algorithm.

The angle of attack and sideslip vanes can rotate freely and they self-align with the local air flow direction. In modern ADS's sensor redundancy is achieved not only by installing multiple identical sensors but also by using extra sensors that perform multiple measurements each as in the case of the multi-function pitot probe of the A350XWB shown in Figure 1 [23] or the five-hole Pitot probe used on the F-16A jet fighter shown in

Figure 3 [24], [25], which measures angle of attack and sideslip by differential pressure in addition to Pitot/static data.

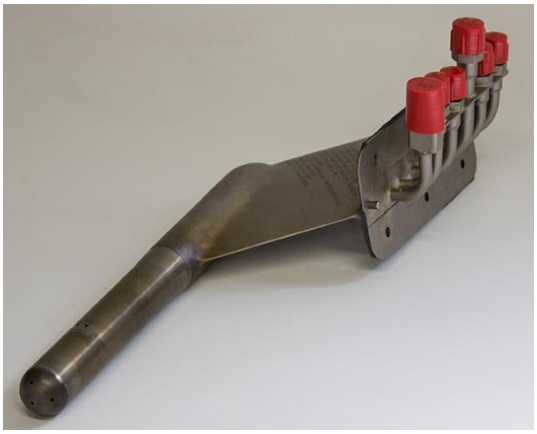

Figure 3. Rosemount five-hole Pitot used in the F-16A.

2.2 Air Data System Failure

The Pitot tube sensor is equipped with a heating element that will be activated to prevent ice formation depending on the atmospheric conditions. In a cold and humid atmosphere, excessive amount of ice can build up in the Pitot tube and the heating element might not be able to fully clear up the ice build-up. If the drain hole is not affected by icing, the dynamic pressure inside the sensor will gradually decrease to zero when the total pressure reaches the static atmospheric pressure. In this situation, the computed airspeed will reach very low values because p_t will be very similar to p_0.On the other hand, if both the drain orifice and the Pitot tube are completely blocked, p_t will remain constant and how the airspeed varies depends on the vertical movement of the aircraft. As the aircraft climbs, p_0 decreases resulting in increase of airspeed estimate. In this case the aircraft can be subjected to inadvertent stall because the pilot would react by increasing the angle of attack or reducing the thrust to reduce the apparent increase in airspeed. In the opposite

scenario, during an aircraft descent, p_0 increases resulting in a decrease in airspeed estimate and a structural failure can occur if the pilot counteracts the situation by reducing the angle of attack or increases the thrust to compensate for the apparent decrease in airspeed.

In the general aviation arena, the loss of control of the airplane as a result from aerodynamic stall has been the major cause of accidents. The National Transportation Safety Board – NTSB statistics shows that there have been 1,396 fatal accidents between 2006 and 2015 caused by loss of control [26] where 1% are attributed to sensor system failure/malfunction and erroneous response. Trying to minimize the overall loss of control statistics, which accounts for 1,200 fatalities between 2001 and 2010, the FAA approved in 2014 the design and use of non-certified, non-required and supplemental angle of attack indicator on general aviation airplanes as an independent and stand-alone systems [27].

It can be concluded that a reliable ADS is fundamental to ensure that the aircraft flies within the safe flight envelope. The current solutions used to achieve the required level of reliability and dependability is through physical redundancy of sensors, actuators and flight computers.

Figure 4 shows how reliability and dependability is achieved on the Airbus A330 which features three air data systems. Each air data computer, denoted as ADIRU from the acronym for Air Data and Inertial Reference Unit, is completely independent from each other and could be used in the control and guidance of the airplane.

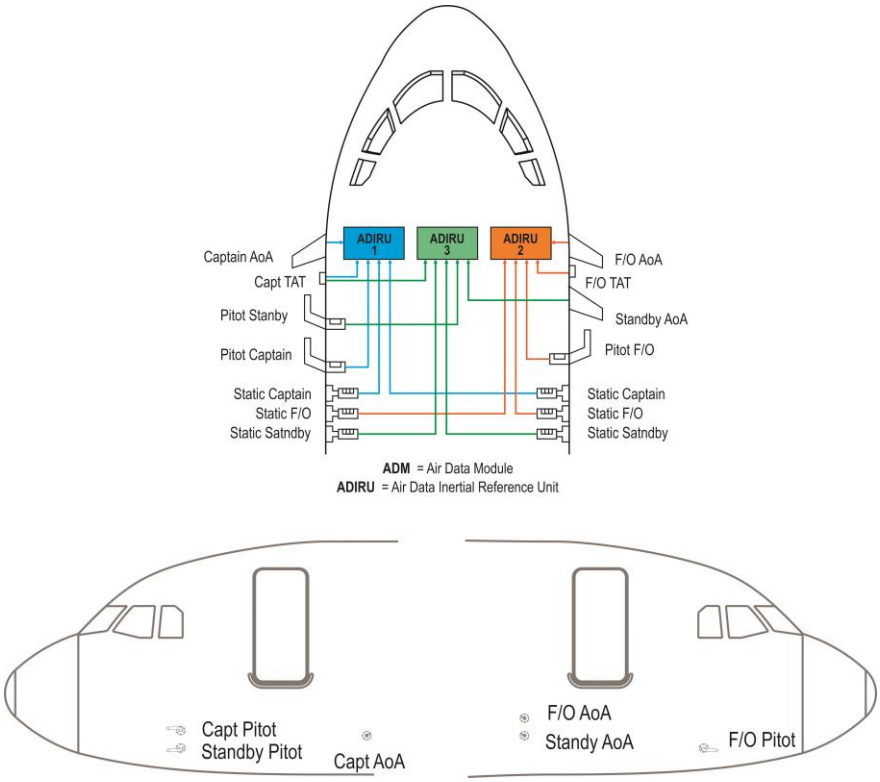

Figure 4. Airbus A330 ADS layout.

2.3 Importance of an Air Data System Tolerant to Sensor Failure

The recent emergence of small unmanned aerial system – UAS technologies and their potential to perform missions sharing the airspace with other flight vehicles had attracted the interest of the research and industry community in how to design low cost and highly reliable avionics that meets the requirements of low weight, size and power. In these small vehicles usually system redundancy is not possible to be implemented, so the idea of numerical redundancy has received some attention. Numerical redundancy, also known as virtual sensors, is a term used to describe the idea of replacing physical hardware with numerical models in redundant and safety critical aircraft systems. A device or method capable of making reliable estimates of air data quantities is highly attractive. This new

technology will not only be useful for small UAS's, but also will be attractive for general and commercial aviation where it could help to mitigate the effects of unreliable avionic systems that causes the loss of control described earlier.

The virtual sensor necessary hardware is composed of some ADS-independent sensors and a computer. An estimation/fault detection algorithm will run in real time to isolate the effects of the faulty sensor. The work presented here uses this technique to provide an estimation of air data using the inertial measurement unit (IMU) as vital measuring sensors, and available air data sensors. Aiding sensors like GPS and Magnetometer are used to help improving the accuracy of the estimates, but they are not compulsory for the system to work. Apart from providing a backup for the primary air data computers, the independence of this system from the conventional ADS, allows a sensor fault detection, isolation and accommodation in the event of a failure in the primary or a traditional ADS.

2.4 Air Data Systems with Non-Traditional Sensors

There have been recent research efforts of some companies and research organizations in the development of an alternative device for airspeed measurement that can be applied to general aviation, commercial and military aircraft. The airspeed sensing devices that are being developed can be classified in laser-based and ultrasound-based technology.

NASA Dryden Research Center has been working with Ophir Corporation since 2002 to provide proof-of-concept laboratory testing of a new approach to Optical Air Data Systems (OADS) [28]. The resulting technology is called SenseAir, which uses

Rayleigh/Mie LIDAR, an optical laser-based, remote sensing technique that measures air data outside a vehicle's boundary layer. Laser light is scattered by the atmospheric aerosols and gas molecules, and an optical receiver collects a fraction of that light. The air molecules or aerosols have a net velocity along the direction of the laser light propagation, and the center wavelength of the scattered light is Doppler shifted. Measuring this frequency shift allows the relative airspeed to be measured along the laser's line of propagation. The laser light is transmitted and collected along three different axes, and then the three-dimensional air velocity is measured. This provides total airspeed, angle of attack, and angle of sideslip. The optical scattering from atmospheric gas molecules, known as Rayleigh scattering, is used to obtain the mean random molecular velocity from which atmospheric temperature is derived and density is obtained. Then, using the ideal gas law formula, pressure is obtained to provide the remaining parameter in the set of air data parameters, with the three vector components of airspeed and atmospheric temperature. When the air data measurement is made sufficiently ahead of the flight vehicle, the disturbance to the measurement by the presence of the airframe is acceptably small so that one system calibration can be used for all aircraft.

The European Union has funded a similar development to the SenseAir system, the "NEw Standby Lidar InstrumEnt" – NESLIE [29]. The NESLIE instrument was further developed under the "Demonstration of ANemometry InstrumEnt based on Laser" - DANIELA project, where hardware and software developments as well as extensive flight test were carried out on the NRL Cessna Citation II research aircraft [30].

Similar Laser based developments are currently carried out by BAE Systems and Airbus. The instrument is called "Laser Air Speed Sensing Instrument"- LASSI [31].

Airbus has developed a laser-based airspeed sensor that was tested successfully in the Airbus Dauphin 6542 helicopter and is currently starting the certification process [32].

The University of New South Wales has developed an airspeed sensor based in diode sensors which use laser absorption spectroscopy to measure the flow properties. Mach number and angle of attack are estimated based on the measured absorption spectrum of oxygen molecules on the flow. This technology was initially developed for sensing airspeed in Scramjets, but according to their publications it has been successfully tested in subsonic flows [33].

The non-traditional air data systems presented in this section is a promising and attractive technology to be incorporated in flight vehicles, however technology challenges must be addressed before these systems can operate on fixed wing aircraft. The first challenge to be addressed is the effect of the variability of wind tracer concentration and distribution on the airspeed measurement accuracy during operation in actual conditions. The third challenge is the un existent certification procedures and the difficulty of integrating this systems un current aircraft.

2.5 Air Data Systems with Numerical Redundancy

The airspeed sensor failure problem has been considered within the general research area known as Sensor failure detection identification and accommodation (SFDIA). In General, SFDIA has been traditionally divided in two main activities. The first task involves the detection of the corrupt sensor signal and identification (SFDI) of the corresponding malfunctioning sensor hardware so it can be isolated (remember that redundancy in the quantity of a particular sensor is usually present in commercial

airplanes). Accommodation (SFA), the second task, consist on the numerical estimation of the airspeed by means of a virtual sensor and the statistical comparison of this value with the sensor (or sensors) signal so the healthy one can be selected and used by the flight control system.

Two approaches are used in the development of virtual sensors. Model-based virtual sensors use dynamic models which are unique for a particular aircraft configuration and model-independent virtual sensors that can be applied to any aircraft configuration. The definitions “model-based” and “model-independent” refers to a particular aircraft type and not to the type of dynamic model employed. A dynamic model for a particular aircraft is usually created from confidential flight test data from which the relevant parameters are identified. Trying to overcome the confidentiality barrier of flight test data from manufacturers, researchers are working in the development model-independent virtual sensors, however aircraft flight test data is necessary for the development of this type of virtual sensor.

2.5.1 Model Independent Virtual Sensors

The model-independent virtual sensors are of great interest because the estimation method can be applied conveniently to any aircraft without information of the vehicle dynamics which most of the time is difficult to obtain as explained above. Kalman Filter based estimators and data driven methods based on artificial neural networks - ANN are commonly employed. Napolitano et al. [34], [35], developed model independent virtual sensors by using extended and unscented Kalman Filter estimators (EKF and UKF respectively). Their data-driven approach of model independent virtual sensors take

advantage of ANN's (Multi-Layer Perceptron - MLP and Extended Minimal Resource Allocating Network - EMRAN). These approaches are evaluated in terms of their performance (by means of mean and standard deviation) in the estimation of the air speed. The advantage of neural networks methods relies that it is not required and exact knowledge of the aircraft dynamic model but a large flight data base with sensors information on the complete flight envelope will be necessary. The previous methods also use information from the other airborne sensors to estimate the airspeed, particularly the three axis accelerations and angular rates from the IMU, roll and pitch angle measurements from the vertical gyro and angles of attack and sideslip from the ADS.

The MLP ANN is used for the estimation of the airspeed because of its flexibility for a number of applications, including function fitting and pattern recognition. The MLP is trained to learn the functional relationships between the airspeed and a set of correlated measurements provided by the other aircraft sensors. It was determined experimentally, and also by the fact that the function being approximates is nearly linear, that using approximately half of the number of inputs as the number of neurons in the hidden layers a compromise between real time performance and airspeed accuracy is achieved. The MLP architecture employed in airspeed estimation consists of one input layer with 7 inputs, one hidden layer with three neurons, a sigmoid activation function and a single output layer. The training of the ANN uses as inputs the angular rates, linear accelerations, angle of attack, pitch rate, pitch angle, stabilator deflection, throttle control, longitudinal linear acceleration and vertical linear acceleration. Once these inputs are identified, the general procedure that is followed in training and validating the performance is:

a) Create and configure the network.

b) Initialize the interconnection weights and biases using the Widrow-Nyguyen initialization procedure, which distributes the weights in the active region of each neuron in the hidden layer approximately evenly in the layer's input space decreasing the training time.

c) Process the input data and train the network for a user-defined number of instants or until a low output error is achieved.

d) Validate the performance of the network in terms of a tracking error defined as the difference between the Pitot sensor airspeed and the ANN estimated airspeed.

The EMRAN ANN architecture allows only the parameters of the most active neurons to be updated, while all the others are left unchanged. Essentially, the EMRAN algorithm allocates neurons in order to decrease the estimation error in regions of the state space where the mapping accuracy is poor. This strategy results in a significant reduction of the number of parameters to be updated online, thus reducing the computational burden, and therefore making this architecture particularly suitable for online applications. The procedure followed to train and validate the EMRAN ANN is similar to the MLP ANN with the difference that in initializing the interconnection weights and biases step the network starts without hidden layer neurons based on the three error criteria [35].

Some other types ANNs had been used in aircraft airspeed estimation. Husain [36] performed simulations using a fully connected cascade ANN for airspeed estimation arguing that it requires a much lower computational effort than MLP and MRAN ANNs.

Turkmen et al. [37] used information from a 737-flight data recorder to train an MLP and applied the Levenberg- Marquardt training algorithm claiming fast learning and good convergence properties. Turkmen results showed that the ANN was capable of predicting data with a high level of accuracy, however the method hasn't been tested during flight.

Alireza et al. [38] used an adaptive NN in which its learning coefficients are updated by an EKF. The online adaption of these weighting parameters helps to detect abrupt intermittent and incipient sensor and actuator faults accurately. The researchers state that his method achieved higher accuracy, less convergence time and greater efficiency than related approaches in literature. Their approach was implemented and evaluated using a six degree of freedom (DoF) nonlinear dynamic model of the WVU YF-22 unmanned aircraft in cruise flight.

An effort in reducing false alarms on sensor failure detection was done by Qi et al. [39] by the development of an adaptive threshold neural-network scheme for Rotorcraft Unmanned Aerial Vehicle (RUAV) sensor failure diagnosis which can be applied to fixed wing aircraft. The approach uses two threshold values (low and high). If a lower threshold is exceeded once by the sensor signal value, the status of the corresponding sensor is declared suspect and the numerical architecture of the NN containing the sensor estimator is not updated. If this status persists for a certain number of time instants and/or the estimation error in successive time instants exceeds the higher threshold, the sensor is then declared failed.

Swischuk et al [40] proposed a machine learning approach to aircraft sensor error detection and flight data correction. The method was demonstrated on real flight data

containing pitot static system failures. By calculating autocorrelation online and comparing it to an off-line library, the aircraft was able to accurately identify the failures in the pitot static system during climb and cruise flight. The research showed that faults were identified after 20 seconds and that the airspeed prediction that can be incorrect by up to 41 knots. The researchers claim that the accuracy of the airspeed estimation and the fault detection speed can be improved by using more flight data.

Garbarino et al [41] used a set of three MLP ANN's to estimate the airspeed, each one using a different set of inputs. The inputs to ANN1 are the GPS Velocity, AHRS data, Surface Position, Angle of Attack and Sideslip. In the absence of GPS data, the ANN2 is used which receives inputs of the AHRS, Surface Position, Angle of Attack and Sideslip. The ANN3 is used in the absence of AHRS data, so it receives inputs in terms of GPS Velocity, Surface Position, Angle of Attack and Sideslip. The selection of the appropriate ANN is done by a diagnostic module which receives data from the built-in diagnostic systems available on some of the sensors installed on-board, i.e. GPS and AHRS, which allows to determine the sensor output availability. Each time a failure occurs, the best performing ANN will be selected, based on the residual input's availability. The output of the selected ANN is compared with the actual measure to identify the fault occurrence, which is signaled if difference overcomes a predefined threshold. The detection thresholds are dynamically scheduled on the base of the ANN performance. To avoid false alarms the fault must be on hold for a fixed continuous time, also in this case the value of this observation time is dynamically scheduled based on the ANN selected. This fault tolerant architecture was tested by means of MATLAB simulations during cruise flight with

reasonable accuracy in the results. The authors claim the system works during take-off and landing phases of flight.

Hansen et al. [42] presented a solution to the Pitot tube failure where wind speed vector and propulsion dynamics are unknown. An EKF technique is adopted to estimate the wind velocity vector while propulsion system parameters and airspeed are estimated by a dedicated model-based nonlinear observer. The fault detection thresholds are designed by analyzing the sensor signal residual statistics in the fault free condition by means of a generalized likelihood change detection. The method was verified both using real data from a UAV lost due to an airspeed sensor fault. The simulation showed that, with the false alarm rate chosen, speed sensor faults exceeding 6.5 m/s are detectable.

In [43] a method is developed for estimating airspeed, angle of attack, and sideslip based on measurements from GPS, inertial measurement unit (IMU) and a low-fidelity model of the aircraft's dynamics which are fused using two, cascaded Extended Kalman Filters. In the cascaded architecture, the first filter uses information from the IMU and GPS to estimate the aircraft's absolute velocity and attitude. These estimates are used as the measurement updates for the second filter where they are fused with the aircraft dynamics model to generate estimates of airspeed, angle of attack and sideslip. The performance of is scheme is shown to be relatively insensitive to off-trim attitudes but very sensitive to off-trim velocity. C. Lu et al. [44] extended Lie´s approach by using an Initialized Three-step Extended Kalman Filter (ITEKF), which considers the wind speed as an unknown input to track the variation of wind speed.

The estimation methods described above require time consuming tuning procedures. To overcome this problem current research effort is being conducted based on a semiautomatic data driven approach to select model regressors, to identify Nonlinear Autoregressive Exogenous – NARX input-output NN prediction models, to set up fault detection filters and to compute fault detections thresholds (EWMA filters) [45]. This approach provides online model adaptation mechanisms to cope with time dependent and flight dependent levels of uncertainties. In this approach the data driven modeling of the aircraft was obtained during flight tests at relatively constant speed and altitude, excluding the take-off, initial climb, final descent and landing phases because they are associated with specific aerodynamic configurations due to the deployment of flaps. The instrument data was synchronized and sampled at a rate of 10 Hz. These flights contained a large number of maneuvers with only a limited number of straight steady-state segments.

2.5.2 Model-Based Virtual Sensors

The model-based airspeed estimation approach usually takes advantage of the well-known aircraft non-linear model [46] and redundant measurements from the on-board sensors. The commonly used model state variables are the true airspeed, angles of attack and sideslip, angular rates, Euler attitude angles and the GPS-based aircraft positions. The state control variables rely on the thrust force and the elevator, rudder, and aileron deflections. The airspeed is implicit in all the 12 equations of the aircraft nonlinear model, however in [34] an effort was made to reduce the computational effort by using only the angle of attack equation that is strongly correlated with the air speed. This equation is expressed in a form that is suitable for parameter identification with a linear combination of unknown coefficients and known nonlinear function of measured signals. The equation

is evaluated at several instants on a time window to set up a linear system of equations that is solved for the unknown equation coefficients using least squares and taking advantage of previously recorded flight test data. The online estimates of the airspeed at a particular time are based on the online solution of the quadratic equation were the unknown coefficients are substituted by the LS estimate and only the positive value of the airspeed is selected. The results showed that the accuracy of the method is about 5% of the mean inflight velocity. Some model-based recent methods from several researchers will be described below.

Felemban et al. [47] developed a model-based airspeed estimation by using a Continuous Polynomial Adaptive Estimator (CPAE). The CPAE was successfully implemented on a theoretical decoupled longitudinal aircraft dynamic model. This analysis showed that the estimated values of airspeed converge correctly.

Amato et al. [48] approach to sensor failure and isolation in the presence of disturbances exploits the so-called nonlinear unknown input observer UIO in conjunction with an H_∞ procedure to calculate the observer gains to guarantee disturbance decoupling (wind gust, atmospheric turbulence and sensor noise). The proposed nonlinear observer uses a six degree of freedom aircraft model and it was tested via numerical simulations at NRL's research flight simulator using a small commercial aircraft simulation model. The algorithm uses hardware and software measurements to decide if and where a fault has occurred, isolate the faulty signal and provide the corresponding healthy signal. The researchers concluded that a good knowledge of the nonlinear aircraft model significantly improves the performance of the detection and isolation algorithms.

2.5.3 Possible Improvements in Virtual Sensors

Analyzing the multiple researches in airspeed estimation described above, it can be inferred that improvements can be made by revisiting several aspects as described next. The estimation algorithms based on Kalman Filters can be augmented to retrieve the accelerometer and gyroscope bias which are fundamental to improve the accuracy of the estimated variables. Sensors can be modelled by incorporating not only its bias, but also the transfer function that describes its dynamics. The estimation described previously only use the measures of ground velocity of the GPS, angle of attack and angle of sideslip from the respective vanes, better predictions can be achieved by incorporating control inputs and data from a high-fidelity model (a digital twin aircraft dynamic model) that is trained based on flight data. When using a black-box model based on data like a NN, improved accuracy can be achieved on the response of the model if the variation of the inertial parameters and the position of CG is properly modelled. The same applies if the variation of gravity is included in the model. The current steady state black-box models can also be improved if they are constructed with flight test data in such a way that the aircraft dynamic model is fully represented.

2.5.4 Pitot Tube Fault Modeling and Residual Signals.

The fault on the airspeed sensor is usually modeled as an additive bias of amplitude ΔV occurring at time t_f, as in [49]. Thus, the resulting faulty signal is

$$V_{faulty}(t) = V(t) + \Delta V(t - t_f) \tag{2}$$

Two suddenly fault scenarios commonly used in fault detection and identification practice are implemented: a sudden bias (SB) failure and a slow ramp bias (SRB) failure as illustrated on Figure 5 and Figure 6 respectively [49]. The SB fault is a sudden step bias on the airspeed measurement at the instant of failure, whereas the SRB fault is designed to reach its amplitude after a certain $\Delta T_{Failure}$ with the failure amplitude increasing linearly to its final value.

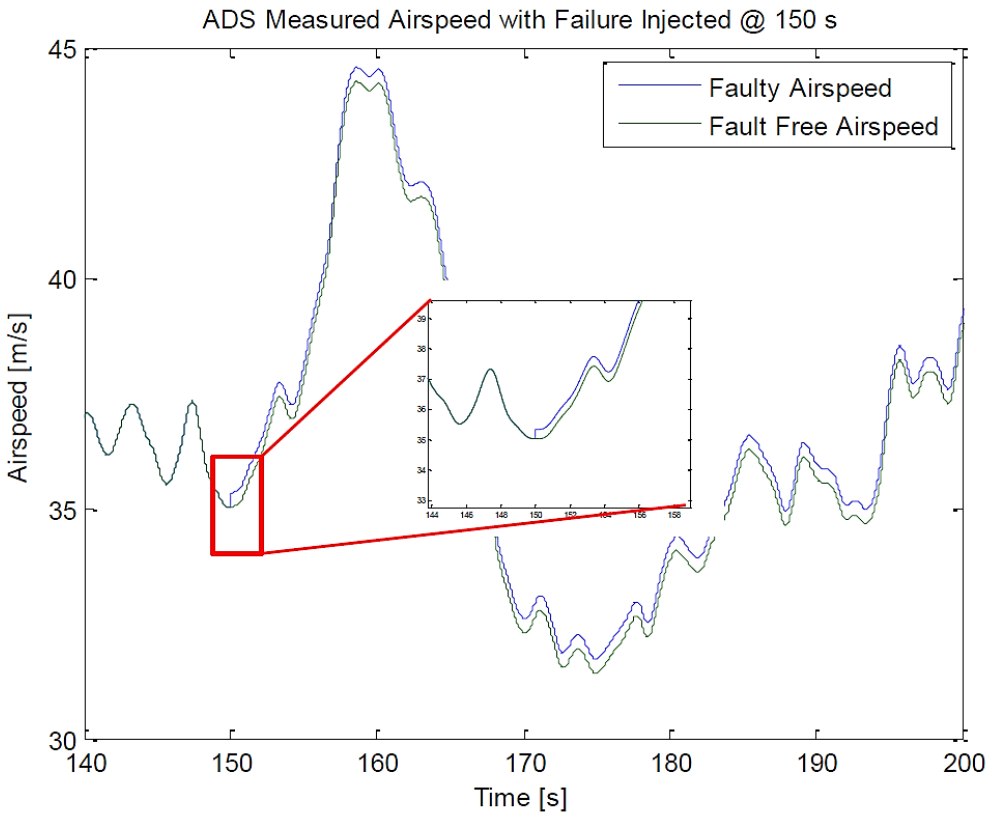

Figure 5. Sudden bias failure.

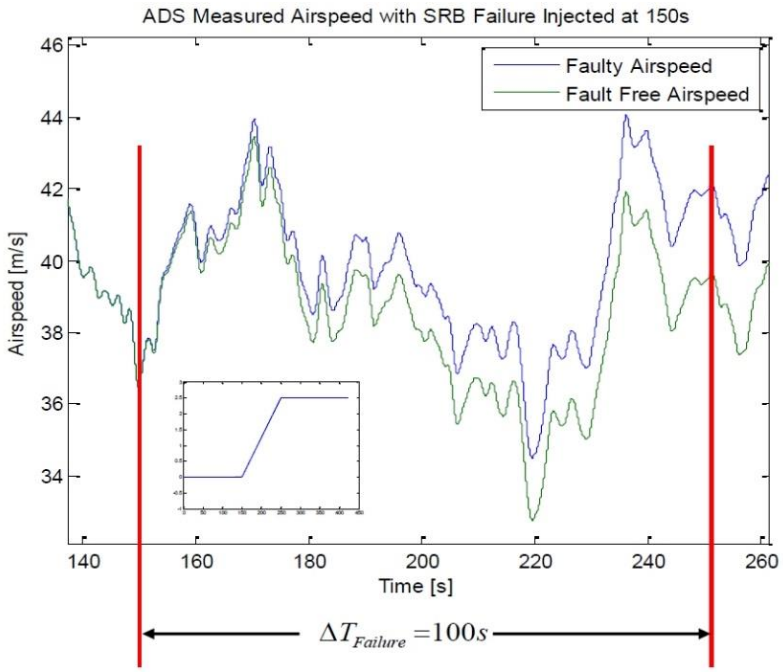

Figure 6. Slow ramp bias failure.

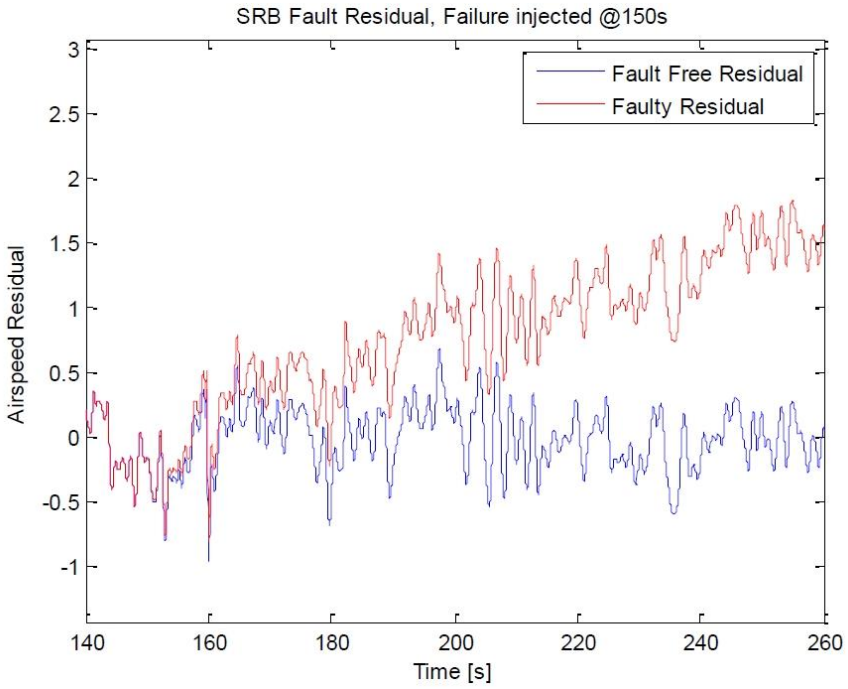

Figure 7. Residual signal for a SRB failure.

The residual signal used for the purpose of SFD is usually defined as the difference between the measured air speed from the ADS $V(t)$ and the estimated airspeed $\hat{V}(t)$:

$$r(t) = V(t) - \hat{V}(t) \tag{3}$$

A typical residual signal for an SRB failure is shown in Figure 7 for a simulated failure of magnitude 2.5 m/s injected into the flight data [49]. Theoretically the residual signal should approach to zero in the fault free condition scenario, but in the real world, due to modeling uncertainties and noise, this is not the case. It is found in experiments that the raw residual signal has a significant auto correlated length that it is introduced mainly by uncertainties in the low frequency range. Since statistical detectors perform optimally when residual signals are completely uncorrelated, a whitening filter is usually designed to remove the residual signal correlation.

For discrete time applications, the failure on the airspeed sensor is usually modeled similarly as an additive fault ΔV that occurs at a time instant t_f and the resulting faulty signal is:

$$V_{faulty}(t) = V(t) + \Delta V(t - t_f) \tag{4}$$

To allow for the analysis of both hard and soft sensor failures, the ΔV fault was derived as the output of a unit gain low pass filter with a time constant τ as follows:

$$\Delta V(t) = \frac{1}{1 + s\tau}\bar{\Delta} V \delta_{-1}(t) \tag{5}$$

where $\delta_{-1}(t)$ is a unitary step fault function and $\bar{\Delta}V$ is the actual steady state fault amplitude. By varying τ, it is possible to simulate fast (hard) and slowly rising (soft) failures.

The residual signal that is used for SFD purpose is defined as the difference between the measured airspeed coming from the sensor and the estimation as follows

$$r(t) = \big(V(t) + \Delta V(t-t)\big) - \hat{V}(t) \tag{6}$$

As with the other approaches, in the fault free conditions, the residual $r(t)$ should statistically approach zero, while in real world applications, due to the presence of modeling uncertainties and noise $r(t) \neq 0$ even without failures. Therefore, research efforts have been focused on the development of decision algorithms able to detect the occurrence of a fault while minimizing the rate of false alarms.

2.5.5 Sensor Fault Detection

Several approaches had been used for on the SFDI task based in fault detection filters. The original approach consists on a fixed failure detection threshold using a Cumulative Sum filter – CUSUM and the second approach is based generalized Likelihood ratio test – GLRT [50]. A third approach consist on a floating limiter – FL [49]. In either approach, the difference between the sensor measurement signals and the numerically estimated value, so called the residual signal, is compared with the threshold and the failure will be identified when the residual exceeds a particular threshold value thus triggering the failure detection. Recent research efforts are focused on minimizing the number of false alarms or undetected failures in the previous approaches by using the exponentially weighted moving

averaged – EWMA filter that detects small shifts in the mean and standard deviation of process variables [51]. The EWMA chart tracks the EWMA mean of all previous samples so that the most recent are weighted more heavily than the older ones.

Recent developments a FDI method that uses a non-linear-in-the-parameter Neural Network model to characterize the nominal nonlinear response in the different phases of flight and by means of an additional linear in the parameters Interval Model the uncertainty is captured [52]. In this method the FDI is done by verifying whether the measures ADS signals fall within time varying bounds predicted by the nonlinear Interval Model.

CHAPTER 3. AIRCRAFT FLIGHT DYNAMICS BACKGROUND

3.1 Aircraft Equations of Motion

Before explaining the proposed Air Data System tolerant to sensor failure, it is worth reviewing the aircraft six-degree of freedom nonlinear dynamic model. The derivation and notation shown in this chapter is developed by Dr. Luís Benigno Gutiérrez Zea in his undergraduate course of flight dynamics and can be found in [53].

The aircraft six degree of freedom dynamic model used for the aircraft flight simulation consist in four differential equations that represents its translational and rotational kinematics and its translational and rotational dynamics. These dynamic equations are expressed in vector form as follows:

$$\dot{P}_e = C_{e/b} V_b \tag{7}$$

$$\dot{\Phi} = \mathrm{H}(\Phi)\Omega_b \tag{8}$$

$$\dot{V}_b = \frac{1}{m} F_{net,b} - \Omega_b \times V_b. \tag{9}$$

$$\dot{\Omega}_b = {I_b}^{-1}\left(M_{net,b} - \Omega_b \times I_b \Omega_b\right). \tag{10}$$

Here, the subscripts "e" and "b" represents variables that are expressed in the earth (inertial) frame F_e and in the aircraft body frame F_b respectively. The aircraft velocity V_e is the rate of change of its position $\dot{P}_e$ expressed in the inertial frame, F_e. The direction

cosine matrix $C_{e/b}$. performs a reference frame transformation from F_b to F_e. The vector V_b represents the aircraft velocity expressed in F_b, and its components are u, v and w.

The angular velocity vector Ω_b, with components p, q, and r, represents the change in orientation of the aircraft with respect to F_e. Its direction specifies the instantaneous axis of rotation and its magnitude represents the instantaneous rate of change of the angle measured around the axis of rotation.

The aircraft attitude is defined by the vector Φ which components are the Euler angles $\emptyset, \theta$ and ψ. The matrix $\mathrm{H}(\Phi)$ relates the rate of change of Φ with Ω_b and is defined as

$$\mathrm{H}(\Phi) = \begin{bmatrix} 1 & tan\theta sin\emptyset & tan\theta cos\emptyset \\ 0 & cos\emptyset & -sin\emptyset \\ 0 & \frac{sin\emptyset}{cos\theta} & \frac{cos\emptyset}{cos\theta} \end{bmatrix}. \tag{11}$$

The aircraft linear acceleration $\dot{V}_b$ is computed by dividing the net force acting on the aircraft $F_{net,b}$ by the aircraft mass m. The term $\Omega_b \times V_b$ accounts for the derivative of the vector V_b acting on the rotating frame F_b.

The aircraft angular rate $\dot{\Omega}_b$ is computed using the net moment actin on the aircraft $M_{net,b}$, Ω_b and the aircraft inertia matrix I_b,which is defined as

$$I_b = \begin{bmatrix} \mathrm{I_x} & -\mathrm{I_{xy}} & -\mathrm{I_{xz}} \\ -\mathrm{I_{xy}} & \mathrm{I_y} & -\mathrm{I_{yz}} \\ -\mathrm{I_{xz}} & -\mathrm{I_{yz}} & \mathrm{I_z} \end{bmatrix}, \tag{12}$$

here, I_x, I_y and I_z are the aircraft moments of inertia.

3.2 Forces and Moments Acting on the Aircraft

The body of the aircraft is subjected to several forces: gravitational forces (weight W), aerodynamic forces F_a, and propulsion forces F_t. The sum of these forces represents the net force acting on the aircraft expressed in the body frame is

$$F_{net,b} = W_b + F_{a,b} + F_{t,b} \, , \tag{13}$$

The body of the aircraft is also subjected to aerodynamic and propulsion moments M_a and M_t respectively. The net moment acting on the aircraft expressed in the body frame is

$$M_{net,b} = M_{a,b} + M_{t,b} \, . \tag{14}$$

The weight is the gravitational force acting on the body of the aircraft

$$W = mG \, , \tag{15}$$

where G is the gravity vector expressed in F_b. This force acts on the center of mass of the aircraft.

3.3 Aircraft Controls

The aircraft controls are used in the mathematical modelling of the aerodynamics. The aircraft basic controls are the ailerons for roll control, elevator for pitch control, rudder for yaw control, throttle for thrust control and flaps to allow the aircraft to fly at lower speed as shown in Figure 8. There are aircraft designs where the horizontal tail incidence

is varied instead of the elevator control. The sign convention of control deflection is as follows; Flaps: δ_f (positive down, always positive or zero), Elevator: δ_e (positive down), Aileron: δ_a(positive when right aileron down) and Ruder: δ_r(positive to the left).

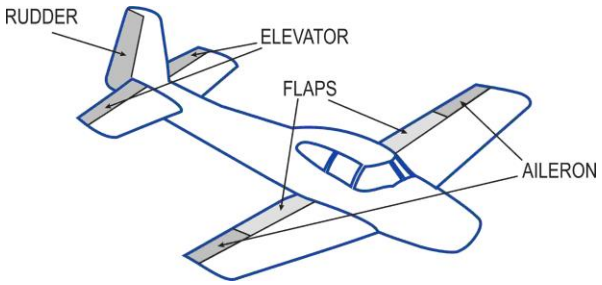

Figure 8. Aircraft control surfaces.

3.4 Aircraft Aerodynamic Forces

Figure 9, shows the aerodynamic forces acting on the aircraft. The aerodynamic forces expressed in the wind frame F_w are

$$F_{a,w} = \begin{bmatrix} -D \\ -C \\ -L \end{bmatrix}, \tag{16}$$

where D is drag force, L is the lift force component perpendicular to drag force and C is the crosswind force component. The angle of attack α defines the orientation of the stability frame F_s and the angle of sideslip β defines the rotation of F_s to reach F_w. Three dimensionless coefficients are defined to represent the aerodynamic forces as follows

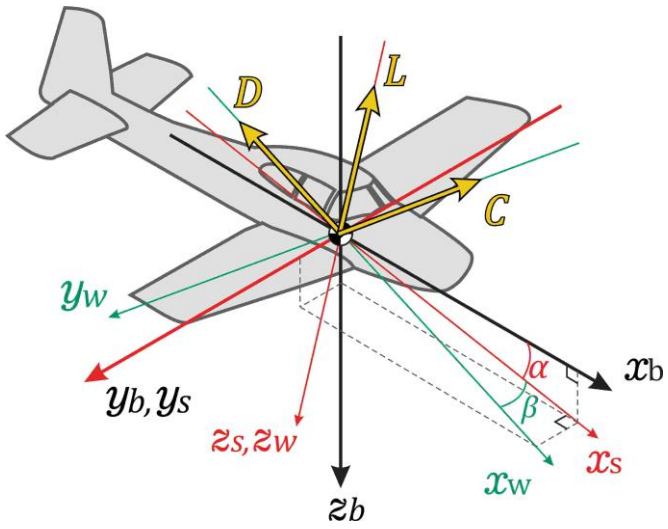

Figure 9. Aerodynamic forces acting on the aircraft.

$$C_D = \frac{D}{\bar{q}S}, \tag{17}$$

$$C_C = \frac{C}{\bar{q}S}, \tag{18}$$

$$C_L = \frac{L}{\bar{q}S}, \tag{19}$$

where S is the wing's reference area and $\bar{q}$ is the air dynamic pressure defined by

$$\bar{q} = \frac{1}{2}\rho V^2, \tag{20}$$

with ρ being the air density and V the airspeed.

The model for lift coefficient is defined as follows:

$$C_L = C_{L,0} + C_{L,\alpha}\alpha + C_{L,\delta_f}\delta_f + C_{L,i_h}\delta_{i_h} + C_{L,\delta_e}\delta_e + \frac{c}{2V}\left(C_{L,\dot{\alpha}}\dot{\alpha} + C_{L,q}q\right) + C_{L,M}(M - M_0)\,, \tag{21}$$

where c is the mean aerodynamic chord, V is the airspeed, M is the Mach number, M_0 is the Mach number at a given flight condition.

- $C_{L,0}$: Value of lift coefficient C_L, when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- $C_{L,\alpha}$: $\frac{\partial C_L}{\partial \alpha}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- C_{L,δ_f}: $\frac{\partial C_L}{\partial \delta_f}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- C_{L,i_h}: $\frac{\partial C_L}{\partial i_h}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- C_{L,δ_e}: $\frac{\partial C_L}{\partial \delta_e}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- $C_{L,\dot{\alpha}}$: $\frac{2V}{c}\frac{\partial C_L}{\partial \dot{\alpha}}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- $C_{L,q}$: $\frac{2V}{c}\frac{\partial C_L}{\partial q}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- $C_{L,M}$: $\frac{\partial C_L}{\partial M}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$.

The Model for drag coefficient is defined as follows

$$C_D = C_{D,0} + \frac{\left(C_L\left(\alpha, \dot{\alpha}, q, M, \delta_f, i_h, \delta_e\right) - C_{L,\min drag}\right)^2}{\pi A e} + C_{D,\delta_f}|\delta_f| + C_{D,i_h}|i_h| + C_{D,\delta_e}|\delta_e| + C_{D,\delta_a}|\delta_a| + C_{D,\delta_r}|\delta_r| + C_{D,M}(M - M_0)\,, \tag{22}$$

where

- $C_{D,0}$: Value of drag coefficient C_D, when α is such that $C_L = C_L\left(\alpha, \dot{\alpha}, q, M, \delta_f, i_h, \delta_e\right) = C_{L,\min drag}$, $\dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0, \delta_a = 0, \delta_r = 0$;
- $C_{Di} = \frac{\left(C_L\left(\alpha,\dot{\alpha},q,M,\delta_f,i_h,\delta_e\right) - C_{L,\min drag}\right)^2}{\pi A e}$: Induced drag coefficient.

- C_{D,δ_f}: $\frac{\partial C_L}{\partial \delta_f}$ when $\dot{\alpha} = 0$, $q = 0$, $M = M_0$, $\delta_f = 0$, $i_h = 0$, $\delta_e = 0$, $\delta_a = 0$, $\delta_r = 0$;
- C_{D,i_h}: $\frac{\partial C_L}{\partial |i_h|}$ when $\dot{\alpha} = 0$, $q = 0$, $M = M_0$, $\delta_f = 0$, $i_h = 0$, $\delta_e = 0$, $\delta_a = 0$, $\delta_r = 0$;
- C_{D,δ_e}: $\frac{\partial C_L}{\partial |\delta_e|}$ when $\dot{\alpha} = 0$, $q = 0$, $M = M_0$, $\delta_f = 0$, $i_h = 0$, $\delta_e = 0$, $\delta_a = 0$, $\delta_r = 0$;
- C_{D,δ_a}: $\frac{\partial C_L}{\partial |\delta_a|}$ when $\dot{\alpha} = 0$, $q = 0$, $M = M_0$, $\delta_f = 0$, $i_h = 0$, $\delta_e = 0$, $\delta_a = 0$, $\delta_r = 0$;
- C_{D,δ_r}: $\frac{\partial C_L}{\partial |\delta_r|}$ when $\dot{\alpha} = 0$, $q = 0$, $M = M_0$, $\delta_f = 0$, $i_h = 0$, $\delta_e = 0$, $\delta_a = 0$, $\delta_r = 0$;
- $C_{D,M}$: $\frac{\partial C_L}{\partial |M|}$ when $\dot{\alpha} = 0$, $q = 0$, $M = M_0$, $\delta_f = 0$, $i_h = 0$, $\delta_e = 0$, $\delta_a = 0$, $\delta_r = 0$.

Model for Crosswind Force coefficient is defined as follows

$$C_C = C_{c,\beta}\beta + C_{C,\delta_a}\delta_a + C_{C,\delta_r}\delta_r + \frac{b}{2V}\left(C_{C,p}p + C_{C,r}r\right), \tag{23}$$

where

- b: is the wing span;
- V: is the airspeed.
- $C_{c,\beta}$: $\frac{\partial C_C}{\partial \beta}$ when $\beta = 0$, $p = 0$, $r = 0$, $\delta_a = 0$, $\delta_r = 0$;
- C_{C,δ_a}: $\frac{\partial C_C}{\partial \delta_a}$ when $\beta = 0$, $p = 0$, $r = 0$, $\delta_a = 0$, $\delta_r = 0$;
- C_{C,δ_r}: $\frac{\partial C_C}{\partial \delta_r}$ when $\beta = 0$, $p = 0$, $r = 0$, $\delta_a = 0$, $\delta_r = 0$;
- $C_{C,p}$: $\frac{2V}{b}\frac{\partial C_C}{\partial r}$ when $\beta = 0$, $p = 0$, $r = 0$, $\delta_a = 0$, $\delta_r = 0$;
- $C_{C,r}$: $\frac{2V}{b}\frac{\partial C_C}{\partial r}$ when $\beta = 0$, $p = 0$, $r = 0$, $\delta_a = 0$, $\delta_r = 0$;

3.5 Aircraft Aerodynamic Moments

Figure 10, shows the aerodynamic moments acting on the aircraft. These aerodynamic moments are expressed in the body frame F_b as

$$M_{a,b} = \begin{bmatrix} l \\ m \\ n \end{bmatrix}, \tag{24}$$

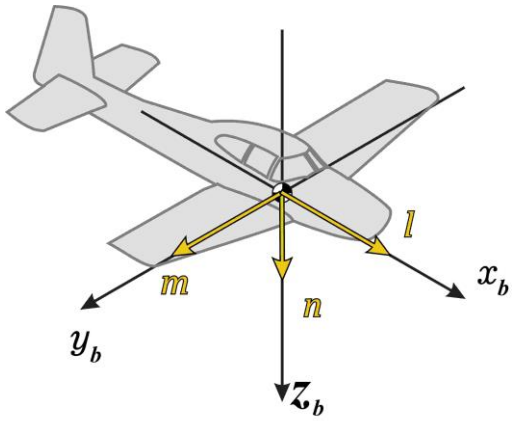

Figure 10. Moments action on the aircraft.

where l is the rolling moment, m the pitching moment and n the yawing moment. The three dimensionless moment coefficients are defined to represent the aerodynamic moments as follows

$$C_l = \frac{l}{\bar{q}Sb}, \tag{25}$$

$$C_m = \frac{m}{\bar{q}Sc}, \tag{26}$$

$$C_n = \frac{n}{\bar{q}Sb}, \tag{27}$$

where c is the mean aerodynamic chord and b is the wing span.

The rolling moment coefficient is modelled as follows

$$C_l = C_{l,\beta}\beta + C_{l,\delta_a}\delta_a + C_{l,\delta_r}\delta_r + \frac{b}{2V}\left(C_{l,p}p + C_{l,r}r\right), \tag{28}$$

where

- $C_{l,\beta}$: $\frac{\partial C_l}{\partial \beta}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;
- C_{l,δ_a}: $\frac{\partial C_l}{\partial \delta_a}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;
- C_{l,δ_r}: $\frac{\partial C_l}{\partial \delta_r}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;
- $C_{l,p}$: $\frac{2V}{b}\frac{\partial C_l}{\partial r}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;
- $C_{l,r}$: $\frac{2V}{b}\frac{\partial C_l}{\partial r}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;

The pitching moment coefficient is modelled as follows

$$\begin{aligned} C_m = {} & C_{m,0} + C_{m,\alpha}\alpha + C_{m,\delta_f}\delta_f + C_{m,i_h}\delta_{i_h} + C_{m,\delta_e}\delta_e \\ & + \frac{c}{2V}\left(C_{m,\dot{\alpha}}\dot{\alpha} + C_{m,q}q\right) + C_{m,M}(M - M_0), \end{aligned} \tag{29}$$

where

- M: Mach number. $M = \frac{V}{a}$ where a is the speed of sound;
- M_0: Mach number at a given flight condition;
- $C_{L,0}$: Value of lift coefficient C_L, when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- $C_{m,\alpha}$: $\frac{\partial C_m}{\partial \alpha}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- C_{m,δ_f}: $\frac{\partial C_m}{\partial \delta_f}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- C_{m,i_h}: $\frac{\partial C_m}{\partial i_h}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- C_{m,δ_e}: $\frac{\partial C_m}{\partial \delta_e}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- $C_{m,\dot{\alpha}}$: $\frac{2V}{c}\frac{\partial C_m}{\partial \dot{\alpha}}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- $C_{m,q}$: $\frac{2V}{c}\frac{\partial C_m}{\partial q}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;
- $C_{m,M}$: $\frac{\partial C_m}{\partial M}$ when $\alpha = 0, \dot{\alpha} = 0, q = 0, M = M_0, \delta_f = 0, i_h = 0, \delta_e = 0$;

The Yawing moment coefficient is modelled as follows:

$$C_n = C_{n,\beta}\beta + C_{n,\delta_a}\delta_a + C_{l,n}\delta_r + \frac{c}{2V}\left(C_{n,p}p + C_{n,r}r\right), \tag{30}$$

where

- $C_{n,\beta}$: $\frac{\partial C_n}{\partial \beta}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;
- C_{n,δ_a}: $\frac{\partial C_n}{\partial \delta_a}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;
- C_{n,δ_r}: $\frac{\partial C_n}{\partial \delta_r}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;
- $C_{n,p}$: $\frac{2V}{c}\frac{\partial C_n}{\partial r}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;
- $C_{n,r}$: $\frac{2V}{c}\frac{\partial C_n}{\partial r}$ when $\beta = 0, p = 0, r = 0, \delta_a = 0, \delta_r = 0$;

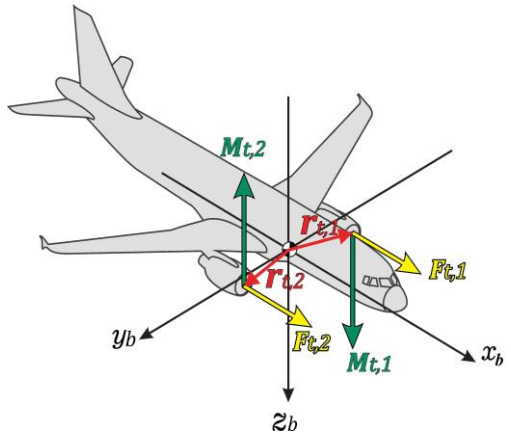

Figure 11. Propulsion forces and moments acting on the aircraft.

Figure 11, shows the propulsion forces and moments acting on the aircraft in a multi engine design. Each aircraft engine exerts a propulsion force $F_{t,i}$ at its airframe mounting location. The offset between each engine position and the aircraft center of gravity generates a propulsion force moment $M_{t,i}$. For n engines, the net propulsion force and moment will be

$$F_{t,b} = \sum_{i=1}^{n} F_{t,i} \, , \tag{31}$$

$$M_{t,b} = \sum_{i=1}^{n} M_{t,i} \, , \tag{32}$$

where

$$F_{t,i} = \begin{bmatrix} cos(\theta_{t,i})cos(\psi_{t,i}) \\ cos(\theta_{t,i})sin(\psi_{t,i}) \\ -sin(\theta_{t,i}) \end{bmatrix} F_{t_{max},i}\delta_t \, , \tag{33}$$

$$M_{t,i} = r_{t,i} \times F_{t,i} \, , \tag{34}$$

where

- $r_{t,i}$: Position of engine i expressed
- $\theta_{t,i}$: Engine toe up angle
- $\psi_{t,i}$: Engine toe up angle
- δ_t: Thrust control ($0 \leq \delta_t \leq 1$)
- $F_{t_{max},i}$: Maximum thrust force for engine i

CHAPTER 4. PROPOSED ADS TOLERANT TO SENSOR FAILURES

4.1 Introduction

It is clear from the discussion in chapter 1 that a Pitot or angle of attack vane sensor failure is critical for flight safety. This chapter will discuss the proposed architecture of an ADS capable of predicting the air data parameters even when the Pitot tube and angle of attack sensors are providing erroneous measurement signals. The proposed system detects the fault in a particular sensor (or sensors) and adjust its air data prediction algorithm accordingly. The sensor failure detection is useful to warn the aircraft pilot of the situation. Several sensors were used to generate true signals as inputs:

- IMU: An inertial measurement unit that contains a three-axis accelerometer and gyroscope sensors. These sensors will provide linear acceleration and angular rates measurements respectively. This device is vital for the fault tolerant ADS system to work.
- Magnetometer: Provides the measurement of the magnetic field vector expressed in F_e in a particular location on Earth. This measurement is used to compensate for heading.
- GPS: The Global Positioning System provides measurements of position and speed in F_e.

- ADS: The air data system features a Pitot tube, Angle of attack vane and angle of sideslip vane sensors. Measurements provided by these sensors are the airspeed V, Angle of attack α, and angle of sideslip β.
- Linear position transducers: These sensors are used to record the position of the control surfaces and the throttle. One sensor is necessary to be linked mechanically to each control surface (elevator, aileron, rudder and flap) and the engine throttle.

4.2 Fault Tolerant ADS Architecture

Figure 12 depicts a block diagram of the proposed ADS tolerant to sensor failures. The system is composed by the following components: a navigation system, an aircraft aerodynamic and thrust forces computation module, an aircraft digital twin, an air data estimation module, and a sensor fault detection module.

The navigation system uses the linear acceleration measurements in the body frame $a_{b,meas}$, angular rates in the body frame $\Omega_{b,meas}$ from the IMU's accelerometer and gyroscope respectively as vital signals to estimate position P_e, velocity V_e, attitude quaternion q, accelerometer bias b_a and gyroscope bias b_w. The bias values are used to obtain the linear acceleration a_b and the angular rate Ω_b quantities expressed in F_e. The navigation system also uses position $P_{e,meas(GPS)}$ and speed $V_{e,meas(GPS)}$ measurements expressed in F_e from the GPS and the derived yaw ψ_{meas} from the magnetic field measurement provided by the Magnetometer to increase the accuracy of the estimation.

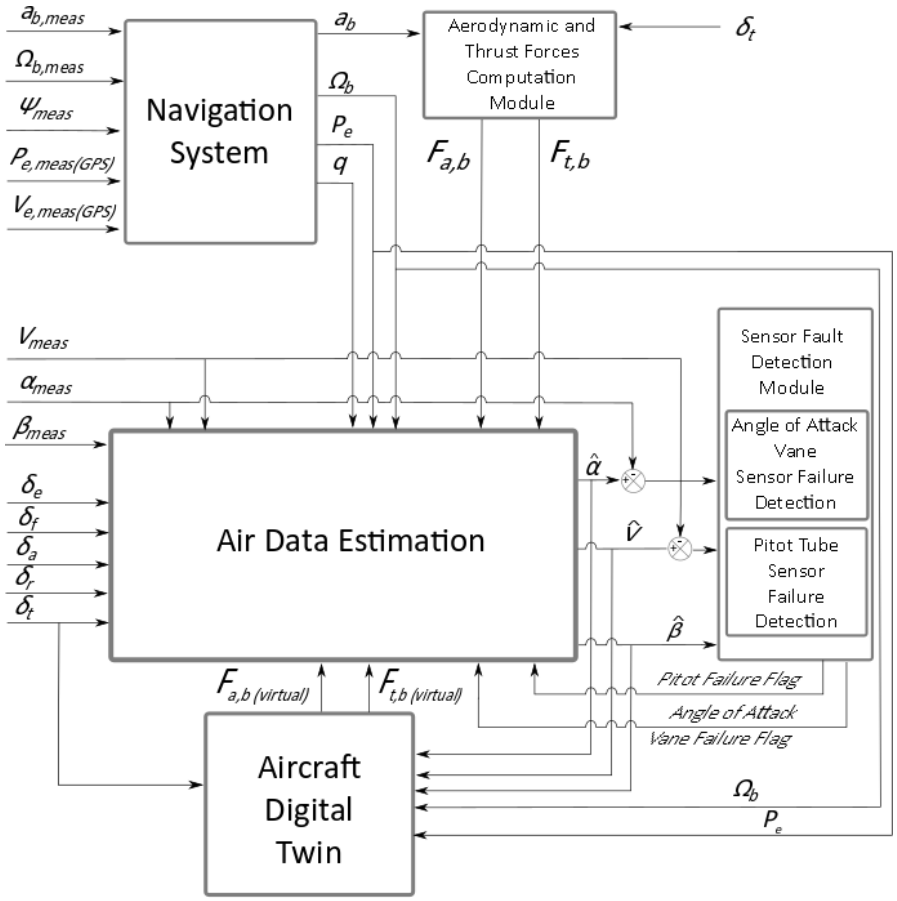

Figure 12. Air data estimation tolerant to sensor failures scheme.

The aerodynamic and thrust forces computation module uses the information of a_b provided by the navigation filter and the throttle control signal δ_t in conjunction with the aircraft an aircraft propulsion model, aircraft mass m, center of gravity position cg, and inertia I, to feed the aircraft's longitudinal equation of motion so the aerodynamic force $F_{a,b}$ and $F_{t,b}$ that the aircraft is being subjected during flight can be computed.

The digital twin block uses the estimated airspeed $\hat{V}$, estimated angle of attack $\hat{\alpha}$, estimated angle of sideslip $\hat{\beta}$, angular rates Ω_b, the position P_e and the trust control δ_t as inputs to infer from a fuzzy inference systems (FIS) the aircraft's virtual measurements of aerodynamic and propulsion forces acting on the aircraft center of gravity, $F_{a,b(virtual)}$ and

$F_{t,b(virtual)}$ respectively. The digital twin act as a virtual sensor that helps preserving the estimation accuracy of the air data variables during the Pitot and angle of attack sensor failures. The FIS is trained from flight test data.

The Air Data Estimation module uses Ω_b, P_e, q, airspeed pitot measurement V_{meas}, Angle of attack vane measurement α_{meas}, angle of sideslip vane measurement β_{meas}, $F_{a,b}$ and $F_{t,b}$, $F_{a,b(virtual)}$, $F_{t,b(virtual)}$, δ_t, δ_f, δ_e, δ_a, δ_r, to estimate the air data variables $\hat{V}$, $\hat{\alpha}$, $\hat{\beta}$. The Pitot tube and angle of attack residual signals are computed by subtracting V_{meas} from $\hat{V}$ and α_{meas} from $\hat{\alpha}$. The residual signals are analyzed by the sensor fault detection module where a Pitot failure flag or (and) an angle of attack vane failure flag is sent back to the air data estimator. The failure flags are used to modulate the Pitot and angle of attack vane sensor noise covariance matrix R, the process noise covariance matrix Q and the error covariance matrix P in the air data estimator algorithm in such a way the corrupted sensor signals from the unhealthy sensors are weighted less during the estimation process. It is worth pointing out a novelty introduced by this research where the sensors are always used in the estimation process independently of the occurrence of the failure and that there is always a permanent closed loop feedback between the estimator and the fault detection module. Now in the remaining sections of this chapter, each item in Figure 12 will be explained in detail.

4.3 Navigation System

The navigation system is in charge of estimating accurately the aircraft position, velocity and attitude based on the inertial sensor measurements of the IMU with the help

of a Magnetometer and a GPS as aiding sensors. This estimator is implemented based on an Extended Kalman Filter algorithm.

The nonlinear characteristic of the aircraft dynamic model gives an idea of the type of estimation algorithm to use. Three possible candidates are available, the Extended Kalman Filter - EKF, Unscented Kalman Filter - UKF and the Particle filter – PF. The EKF has the advantage of being the algorithm that required less computational cost; however, its downside is the need of computing Jacobian matrices to linearize the process model. The UKF has the advantage that there is no need to linearize the process model, however the calculation of sigma points requires considerable computational cost. The PF requires a large set of samples (particles) to represent the posterior distribution of the stochastic process which makes the algorithm computationally costly. Recent research by T. Nakamura and E.N. Johnson [54] shows that the EKF aircraft state estimation has the same level of accuracy as the UKF and PF but with the lower computational cost. For these reasons, the EKF algorithm is adopted in this research following the methodology presented in [55].

4.3.1 Inertial Navigation System Description

When combining IMU hardware, which includes a three-axis accelerometer and a rate gyroscope, with appropriate navigation algorithms, an inertial navigation system (INS) is obtained. The INS can be used to determine position, velocity and attitude of an aerospace vehicle by measuring the three axis-accelerations from the accelerometer and the three axis angular rates from rate gyroscope. The heart a navigation algorithm is the

navigation equation, which for local navigation in geodetic coordinates can be expressed as follows [56]:

$$\dot{V}_e = a_{cg,e} + G_e \,, \tag{35}$$

where $\dot{V}_e$ is the time derivative of the velocity vector V_e, with the North component of velocity V_N, East component of velocity V_E and the Down component of velocity V_D as expressed in equation (36).

$$V_e = [V_N \quad V_E \quad V_D]^T \tag{36}$$

The velocity components V_N, and V_E, are related with the change in latitude $\dot{\emptyset}_{lat}$, and the change in longitude $\dot{\ell}$ respectively as in equations (37) and (38). The velocity component V_D is related to the change in altitude $\dot{h}$ as in equation (39).

$$V_N = (M + h)\, \dot{\emptyset}_{lat} \,, \tag{37}$$

$$V_E = (N + h)\, cos\dot{\emptyset}_{lat}\, \dot{\ell} \,, \tag{38}$$

$$V_D = -\dot{h} \,, \tag{39}$$

The variables M and N are the geoid meridian radius of curvature and the prime vertical radius of curvature respectively and they are calculated by equations (40) and (41) as follows

$$M = \frac{a(1-e^2)}{(1-e^2 sin^2 \emptyset_{lat})^{3/2}}, \qquad \frac{b^2}{a} \leq M \leq \frac{a^2}{b} \tag{40}$$

$$N = \frac{a}{(1-e^2 sin^2 \emptyset_{lat})^{1/2}}, \qquad a \leq N \leq \frac{a^2}{b} \tag{41}$$

where a and b are the semimajor and semiminor axes of the generating ellipse of earth spheroid. The WGS-84 earth spheroid model defines a value of a as 6 378137 m and the value of b as 6356752 m. The eccentricity e is then computed by the equation (42)

$$e = \frac{(a^2 - b^2)^{\frac{1}{2}}}{a}. \tag{42}$$

The navigation equation can be integrated to obtain V_e. The position in geodetic coordinates, P_e, can be obtained by solving equations (37), (38) and (39) for $\dot{\emptyset}_{lat}$, $\dot{\ell}$ and $\dot{h}$ respectively and integrating each expression. Since the aircraft flights simulated in this research are performed in a relatively small area, the earth can be considered flat and its rotation neglected.

The accelerometer and rate gyroscope sensors are rigidly attached to the aircraft airframe (strapdown); thus, accelerometer measurements of specific forces are obtained in F_b, hence they need to be transformed to F_e in order to be used in the navigation equation as in

$$a_{cg,e} = {C_{b,e}}^T a_{cg,b} \tag{43}$$

then the navigation equation becomes

$$\dot{V}_e = {C_{b,e}}^T a_{cg,b} + G_e \ . \tag{44}$$

The direction cosine matrix $C_{e,b}$ is a time-varying quantity dependent of current attitude so, in order to determine attitude, it is necessary an attitude evolution equation expressed in quaternions, as in

$$\dot{q} = \frac{1}{2}\Omega(w_b)q \tag{45}$$

where Ω is given below.

Using the vehicle's angular rate vector $w_b(t)$ from the rate gyroscope and the initial quaternion vector $q(t=0)$, this attitude evolution equation can be time-integrated to get the quaternion vector or attitude at any time, $q(t)$. Combining the navigation equation, equation (35), expressed in state space form by using the vehicle velocity vector in equations (37), (38) and (39), and attitude evolution, the following ideal INS mechanization equations are obtained

$$\begin{bmatrix} \dot{\phi}_{lat} \\ \dot{\ell} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)cos\phi} & 0 \\ 0 & 0 & -1 \end{bmatrix} {C_{b/e}}^T V_b \ , \tag{46}$$

$$\dot{V}_e = {C_{b,e}}^T(q) a_{cg,b} + G_e \ , \tag{47}$$

$$\dot{q} = \frac{1}{2}\Omega(w_b)q \ , \tag{48}$$

where

$$C_{b,e}{}^{T} = \begin{bmatrix} 1-2({q_2}^2+{q_3}^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2({q_1}^3+{q_3}^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2({q_1}^2+{q_2}^2) \end{bmatrix}, \quad (49)$$

$$\Omega(w_b) = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}, \quad (50)$$

The magnitude of the gravity at the surface of the WGS-84 ellipsoid can be expressed in the form [57]:

$$G_e = g_0 \frac{1+g_1 sin^2 \emptyset_{lat}}{(1-e^2 sin^2 \emptyset_{lat})^2}, \quad (51)$$

where, $g_0 = 9.7803267714\ m/s^2$ is the gravity at the equator, $g_1 = 0.00193185138639\ m/s^2$ is the gravity formula constant and $e = 0.0818191908426$ is the first eccentricity. The states of the ideal INS mechanization equations include three components of position, three components of velocity, and four components of the attitude quaternion. IMU is considered to provide three accelerations from the accelerometer and three angular rates from the rate gyroscope.

4.3.2 Continuous Process Model of INS Navigation

The ideal INS equations are derived based on the assumption that the IMU is installed at the aircraft center of gravity and hence the acceleration measurements are at the center of gravity. For the general the case where the IMU is positioned at a distance $r_{imu,b}$ relative to the vehicle's center of gravity, the acceleration vector a_b can be calculated from the following relation:

$$a_b = a_{cg,b} + \Delta a_{imu,b}, \tag{52}$$

where,

$$\Delta a_{imu,b} = \dot{w}_b \times r_{imu,b} + w_b \times \left(w_b \times r_{imu,b}\right), \tag{53}$$

Here, $\Delta a_{imu,b}$ is the acceleration effect due to the IMU offset from the vehicle's center of gravity location, $r_{imu,b}$. Since the rate gyroscope is rigidly fixed to the vehicle, angular rates at IMU position are the same as those at vehicle's center of gravity.

The IMU sensor measurements are corrupted by various types of errors such as scale factors, misalignments, biases, and random noises [58]. Depending on the type of sensors used and the navigation accuracy required, sensor models can be detailed to account for these phenomena. For the type of IMU used in this research it suffices to consider that true values are perturbed by two effects, a bias and a measurement noise in both the acceleration and angular rate measurements [59].

$$a_{b,meas} = a_b + b_a + n_a\ , \tag{54}$$

$$w_{b,meas} = w_b + b_w + n_w\ , \tag{55}$$

or

$$a_b = a_{b,meas} - b_a - n_a = \bar{a}_b - n_a\ , \tag{56}$$

$$w_b = w_{b,meas} - b_w - n_w = \bar{w}_b - n_w\ , \tag{57}$$

Where $\bar{a}_b = a_{b,meas} - b_a$ and $\bar{w}_b = w_{b,meas} - b_w$ are bias corrected acceleration and angular rate vectors, respectively. a_b and w_b are true acceleration and angular rates of the vehicle. $a_{b,meas}$ and $w_{b,meas}$ are measured acceleration and angular rates from the IMU. n_a and n_w are the IMU acceleration and gyro rate measurement noise terms which are assumed to be zero-mean, white Gaussian noises.

By substituting equations (54) to (57) into equation (49), the velocity navigation equation becomes

$$\begin{aligned} \dot{V}_e &= C_{b,e}{}^T(q)a_{cg,b} + G_e = C_{b,e}{}^T(q)\left(a_b - \Delta a_{imu,b}\right) + G_e \\ &= C_{b,e}{}^T(q)\left(\bar{a}_b - \Delta\bar{a}_{imu,b}\right) + G_e - C_{b,e}{}^T(q)n_a \,, \end{aligned} \tag{58}$$

where

$$\Delta\bar{a}_{imu,b} \cong \dot{\bar{w}}_b \times r_{imu,b} + \bar{w}_b \times \left(\tilde{w}_b \times r_{imu,b}\right). \tag{59}$$

For the approximation of acceleration correction, $\Delta\bar{a}_{imu,b}$, of IMU position offset from center of gravity, bias corrected angular rate, $\bar{w}_b$, and bias corrected angular acceleration, $\dot{\bar{w}}_b$, are used.

By substituting equation (144) and (57) into equation (136), the attitude equation becomes

$$\dot{q} = \tfrac{1}{2}\Omega(w_b)q = \tfrac{1}{2}\Omega(\bar{w}_b)q - \tfrac{1}{2}Z(q)n_w \,. \tag{60}$$

In order to provide more fidelity on the accelerometer and rate gyro error models, time varying dynamics for acceleration biases and rate gyro biases are introduced. The biases are modeled as random walks with zero mean Gaussian driving terms in both the acceleration and rate gyroscope measurements.

$$\dot{b}_a = n_{b_a}, \tag{61}$$

$$\dot{b}_w = n_{b_w}, \tag{62}$$

where $n_{b_a}$and $n_{b_w}$are accelerometer and rate gyro bias noises which are considered to be zero-mean, white Gaussian noises.

In this research a low-cost MEMS-based IMU is used. The accelerations and angular rates for this type of sensor usually include large bias and scale factor errors. The scale factor effect can be considered as extra independent states [60], or it can be included in the time-varying bias term as commonly used [61]. Because of the time varying bias effects, additional six states, three for acceleration bias errors and three for rate gyro bias errors, are needed for augmenting the state vector. Now, gathering the position navigation (46), the velocity navigation equation (58) the attitude equation (60), the acceleration bias equation (61) and the rate gyroscope bias equation (62), the following INS navigation process model in continue-time state-space form is obtained:

$$\dot{x}(t) = f(x(t), u(t), t) + G(x(t), t)w(t), \tag{63}$$

or

$$\begin{bmatrix} \dot{P}_e \\ \dot{V}_e \\ \dot{q} \\ \dot{b}_a \\ \dot{b}_w \end{bmatrix} = \begin{bmatrix} C_{e,b}V_b \\ C_{e,b}a_{cg,b} + G_e \\ \frac{1}{2}\bar{\Omega}_b q \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix} + \begin{bmatrix} I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -C_{e,b} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\frac{1}{2}Z(q) & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix} \begin{bmatrix} n_r \\ n_a \\ n_w \\ n_{ba} \\ n_{bw} \end{bmatrix}, \quad (64)$$

where

$$\Delta\bar{a}_{imu,b} \cong \dot{\bar{w}}_b \times r_{imu,b} + \bar{w}_b \times \left(\tilde{w}_b \times r_{imu,b}\right)$$

$$= \begin{bmatrix} \dot{\bar{q}}z_a - \dot{\bar{r}}y_a & \bar{p}(\bar{q}y_a - \bar{r}z_a) & -x_a(\bar{q}^2 + \bar{r}^2) \\ \dot{\bar{r}}x_a - \dot{\bar{p}}z_a & \bar{q}(\bar{p}x_a - \bar{r}z_a) & -y_a(\bar{p}^2 + \bar{r}^2) \\ \dot{\bar{p}}y_a - \dot{q}x_a & \bar{r}(\bar{p}x_a - \bar{q}y_a) & -z_a(\bar{p}^2 + \bar{q}^2) \end{bmatrix}, \quad (65)$$

$$\bar{a}_b = a_{b,meas} - b_a = [a_1\ a_2\ a_3]^T\ , \quad (66)$$

$$\bar{w}_b = w_{b,meas} - b_w = [p\ q\ r]^T\ , \quad (67)$$

The state vector of the process model in the INS process model includes three position components, three velocity components, four quaternion components, three acceleration biases, and three angular rate biases:

$$x = \begin{bmatrix} P_e \\ V_e \\ q_e \\ b_a \\ b_w \end{bmatrix}, \quad (68)$$

where

$$P_e = \begin{bmatrix} \emptyset_{lat} \\ \ell \\ h \end{bmatrix}, V_e = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, b_a = \begin{bmatrix} a_{x,b} \\ a_{y,b} \\ a_{z,b} \end{bmatrix}, b_w = \begin{bmatrix} w_{x,b} \\ w_{y,b} \\ w_{z,b} \end{bmatrix}, \quad (69)$$

here P_e and V_e are the position and velocity vector of the vehicle in geodetic coordinates respectively. The variable q represents the quaternion vector that expresses the vehicle attitude. b_a and b_w are IMU acceleration bias vector and IMU rate gyroscope bias vector, respectively. n_r is a fictitious zero-mean, white noise associated to the position navigation equation.

In order to apply EKF to the continuous-time process model, it is necessary to calculate the following Jacobian matrix from the nonlinear system in equations (63) to (67):

$$F_k = \frac{\partial f}{\partial x}|_{x=x_k} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \left(\frac{\partial f_v}{\partial q}\right)_{3\times4} & \left(\frac{\partial f_q}{\partial b_a}\right)_{3\times4} & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} & \left(\frac{\partial f_q}{\partial q}\right)_{4\times4} & 0_{4\times3} & \left(\frac{\partial f_q}{\partial b_w}\right)_{4\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}_k, \quad (70)$$

where

$$\left(\frac{\partial f_v}{\partial q}\right)_{3\times4} = \begin{bmatrix} 2(-q_3\bar{a}_2 + q_2\bar{a}_3) & 2(q_2\bar{a}_2 + q_3\bar{a}_3) & 2(-2q_2\bar{a}_1 + q_1\bar{a}_2 + q_0\bar{a}_3) & 2(-2q_3\bar{a}_1 - q_0\bar{a}_2 + q_1\bar{a}_3) \\ 2(q_3\bar{a}_1 - q_1\bar{a}_3) & 2(q_2\bar{a}_1 - 2q_1\bar{a}_2 - q_0\bar{a}_3) & 2(q_1\bar{a}_1 + q_3\bar{a}_3) & 2(q_0\bar{a}_1 - 2q_3\bar{a}_2 + q_2\bar{a}_3) \\ 2(q_2\bar{a}_1 + q_1\bar{a}_2) & 2(q_3\bar{a}_1 + q_0\bar{a}_2 - 2q_1\bar{a}_3) & 2(-q_0\bar{a}_1 + q_3\bar{a}_2 - 2q_2\bar{a}_3) & 2(q_1\bar{a}_1 + q_2\bar{a}_2) \end{bmatrix}, \quad (71)$$

$$\left(\frac{\partial f_v}{\partial b_a}\right)_{3\times3} = -C_{n,b}(q)\,, \tag{72}$$

$$\left(\frac{\partial f_q}{\partial q}\right)_{4\times4} = \frac{1}{2}\Omega(\bar{\omega}_b)\,, \tag{73}$$

$$\left(\frac{\partial f_q}{\partial b_w}\right)_{4\times4} = -\frac{1}{2}Z(q)\,, \tag{74}$$

The IMU offset effect from vehicle center of gravity, $\Delta\bar{a}_{imu}^{b}$, and gravity change due to vehicle position, g^n, are neglected in the calculation of the Jacobian matrix, F_k.

4.3.3 Discrete Process Model of INS Navigation

In order to apply the EKF filter, it is necessary to write the process equations in discrete time. This is carried out by expressing system in equations (63) to (67) by the following first-order Euler integration algorithm:

$$r_{e,k+1} = P_{e,k} + \dot{r}_{e,k}\Delta t = r_{e,k} + V_{e,k}\Delta t\,, \tag{75}$$

$$V_{e,k+1} = V_{e,k} + \dot{V}_{e,k}\Delta t$$
$$= V_{e,k} + \left[{C_{b,e}}^{T}(q_k)\left(\bar{a}_{b,k} - \Delta\bar{a}_{imu,b,k}\right) + G_e\right]\Delta t \quad (76)$$
$$- \Delta t {C_{b,e}}^{T}(q_k) n_{a,k} ,$$

$$q_{k+1} = q^{\frac{1}{2}\Omega\left(w_{b,k}\right)q\Delta t} q \cong \left[I_{4\times4} + \tfrac{1}{2}\Omega\left(w_{b,k}\right)\Delta t\right] q_k$$
$$= \left[I_{4\times4} + \tfrac{1}{2}\Omega\left(\overline{w}_{b,k} - n_{w,k}\right)\Delta t\right] q_k , \quad (77)$$
$$= \left[I_{4\times4} + \tfrac{1}{2}\Omega\left(\overline{w}_{b,k}\right)\Delta t\right] q_k - \frac{\Delta t}{2}\Omega\left(n_{w,k}\right) q_k ,$$

$$b_{a,k+1} = b_{a,k} + n_{b_a,k}\Delta t \ , \quad (78)$$

$$b_{w,k+1} = b_{w,k} + n_{w_a,k}\Delta t \, . \quad (79)$$

If the above equations are written in matrix form separating the noise terms, the following approximate discrete nonlinear equation for INS process model is obtained:

$$x_{k+1} = f_d(x_k, k) + G_k w_k , \quad (80)$$

or

$$\begin{bmatrix} r_{e,k+1} \\ V_{e,k+1} \\ q_{k+1} \\ b_{a,k+1} \\ b_{w,k+1} \end{bmatrix}$$

$$= \begin{bmatrix} r_{e,k} + V_{e,k}\Delta t \\ V_{e,k} + \left[{C_{b,e}}^{T}(q_k)\left(\bar{a}_{b,k} - \Delta\bar{a}_{imu,b,k}\right) + G_e\right]\Delta t \\ \left[I_{4\times4} + \frac{1}{2}\Omega\left(\bar{w}_{b,k}\right)\Delta t\right] q_k \\ b_{a,k} \\ b_{w,k} \end{bmatrix} \tag{81}$$

$$+ \begin{bmatrix} \Delta t\ I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -\Delta t\ {C_{b,e}}^{T} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\frac{\Delta t}{2}\Omega\left(n_{w,k}\right)q_k & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \Delta t\ I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \Delta t\ I_{3\times3} \end{bmatrix} \begin{bmatrix} n_{r,k} \\ n_{a,k} \\ n_{w,k} \\ b_{a,k} \\ b_{w,k} \end{bmatrix}.$$

where

$\bar{a}_{b,k} = a_{b,k} - b_{a,k}$ and $\bar{w}_{b,k} = w_{b,k} - b_{w,k}$ are bias corrected IMU accelerometer and rate gyro measurements.

The Matlab/Simulink® linearization of the process can be carried out numerically by implementing the Jacobian matrix computation using finite differences. This method was used in this research.

4.3.4 INS Measurement Model

In order to compensate the data degradation of the small and low-cost IMU with time, the integrated navigation system combines a low-cost IMU with aiding sensors. In

this research and Xsens MTi G 700 sensor suite is used as measurement hardware, which includes an IMU that measures three axis acceleration and angular rates, a GPS receiver that measures inertial position and velocity and a magnetometer which gives heading information. The former three sensors acting as aiding sensors. Sensor update rates are summarized in Table 2 below.

Table 2 Sensor update rates.

Sensor	Update rate	Measurements
IMU	100 Hz	vehicle acceleration & angular rates
GPS	4 Hz	inertial position & velocity
Magnetometer	50 Hz	heading information

Using the high frequency vehicle accelerations and angular rates, inertial navigation mechanization algorithm can provide high rate measurements of vehicle position/velocity and attitude. Because of IMU measurement errors, errors in the estimates of vehicle position/velocity and attitude grow with time. In order to compensate these long-term errors, aiding sensors such as GPS, Magnetometer, altimeter, and so on are generally equipped in addition to IMU. These aiding sensors measure data in relatively low frequency but obtain time independent accurate data. IMU can provide complete navigation information such as vehicle position/velocity and attitude, while IMU-measured accelerations and angular rates need to be time-integrated to get this navigation

information. On the other hand, aiding sensors usually provide only part of navigation information, but it can be directly used to get navigation information without involvement of dynamic equations.

Since IMU measurement rate is relatively higher than other aiding sensors, IMU measurements are considered to be continuous data flow. IMU measured vehicle accelerations and angular rates are treated as the inputs to the process model.

Since aiding sensors have several different update rates, it is necessary to carefully treat measurement updates in Kalman filtering framework. Due to the variation in the number of available measurements depending on time instant, the dimensions of measurement vector and Kalman gain matrix are highly varying. In order to easily deal with this multi-rate sensor fusion problem, "sequential processing of measurement updates" method is applied [62], [63]. Measurement updates are not considered in a whole and in a big measurement matrix/vector, but rather each measurement is treated separately, sequentially, and in small-size several vectors/matrices. The addition of new aiding sensors is easier in this framework than in the standard Kalman filtering framework. Keeping this argument in mind, separate measurement model for each aiding sensor are described in next sections.

4.3.5 GPS Position and Velocity Measurement Model

Since the position and velocity measurements of the INS are degraded in time, GPS receiver is augmented to frequently update the vehicle position and velocity in navigation frame and to correct the long-term INS errors. The selected GPS receiver provides position and velocity information at a slower rate compared to INS system. Since the GPS antenna can be mounted off the vehicle's center of gravity, GPS sensor can measure the position and velocity at the GPS mounting location with respect to the center of gravity, $r_{b,gps}$. Furthermore, GPS measurements have a latency and need to be compensated. Considering this GPS latency, current position/velocity measurements are actually previous position/velocity and current updates are based on older state estimate corresponding to this latency. The integrated INS, GPS position and velocity have different latency and their measurement updates are applied independently.

$$y_k^1 = h^1(x_k) + n_{r,k}^{gps} \Leftrightarrow r_k^{gps} = r_{e,k-L_1} + {C_{b,e}}^T(q_{k-L1}) r_{b,gps} + n_{r,k}^{gps}\,, \tag{82}$$

$$\begin{aligned} y_k^2 = h^2(x_k) + n_{v,k}^{gps} &\Leftrightarrow v_k^{gps} \\ &= v_{e,k-L_2} + {C_{b,e}}^T(q_{k-L2}) \bar{w}_{k-L_2} \times r_{b,gps} + n_{v,k}^{gps}\,, \end{aligned} \tag{83}$$

where $r_{e,k-L_1}$ and $v_{e,k-L_2}$ are the time-delayed vehicle position and vehicle velocity vector in the navigation frame. $\bar{w}_{k-L_2}$ is the time-delayed vehicle angular rate vector in the body frame. Time delay comes from the GPS sensor latency and $L_1 = \frac{GPS\ position\ Latency}{\Delta t}$ and $L_2 = \frac{GPS\ velocity\ Latency}{\Delta t}$. The variable $r_{b,gps}$ is the location of the GPS antenna relative to vehicle's center of gravity location in the body frame.

4.3.6 Magnetometer Measurement Model

Three axes magnetometer measures Earth magnetic field vector in body frame. The measured Earth magnetic field vector is used to correct for heading compensation. The basic idea for yaw angle compensation is as follows. The residual quantity in the measurement of declination angle will be the same as the residual of yaw angle. Hence it is first computed the residual value of declination angle instead of that of yaw angle, and then this residual (difference between ideal declination angle given by world magnetic model and measured declination angle) is used to compensate the yaw angle in the framework of extended Kalman filter. The following measurement model is used for yaw angle measurement:

$$\begin{aligned} y_k^3 = h^3(x_k) + n_\psi \;\Leftrightarrow\; \psi &= atan2(C_{21}, C_{11}) + n_\psi \\ &= atan2[2(q_1q_2 + q_0q_3), q_0^2 + q_1^2 - q_2^2 - q_3^2\ ] + n_\psi\,, \end{aligned} \tag{84}$$

where ψ is the true heading angle and C_{ij} are the corresponding components in rotation matrix $C_{e,b}$ expressed in (9).

Here again The Matlab/Simulink linearization of the measurement models was carried out numerically by implementing the Jacobian matrix computation using finite differences.

4.3.7 Integrated INS Navigation Using EKF

In order to fuse aiding sensor measurements with INS navigation algorithm, Extended Kalman Filter (EKF) is used. This section describes the sensor fusion architecture and the details of filter implementation. Figure 13 shows the overall integrated strapdown

INS mechanization. Here only bias effects are considered to be independent states and are used to correct IMU sensor drifts.

The INS outputs are used as a reference trajectory and attitude. Aiding sensor measurements such as vehicle inertial position and velocity from GPS and heading information from three-axis magnetometer are applied to update the states of these trajectory and attitude, and hence limit the long-term error growth with time.

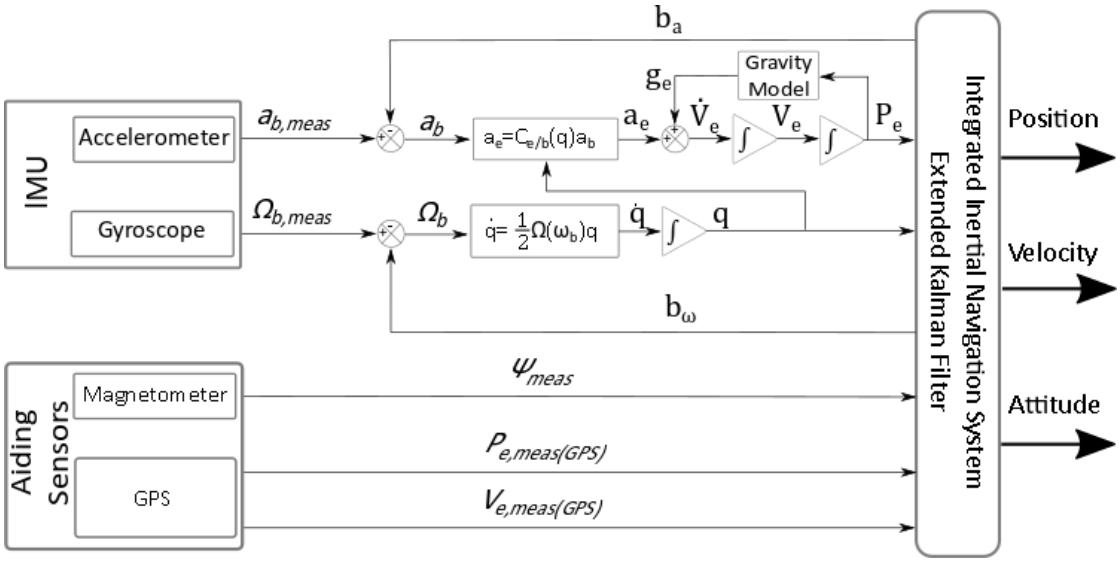

Figure 13. EKF navigation algorithm.

4.3.8 Extended Kalman Filter with Sequential Measurement Update

Now the EKF will be applied to the continuous and discrete process model discussed above. As explained in the previous section, measurement models are expressed in separate form for each sensor in order to apply the "Sequential processing of multi-rate measurements" [64].

The general nonlinear continuous-time process model and discrete-time measurement model in state-space form are given by

$$\dot{x}(t) = f(x(t), u(t), t) + G(x(t), t)w(t), \tag{85}$$

and

$$w(t) \sim N\big(0, Q(t)\big), \tag{86}$$

$$y_k^l = h^l(x_k, k) + v_k^l, v_k^l \sim N\big(0, R_k^l\big), \quad l = 1,2, \dots, r, \tag{87}$$

where r is the number of aiding sensors.

The initial conditions $\big(x(t_0) \sim N(\hat{x}_0, P_0)\big)$ are assumed to be known and have the following form:

$$\hat{x}_0 = E[x(t_0)], \tag{88}$$

$$P_0 = E[(x(t_0) - \hat{x}_0)(x(t_0) - \hat{x}_0)^T]. \tag{89}$$

Jacobian matrices of system dynamics and Measurement model are defined as

$$F(\hat{x}(t), t) = \frac{\partial F}{\partial x}\Big|_{x(t)=\hat{x}(t)}, \tag{90}$$

$$H_k^l = \frac{\partial h^l(x,k)}{\partial x}\Big|_{x=\hat{x}_k^- = \hat{x}(t_k^-)}, \qquad l = 1,2, \dots, r, \tag{91}$$

The time update is performed as follows. Given the state estimate $\hat{x}_{k-1} = \hat{x}(t_{k-1})$ and error covariance matrix $P_{k-1} = P(t_{k-1})$ at time $t = t_{k-1}$, current state estimate $\hat{x}_k^- =$

$\hat{x}(t_k^-)$ and error covariance matrix $\hat{P}_k^- = \hat{P}(t_k^-)$ can be obtained by integrating forward from $t = t_{k-1}$ to $t = t_k^-$ $(k = 1,2,\cdots)$ Using the following state estimate propagation equation and error covariance propagation equation:

$$\dot{\hat{x}}(t) = f(\hat{x}(t), u(t), t), \tag{92}$$

$$\dot{P}(t) = F(\hat{x}(t), t)P(t) + P(t)F^T(\hat{x}(t), t) + G(x(t), t)Q(t)G^T(x(t), t). \tag{93}$$

The sequential measurement update is performed as follows. Given the time updated state and error covariance matrix $\hat{x}_k^-$ and $\hat{P}_k^-$ with new measurement vector $y_k = y(t_k) = [y^1(t_k)^T \cdots y^r(t_k)^T]^T$, we apply the sequential processing of measurement update in order to obtain the measurement updated state and error covariance matrix $\hat{x}_k = \hat{x}(t_k)$, $P_k = P(t_k)$. For each available measurement $y_k^l = y^l(t_k)$ $(y = 1, \cdots, l)$ at time instant $t = t_k$, the state estimate $\hat{x}_k$ and error covariance matrix P_k can be updated by the following error sequential measurement processing.

For $l = 1,2, \ldots, r$ (r measurements updates at time $t = t_k$),

$$K_k^l = P_k^{l-1} {H_k^l}^T (\hat{x}_k^{l-1}) \left[H_k^l (\hat{x}_k^{l-1}) P_k^{l-1} {H_k^l}^T (\hat{x}_k^{l-1}) + R_k^l \right]^{-1}, \tag{94}$$

$$\hat{x}_k^l = \hat{x}_k^{l-1} + K_k^l \left[y_k^l + h^l (\hat{x}_k^{l-1}) \right], \tag{95}$$

$$P_k^l = \left[I - K_k^l H_k^l \right] P_k^{l-1} \left[I - K_k^l H_k^l \right]^T + K_k^l R_k^l {K_k^l}^T, \tag{96}$$

$$\left(P_k^l = \left[I - K_k^l H_k^l \right] P_k^{l-1} \right), \qquad l = 1,2, \ldots, r, \tag{97}$$

where starting initial conditions for this sequential measurement update at $t = t_k^-$ is $\hat{x}_k^0 = \hat{x}_k^- = \hat{x}(t_k^-)$, $P_k^0 = P_k^- = P(t_k^-)$ and final measurement update is set to $\hat{x}_k = \hat{x}(t_k) = \hat{x}_k^r$, $P_k = P(t_k) = P_k^r$.

For any measurement not available at time $t = t_k$, the correspondent update step is omitted. Whenever any measurement is available at time $t = t_k$, that measurement can be included for this sequential measurement update processing.

4.3.9 GPS Measurement Update with Sensor Latency Compensation

In equation (194) it is necessary to use proper state vector for the calculation of nonlinear measurement model in case there is sensor latency. Since GPS position and velocity measurement model need to use latency compensated state vector (i.e., previous states depending on correspondent latency), measurement estimate components $\hat{y}_k^l$, $(l = 1,2)$ corresponding to GPS position and velocity are computed by used latency compensated states.

$$\hat{x}_k^l = \hat{x}_k^{l-1} + K_k^l\left[y_k^l - \hat{y}_k^l\right], \qquad l = 1,2 \tag{98}$$

where

$$\hat{y}_k^l = h^l\left(\hat{x}_{k-L}^l\right), \tag{99}$$

and

$$L = \frac{Aiding\ sensor\ latency}{\Delta t} \tag{100}$$

are time delayed steps corresponding to sensor latency ($L = L_1$ for GPS position latency and $L = L_2$ for GPS velocity latency).

4.3.10 Magnetometer Measurement Update

Three axis magnetometer measures earth magnetic field line in body frame. The measured earth magnetic field is used to compensate for vehicle heading. Yaw angle measurement update is based on the assumption that the residual in the measurement of the declination angle, $\Delta\psi_d$, will be the same as the residual of yaw angle, $\Delta\psi$. Then it is possible to use the residual value of declination angle for the compensation of yaw angle in the extended Kalman filter framework as follows:

$$\hat{x}_k^l = \hat{x}_k^{l-1} + K_k^l\left[y_k^l - \hat{y}_k^l\right], \qquad l = 3 \tag{101}$$

where

$$\Delta\psi = \Delta\psi_d = y_k^l - \hat{y}_k^l, \tag{102}$$

$$y_k^l = \psi_{d,measure} = atan2\left(h_x^b, h_y^b\right), \tag{103}$$

$$\hat{y}_k^l = \psi_{d,estimate}. \tag{104}$$

Here, ψ_d is local variation or declination angle that is the angle between the true north and magnetic north, and $\psi_{d,estimate}$, can be obtained by a recent world magnetic

model, for example 2020-2025 epoch (WMM 2020) [65]. However, if flights are performed in a relatively small area a constant magnetic field and its declination can be used for simulations without sacrificing accuracy. In addition, $h^b = \left[h_x^b\ h_y^b\ h_z^b\right]^T$ is the magnetic field triad in the North-East-Down (NED) local earth frame. This is obtained by projecting the magnetic field vector on the local NED frame using the last vehicle attitude estimate $\hat{q}$, to eliminate the effects of magnetic dip. Since $h^n = C_{n,b}h^b$ and $h^b = \left[h_x^b\ h_y^b\ h_z^b\right]^T$ are measured from magnetometer triad, the following relations are obtained:

$$h_x^b = h_x^b C_{11} + h_y^b C_{12} + h_z^b C_{13}, \tag{105}$$

$$h_y^b = h_x^b C_{21} + h_y^b C_{22} + h_z^b C_{23}, \tag{106}$$

where C_{ij} are corresponding components in rotation matrix $C_{n,b}$ expressed in equation (136).

4.3.11 Implementation

In order to reduce computational cost, the following noise definitions are usually introduced:

$$n_v \triangleq -{C_{b,e}}^T(\mathrm{q})n_a\ , \tag{107}$$

$$n_q \triangleq -\frac{1}{2}\mathrm{Z}(\mathrm{q})n_w\ . \tag{108}$$

then the INS navigation process model in continue-time state-space form becomes as the following rather simple form:

$$\dot{x}(t) = x(t) + w_1(t)\,, \tag{109}$$

or

$$\begin{bmatrix} \dot{P}_e \\ \dot{V}_e \\ \dot{q} \\ \dot{b}_a \\ \dot{b}_b \end{bmatrix} = \begin{bmatrix} C_{e,b} V_b \\ C_{b,e}{}^T(q)(\bar{a}_b - \Delta\bar{a}_{imu}) + G_e \\ \frac{1}{2}\Omega(w_b)q \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix} + \begin{bmatrix} n_r \\ n_v \\ n_q \\ n_{b_a} \\ n_{b_w} \end{bmatrix}, \tag{110}$$

Expressing the INS process model in equation (64) is more intuitive as it is represented by direct sensor noises (n_a, n_w). The random noise characteristics of accelerometers and rate gyros (n_a, n_w) are relatively easily modeled from sensor characteristics and sensor test outputs. On the other hand, the INS process model expressed in equation (110) involves difficulty in modeling the noise characteristics of n_v and n_q while it has simpler form and thus is computationally more efficient. By using this process model expression, the error covariance propagation equation becomes

$$\dot{P}(t) = f(\hat{x}(t), t)P(t) + P(t)F^T(\hat{x}(t), t) + Q(t)\,, \tag{111}$$

where $Q = E[w_1(t)w_1(t)^T]$ is 16x16 matrix. Comparing the original error covariance propagation equation,

$$\begin{aligned} \dot{P}(t) = f(\hat{x}(t), t)P(t) + P(t)F^T(\hat{x}(t), t) + Q(t) \\ + G(\hat{x}(t), t)Q(t)G^T(\hat{x}(t), t)\,, \end{aligned} \tag{112}$$

where $Q = E[\mathrm{w}(t)\mathrm{w}(t)^T]$ is 15x15 matrix, the process model in equation (110) contributes in the computational efficiency of navigation filter since it removes 16x15 big matrix $G(x)$ multiplications. As n_v is just accelerometer noise in navigation frame and $det\left({C_{b,e}}^T\right) = 1$ in equation (107), statistical characteristics of noise n_v can be modeled to have the same statistical characteristics of noise n_q. The noise n_q in $\dot{\mathrm{q}}$ equation is more difficult to characterize. Since Z matrix is composed of quaternions that have values lower than 1, the statistical characteristics of noise n_q can be approximately determined based on those of noise n_w and have been tuned through flight tests. The process noise covariance matrix Q in current filter is

$$\begin{aligned}\mathbf{Q} &= diag\left[Q_{n_r}, Q_{n_v}, Q_{n_q}, Q_{n_{b_a}}, Q_{n_{b_w}}\right] \\ &= diag[0.0\ (ft/s)^2, 0.0\ (ft/s)^2, 0.0\ (ft/s)^2, 0.01\ (ft/s^2)^2, 0.01\ (ft/s^2)^2, 0.01\ (ft/s^2)^2, 0.001, 0.001, 0.001, 0.001, \\ &\quad 0.001\ (ft/s^2)^2, 0.001\ (ft/s^2)^2, 0.001\ (ft/s^2)^2, \quad 0.00001\ (ft/s)^2, 0.00001\ (ft/s)^2, 0.00001\ (ft/s)^2\ ].\end{aligned} \tag{113}$$

The measurement noise covariance matrix of GPS position is $R_{gps,pos} = diag[25^2\ 25^2\ 30^2]\ ft^2$. Similarly, the measurement noise covariance matrix of GPS velocity is $R_{gps,Vel} = diag[10^2\ 10^2\ 10^2]\ ft^2$. Finally, the measurement noise covariance matrix of magnetometer is $R_{mag} = 10^2\ deg^2$.

In the implementation of time update in equations (92) and (93), state estimate is integrated with Runge-Kutta second-order algorithm and error covariance matrix is updated using first-order Euler integration algorithm:

$$\hat{x}_k^- = \int_{t_{k-1}}^{t_k^-} f(\hat{x}(t), u(t), t)dt\ , \tag{114}$$

$$P_k^- = P_{k-1} + \Delta t[f(\hat{x}(t_{k-1}), t_{k-1})P(t_{k-1}) + P(t_{k-1})F^T(\hat{x}(t_{k-1}), t) + Q(t_{k-1})]\,. \tag{115}$$

The sequential measurement update method has an advantage. Aiding sensors in INS mechanization are usually not correlated each other and hence sequential measurement update is possible. In this case, measurement update is computationally efficient since series of smaller matrix inversion is involved instead of one big matrix inversion in equation (96). Since aiding sensors have several different update rates, it is necessary to carefully treat measurement updates in Kalman filtering framework. Due to the variation in the number of available measurements depending on time instant, the dimensions of measurement vector and Kalman gain matrix are also varying. Sequential measurement update approach is easier to deal with this multi-rate sensor fusion problem. Furthermore, the addition of new aiding sensors in current design is easier in this framework than in the standard Kalman filtering framework.

4.4 Aircraft Aerodynamic and Propulsion Forces Computation Module

The proposed fault tolerant air data system uses the values of the aerodynamic force and the thrust force acting on the aircraft as measurements to improve the estimation of air data. The aircraft aerodynamic and propulsion forces computation module calculates these values based on the accelerometer measurement, and a model for the propulsion system as explained in the following sections.

4.4.1 Aerodynamic Force Acting on the Aircraft

The proposed fault tolerant air data estimation system requires the computation of the aerodynamic force, $F_{a,b}$ acting on the aircraft center of gravity. The $F_{a,b}$ signal is used in the air data estimator and also to generate data to train the fuzzy inference systems of the aircraft digital twin. The accelerometer and its estimated bias will be used for this purpose.

The accelerometer bias b_a is estimated by the navigation system. This quantity is used to correct the sensor measurement drift as in

$$a_{b,unbiased} = a_{b,meas} - b_a \,, \tag{116}$$

The remaining accelerometer signal noise, n_a can be attenuated using a low pass filter, $H_{LP,a}(s)$, and by carefully tuning its signal band width BW_a, in a first order low-pass filter with transfer function

$$H_{LP,a}(s) = \frac{BW_a}{s + BW_a}, \tag{117}$$

The resulting attenuated acceleration signal at the center of gravity of the aircraft will be $\tilde{a}_{cg,b}$, where

$$\tilde{a}_{cg,b} = a_{b,unbiased} - n_a \,. \tag{118}$$

The accelerometer senses the specific net force, f, acting on the aircraft as follows

$$f = \tilde{a}_{cg,b} - G_b \,, \tag{119}$$

where f is the net force per unit mass at the aircraft center of gravity, and G_b is the gravity vector expressed in F_b. Thus, multiplying equation (119) by the aircraft mass it is found that

$$F_{net,b} = F_{a,b} + F_{t,b} + W_b = fm + G_b m = fm + W_b \,, \tag{120}$$

therefore, the aerodynamic force acting on the aircraft center of mass expressed in F_b becomes

$$F_{a,b} = fm - F_{t,b} = \left(\tilde{a}_{cg,b} + G_b\right)m - \, F_{t,b}. \tag{121}$$

4.4.2 Propulsion Force Acting on the Aircraft

The thrust force acting on the aircraft is obtained from a simplified propulsion system model. This simple model suffices for the goal of this research as it is not necessary to go in the detail of modelling the propeller. In addition, the Ryan Navion aircraft fuel tanks are located in the vicinity of the aircraft's center of gravity, thus having negligible effect on the aircraft dynamic behavior as the fuel is consumed. Since the simulation is carried out in 600 seconds, the effect in the change of mass and inertias of the aircraft with fuel consumption can also be neglected. The aircraft thrust line is assumed to be aligned with the center of gravity and in the positive direction of the x-axis of F_b. With these assumptions in mind, the aircraft thrust force can be computed by

$$F_t = F_{t,max} \cdot \delta_t$$

where F_t is the thrust force at a particular thrust control level δ_t. $F_{t,max}$ is the maximum thrust force delivered by the propulsion system. This value is published by the aircraft manufacturer.

4.5 Air Data Estimation Module

The air data estimation module is a nonlinear estimation algorithm used to estimate air data variables. In this research the Extended Kalman Filter algorithm has been selected to implement this module because of its low computational cost. A key feature of this estimator is that the sensor noise covariance matrices associated with the air data sensors are modulated based on the detection of the failures by the failure detection module. Another key feature is that the estimator is based on a nonlinear model of the aircraft implemented in the aircraft digital twin. The purpose of this module is to estimate the air data parameters $\hat{V}, \hat{\alpha}, \hat{\beta}$, and based on the sensor's health information received from the fault detection module decide if the Pitot tube and angle of attack vane measurements are used or not during the estimation process.

Figure 12 shows the EKF aircraft state estimator tolerant to Pitot and angle of attack vane sensor failure and the data flow to it. The estimated air data parameters $\hat{V}$ and $\hat{\alpha}$, are used to calculate the signal residuals to be analyzed at the sensor failure detection module. When a sensor failure is detected, the sensor failure detection module sends a feedback signal flag to the aircraft model EKF estimation algorithm that will stop the use of the unhealthy measurement signals in the estimation process by increasing its measurement noise covariance matrix R. The ability of the fault detection block to detect when the faulty sensor becomes healthy again and to send a healthy sensor flag as feedback to the

estimation algorithm, allows to recover the sensor signals that were omitted during the estimation process. To increase the precision of the estimation specifically during sensor failures, the estimation process uses virtual measurements of aerodynamic and propulsion forces provided by the aircraft digital twin. It is worth to emphasize that this fault tolerant estimation process is performed continuously and in real time.

4.5.1 Air Data Estimation Module Description

4.5.1.1 Continuous Process Model of Air Data Estimation

The Air Data Estimation process model is based on the aircraft equations of motion as presented in [66], and using $\dot{F}_{a,b} = [0\ 0\ 0]^T$ and $\dot{F}_{t,b} = [0\ 0\ 0]^T$, the process model equations can be expressed as:

$$\dot{V} = \left(\frac{F_{t,b}}{m}\right) cos(\alpha + \alpha_t) cos\beta - \left(\frac{D}{m}\right) + g_1 \,, \tag{122}$$

$$\dot{\beta} = \frac{1}{mV}\left[-F_{t,b} cos(\alpha + \alpha_t) sin\beta - C - mVr_s + mg_2\right], \tag{123}$$

$$\begin{aligned}\dot{\alpha} = \frac{1}{mcos(\beta)V}\big[&-F_{t,b} sin(\alpha + \alpha_t) - L + mg_3 + mVcos(\beta)q \\ &- sin(\beta)p_s\big]\,,\end{aligned} \tag{124}$$

$$\dot{F}_{t,b} = \begin{bmatrix}0\\0\\0\end{bmatrix}, \tag{125}$$

$$\dot{F}_{a,b} = \begin{bmatrix}0\\0\\0\end{bmatrix}. \tag{126}$$

Expressing the air data estimation process in continuous-time state-space form, the following expression is obtained:

$$\dot{x}(t) = f(x(t), u(t), t) + G(x(t), t)w(t), \tag{127}$$

or

$$\begin{bmatrix} \dot{V} \\ \dot{\beta} \\ \dot{\alpha} \\ \dot{F}_t \\ \dot{F}_{a,w} \end{bmatrix} = \begin{bmatrix} \left(\frac{F_t}{m}\right) cos(\alpha + \alpha_t) cos\beta - \frac{D}{m} + g_1 \\ -\left(\frac{F_t}{mV}\right) cos(\alpha + \alpha_t) sin\beta - \frac{C}{mV} + \frac{g_2}{V} - r_s \\ -\frac{F_t}{mcos\beta V} sin(\alpha + \alpha_t) - \frac{L}{mcos\beta V} + \frac{g_3}{cos\beta V} q - \frac{sin\beta}{cos\beta} p_s \\ 0 \\ 0_{3\times 1} \end{bmatrix} + \begin{bmatrix} n_V \\ n_\beta \\ n_\alpha \\ n_{F_t} \\ n_{F_{a,w}} \end{bmatrix}. \tag{128}$$

The state vector of the process model in the air data estimation is defined to include three relative velocity components, the angle of sideslip, the angle of attack, three aerodynamic force components and one propulsion force component:

$$x = \begin{bmatrix} V \\ \beta \\ \alpha \\ F_t \\ F_{a,w} \end{bmatrix}, \text{ where, } V = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, F_{a,w} = \begin{bmatrix} -L \\ -C \\ -D \end{bmatrix}, \tag{129}$$

In order to apply EKF to the continuous-time process model, it is necessary to calculate the process Jacobian matrix F of the nonlinear system in equations (122) ~ (126).

The Matlab/Simulink implementation in this research uses finite differences numerical approach to compute the air data estimation process Jacobian matrix.

4.5.1.2 Air Data Estimation Discrete Process Model

Here again it is necessary to express in discrete time the process model in order to apply the Extended Kalman Filter. The discrete time model of the continuous time process model in equations (122) to (126) can be approximated by the following first-order Euler integration algorithm:

$$V_{k+1} = V_k + \dot{V}_k \Delta t = V_k + \left[\left(\frac{F_{t,k}}{m}\right) cos(\alpha_k + \alpha_t) cos\beta_k - \left(\frac{D_k}{m}\right) + g_1\right] \Delta t \quad (130)$$

$$\begin{aligned} \beta_{k+1} &= \beta_k + \dot{\beta}_k \Delta t \\ &= \beta_k \\ &+ \left[\frac{1}{mV_k}\left[-F_{t,k} cos(\alpha_k + \alpha_t) sin\beta_k - C_k - mV_k r_{s,k}\right.\right. \\ &\left.\left. + mg_2\right]\right] \Delta t \end{aligned} \quad (131)$$

$$\begin{aligned} \alpha_{k+1} &= \alpha_k + \dot{\alpha}_k \Delta t \\ &= \alpha_k \\ &+ \left[\frac{1}{mcos(\beta_k)V_k}\left[-F_{t,k} sin(\alpha_k + \alpha_t) - L_k + mg_3\right.\right. \\ &\left.\left. + mV_k cos(\beta_k) q_k - sin(\beta_k) p_{s,k}\right]\right] \Delta t \end{aligned} \quad (132)$$

$$F_{t_{k+1}} = F_{t_k} + \left[\dot{F}_t\right]\Delta t = F_{t_k} \tag{133}$$

$$F_{a,w_{k+1}} = F_{a,w_k} + \left[\dot{F}_{a,w}\right]\Delta t = F_{a,w_k} \tag{134}$$

4.5.2 Air Data Estimation Measurement Equations

Following the same method and nomenclature used in section 3.3.5, the measurement models are:

4.5.2.1 Pitot Measurement Model

$$y_k^1 = h^2(x_k) + n_{\alpha_{\alpha vane},k} \Leftrightarrow \alpha_{\alpha vane,k} = \alpha_{\alpha vane,e,k} + n_{\alpha_{\alpha vane},k} \tag{135}$$

4.5.2.2 Angle of Sideslip Vane Measurement Model

$$y_k^2 = h^2(x_k) + n_{\beta_{\beta vane},k} \Leftrightarrow \beta_{\beta vane,k} = \beta_{\beta vane,e,k} + n_{\beta_{\beta vane},k} \tag{136}$$

4.5.2.3 Angle of Attack vane measurement model

$$y_k^3 = h^3(x_k) + n_{\alpha_{\alpha vane},k} \Leftrightarrow \alpha_{\alpha vane,k} = \alpha_{\alpha vane,e,k} + n_{\alpha_{\alpha vane},k} \tag{137}$$

4.5.2.4 Thrust Force Measurement Model

$$y_k^4 = h^4(x_k) + n_{F_{t,k}} \Leftrightarrow \ F_{t,k} = F_{t,k} + n_{F_{t,k}} \tag{138}$$

4.5.2.5 Aerodynamic Force Measurement model

$$y_k^5 = h^5(x_k) + n_{F_{a,w_{acc}},k} \Leftrightarrow F_{a,w_{acc},k} = F_{a,w_{acc},k} + n_{F_{a,w_{acc}},k} \tag{139}$$

4.5.2.6 Virtual Aerodynamic Force Measurement Model

$$y_k^6 = h^6(x_k) + n_{F_{a,w_g},k} \Leftrightarrow F_{a,w_g,k} = F_{a,w_g,k} + n_{F_{a,w_g},k} \quad (140)$$

4.5.2.7 Virtual Thrust Force Measurement Model

$$y_k^7 = h^7(x_k) + n_{F_{t_g},k} \Leftrightarrow F_{t_g,k} = F_{t_g,k} + n_{F_{t_g},k} \quad (141)$$

The Matlab/Simulink linearization of the air data estimation measurement models was carried out numerically by implementing a numerical computation of the Jacobian measurement matrix H finite differences.

4.5.3 Observability Analysis

An observability analysis shows that the system is locally observable along the aircraft trajectory. This guarantees the observability of the non-linear system [67], [68], [69]. The idea of the proposed air data estimation scheme is not to mathematically demonstrate observability. In this work an estimation methodology is demonstrated where if the observability condition is fulfilled using healthy sensors, the system is able to accommodate the failure. The complete observability guarantees detectability.

The observability analysis of the air data estimator was performed numerically during the simulation using the process Jacobian matrix F and the measurement Jacobian matrix H. The observability matrix is computed as: $Obvs = [H; H \cdot F; H \cdot F^2; H \cdot F^3; H \cdot F^4; H \cdot F^5; H \cdot F^6]$. The rank of $Obvs$ was computed using the "rank" Matlab® function and it was found to be full-rank (The rank of 7 equals to the 7 state variables) during the

different failures generated in the simulation meaning the system is fully observable. The results of this analysis will be shown in chapter 6.

4.5.4 Sensor Failure Detection

The residual sensor signal is defined as the difference between the measured and estimated signal values. When the sensor is working correctly the corresponding residual signal mean should be constrained between thresholds that have a magnitude a bit above the residual signal standard deviation. In contrast, when the sensor failure occurs, its residual signal mean trespasses the residual signal threshold. The goal of this sub-system is to detect when the residual signal mean surpasses the residual signal threshold and also when it returns below the residual signal threshold. The Pitot and Angle of Attack vane residual signals will be constantly monitored by this subsystem.

4.5.5 Residual Signal Mean

The airspeed and angle of attack residual signal mean characteristics should represent the dynamic behavior of the aircraft in all the maneuvers but without all the noise peaks. This is accomplished by implementing a low pass filter with a band width high enough to include all the maneuvers of the aircraft but not as high to include all the noise peaks. Tuning the filter can be done by analyzing aircraft flight data (actual or simulated) that includes maneuvers and check for the variation presented to estimate the required band width. The residual signal mean is computed using the following transfer function:

$$Mean(s) = \frac{(BW)_{mean}}{s + (BW)_{mean}} \cdot u(s) \tag{142}$$

where $u(s)$ is the unfiltered signal, $(BW)_{mean}$ is the residual signal band width of the sensor that is being analyzed and $Mean(s)$ is the approximate residual signal mean in a time window that includes the last samples especially those is the last τ seconds (here τ is the time constant of the filter $\tau = 1/BW$). In fact, this filter exponentially weights the last samples to compute the mean.

4.5.6 Residual Signal Standard Deviation

The airspeed and angle of attack residual signal standard deviation is computed using transfer function:

$$Standard\ Deviation\ (s) = \sqrt{\frac{(u(s) - mean(s))^2}{s + (BW)_{std}}} \tag{143}$$

where $u(s)$ is the signal being processed, $(BW)_{std}$ is the residual signal band width of the sensor that is being analyzed and $mean\ (s)$ is the is the mean computed by the mean filter.

To guarantee a fault detection, the standard deviation filter must be tuned to a lower band width than the mean filter so the computed thresholds for the detection of the failure, which are based on this standard deviation, don't change as fast as the mean signal. This mean and thresholds are computed in real time so they accommodate to the actual data of the sensor.

4.5.7 Threshold Computation

The residual signal upper and lower thresholds are computed by adding and subtracting the standard deviation to the mean value of the residual signal. The standard deviation value is multiplied by a gain to generate a clearance capable of eliminating possible false detections. However, this gain should be tuned carefully to avoid undetected failures.

4.5.8 Failure Detection Logic

The failure detection logic is carried out by a function that compares the absolute value of the sensor residual mean with the residual failure threshold. If the residual signal mean crosses a residual signal threshold a failure is detected and a failure flag called "failureFlag" is set to 1 indicating that a failure happened. On the other hand, if the residual is located between the thresholds then "failureFlag" is set to zero indicating that the sensor being analyzed is healthy.

To be able to restore the system after an unhealthy sensor becomes healthy again, an additional variable called "inRange" is used as a flag that determines if the residual signal mean is located inside or outside the residual threshold failure. The "inRange" flag is also used to determine the time elapsed when the sensor signal returns to a healthy condition in the case the sensor recovers from a failure. If the absolute value of the residual signal mean is smaller than the residual failure threshold a value of 1 is assigned to the "inRange" flag. On the other hand, if the absolute value of the residual signal mean is greater than the residual failure threshold a value of 0 is assigned to the "inRange" flag. The "inRange" flag signal feeds an integrator in such a way that only when the "inRange"

signal has a value of 1 the integrator outputs the time that the sensor has stayed healthy after recovering from the failure. This happens since the integrator automatically resets itself when the failure stops, that is, when the "failureFlag" changes from 1 to 0. The function that implements the failure detection logic resets the failure flag to zero when the time that the sensor has been healthy after the failure is greater than certain specified value called "inRangeTimeThreshold". In this way, when a sensor has failures that repeats in short periods of time, the failure flag remains set to indicate the failure, and after the sensor recovers from the failure the system can return to normal operation.

4.5.9 Air Data Estimation Covariance Modulation

The estimation algorithm of this system is similar to the one explained in section 4.3.7 the only difference is that when the Pitot and (or) angle of attack vane sensors fail, the sensor failure detection system will send a failure flag per failed sensor to the air data estimation module. Each sensor failure flag will activate an increase in magnitude of the respective measurement noise covariance matrix, hence R_{pitot} will be transformed to $R_{pitot}^{failure}$ and $R_{\alpha_{vane}}$ will be transformed to $R_{\alpha_{vane}}^{failure}$ respectively. Similarly the process noise covariance matrix for each sensor will be increased in magnitude, hence Q_{pitot} will be transformed to $Q_{pitot}^{failure}$ and $Q_{\alpha_{vane}}$ will be transformed to $Q_{\alpha_{vane}}^{failure}$. in the estimator update equations. The process noise covariance matrix Q will then be transformed to $Q_{failure} = diag\left[Q_{pitot}^{failure}\ ;\ Q_{\beta_{vane}}; Q_{\alpha_{vane}}^{failure}; Q_{F_t}; Q_{F_a}; Q_{F_{a_g}}; Q_{F_{t_g}}\right]$ and as a consequence a new state error covariance matrix is obtained as in, $P_{failure} = F \cdot P_k \cdot F^T + Q_{failure}$.

The overall effect in the aircraft state estimation process is that the faulty sensor measurements have a very small effect on the estimated values, so the estimation relies on the information provided by the healthy sensors. Notice that in this fault tolerant air data estimation scheme, the faulty sensors are not isolated, they are constantly used during the estimation process. Virtual measurements y_k^6 and y_k^7 from the aircraft digital twin helps in keeping the accuracy of the estimation relatively constant during Pitot and angle of attack vane simultaneous sensor failures.

4.6 Aircraft Digital Twin

The purpose of the aircraft digital twin is to act as a virtual sensor and provide virtual measurements of $F_{a,w}$ and F_t to improve the accuracy of the estimated air data variables $\hat{\alpha}$, $\hat{\beta}$, and $\widehat{V}_{rel}$ when the pitot and angle of attack vane sensors fail. The digital twin model should be capable of representing the aircraft dynamic behavior during all the maneuvers in flight and should be trained with a relatively low amount of flight data. To reduce computational cost during training and operation, the digital twin should allow to use the known dynamic behavior of the aircraft. It is possible to meet all previous requirements if the aircraft digital twin is based on a gray-box Adaptive Neuro-Fuzzy Inference System – ANFIS, based in the Takagi-Sugeno-Kang method also known as a Sugeno Fuzzy Inference System – S-FIS. The S-FIS method is well suited for modeling nonlinear systems by interpolating between multiple linear models and it lends itself to the use of adaptive techniques for constructing fuzzy models. This is the case of the aircraft dynamic model at trim conditions during a particular maneuver, where in the vicinity of the equilibrium condition the dynamic model have a linear behavior [70].

The method used in this research for implementing the S-FIS and training the S-ANFIS is described in detail on [71], [72], [73], [74] and [75].

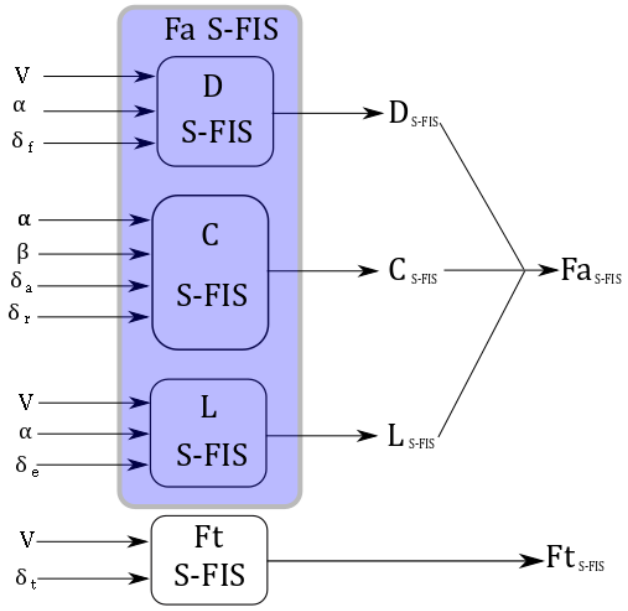

Figure 14. Aircraft digital twin.

Figure 14 shows a block diagram of the aerodynamic and thrust force, $F_{a,w}$ and F_t, S-FIS. To infer F_a, is necessary to build a S-FIS for each of its force components D, C and L. The S-FIS will map aircraft state variables that are strongly correlated to each aerodynamic force component. The drag force component D is a strongly correlated with V, α and in less extent with δ_f. The crosswind force component C is strongly correlated with, β, δ_a and δ_r. The lift force component L is strongly correlated with V, α and in less extent with δ_e. In contrast, the S-FIS for F_t is simpler because it is only strongly correlated with V and δ_t. However only the two strongly correlated inputs for each S-FIS will be used to reduce the training time and computational cost during flight simulations.

CHAPTER 5. SIMULATION MODEL

5.1 Introduction

To test and evaluate the performance of the proposed air data system tolerant to sensor failure it is necessary to count with an instrumented experimental aircraft, a trained test pilot and a well-planned flight test program. Unfortunately, it is difficult to have access to an experimental aircraft, a trained flight test pilot and a flight test facility with the respectively airworthiness authority clearances, so the air data system tolerant to sensor failure developed in this research was tested in simulation. The system simulation model must include an autopilot that replaces the human pilot, an aircraft simulation model that provides true flight data signals to the sensor models and the simulation model of the proposed fault tolerant air data system. Actual aircraft flight test data published in different sources can be used to develop the aircraft dynamic model, which in turns is used to build the aircraft flight simulation model as well as in the autopilot tuning. Actual sensor data can be used to build the sensor simulation models. This section will explain how the fault tolerant air data system simulation model was created.

Figure 15 shows the simulation model block diagram. The autopilot block is based on a Simulink® model developed by Dr. Luís Benigno Gutiérrez Zea in his undergraduate course of automatic flight control. The Autopilot sends throttle δ_t, flap δ_f, elevator δ_e, aileron δ_a, and rudder δ_r control commands based on user defined airspeed, altitude and heading setpoints to the aircraft simulation block where a six degree of freedom non-linear flight simulation is carried out. The airspeed, altitude, and heading setpoints were modified

to include deceleration, descent and change in heading so different flight maneuvers were included in this flight simulation. The autopilot is composed by longitudinal and lateral flight control subsystems. The longitudinal control features two subsystems, an airspeed PID which controls the throttle in such a way that the aircraft flies at constant airspeed, and the altitude PID that generates the pitch setpoint to feed the pitch PID that controls the elevator. The lateral control block features a heading PID that computes a roll set point that feeds the roll PID to control the ailerons. The sideslip PID controls the rudder.

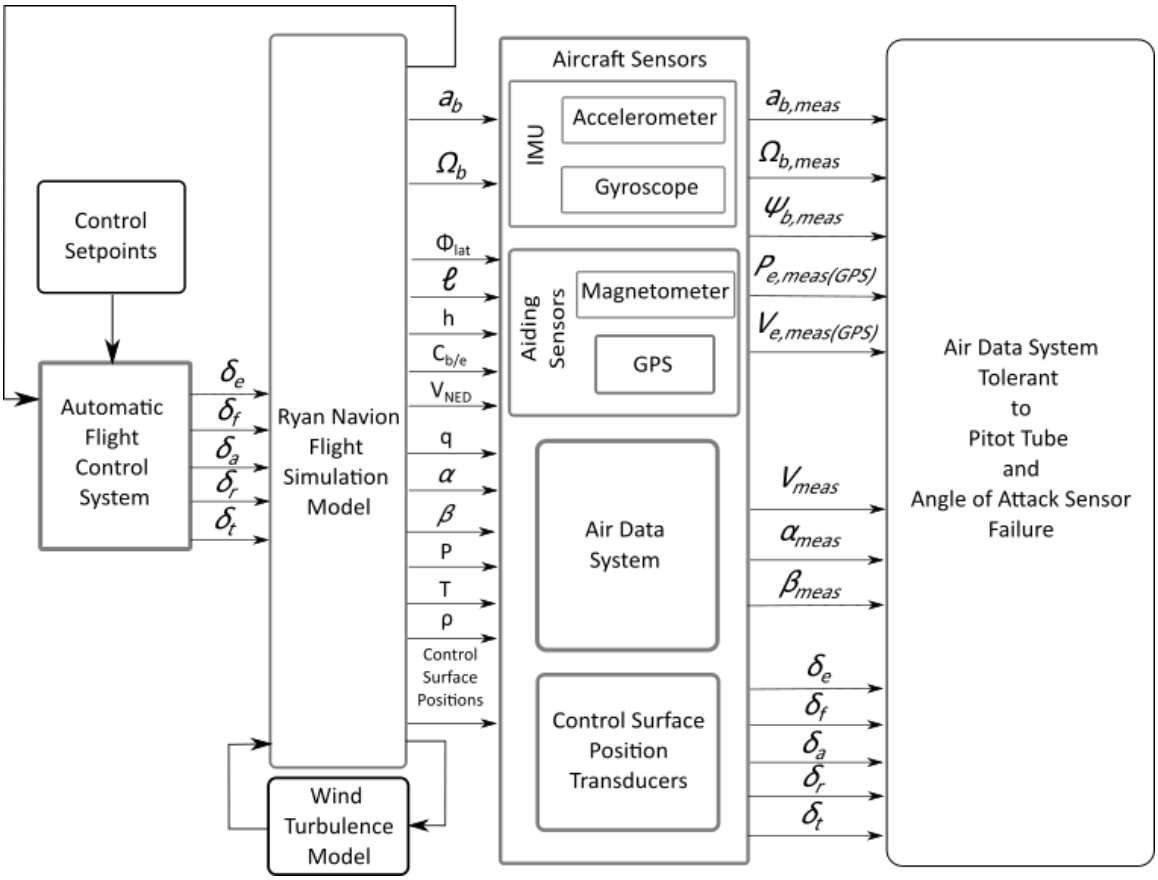

Figure 15. Simulation model block diagram.

The aircraft simulation block is based on a Simulink® model developed by Dr. Luís Benigno Gutiérrez Zea in his undergraduate course of Flight Dynamics. This block provides state true values to the sensor models where noise and bias are added to each signal. The aircraft state includes position in geodetic coordinates (latitude $\emptyset_{lat}$, longitude ℓ and altitude h), direction cosine matrix ($C_{b/e}$), attitude quaternion (q), aircraft velocity expressed in the body frame (V_b), aircraft velocity expressed in the earth frame (V_e), aircraft linear acceleration expressed in the body frame (a_b), aircraft angular velocity expressed in the body frame (Ω_b), atmospheric pressure (P), atmospheric temperature (T), atmospheric density (ρ), dynamic pressure (q_{bar}), aircraft angle of attack (α), and aircraft angle of sideslip (β). The block feedbacks the autopilot with the aircraft airspeed (V), altitude (h), pitch angle (θ), heading angle (ψ), rolling angle ($\emptyset$) and sideslip angle (β). To include wind turbulence on the flight simulation, a Dryden wind turbulence model receives true state data of h, $C_{b/e}$ and V to provide the aircraft simulation model a wind angular velocity expressed in the earth frame ($\Omega_{w,e}$) and a wind velocity expressed in the earth frame ($V_{w,e}$).

The state true-value signals provided by the aircraft simulation model are transformed to sensor measurement signals by adding noise and bias in the aircraft sensors model block that contains an ADS module featuring simulation models of a Pitot tube, angle of attack and sideslip vane sensors, which provides measurement signals of airspeed V_{meas}, angle of attack α_{meas} and angle of sideslip β_{meas} respectively. This block also contains an IMU module featuring a three-axis accelerometer and a gyroscope simulation models that provides measured signals of linear acceleration $a_{b,meas}$ and angular velocity $\Omega_{b,meas}$ both expressed in the body reference frame. The aiding navigation sensors are also modelled in this block (magnetometer and GPS) each one providing measured signals of

the magnetic field expressed in the body reference frame $M_{b,meas}$, position $P_{e,meas(GPS)}$ and ground speed $V_{e,meas(GPS)}$ respectively. As it will be explained below, the sensor noise and bias parameters were obtained by performing measurements from the actual sensors signals by computing its statistics. Sensor faulty signals are modeled inside the Pitot and angle of attack vane models. Pitot tube and angle of attack vane failures were generated independently and simultaneously in different stages of aircraft flight: climb, descent, acceleration, deceleration, straight level flight and turn. In each type of maneuver, the system was capable of detecting the sensor (or sensors) failure (or failures) and estimate with accuracy the airspeed, angle of sideslip and angle of attack.

The control surface position transducers are linear potentiometers that translates the control surface deflections in control signals. These sensors are necessary only if actual aircraft flight test is carried but for the purpose of simulation the control signals are taken directly from the autopilot.

5.2 Aircraft Model

The selected aircraft to perform the flight simulations was the Ryan Navion model "A" as shown in Figure 16. This general aviation aircraft is a well-documented airplane because it has been used as a research flying platform in the last 50 years by NASA, the University of Princeton (registration number N91566) [76] and currently by the University of Tennessee (registration number N66UT) [77]. These institutions had instrumented the aircraft and performed flight tests for the development of automatic flight control systems. The full-size aircraft was also subjected to extensive test at NASA Langley Research Center $30' \times 60'$ wind tunnel in 1969 [78].

Figure 16. Ryan Navion.

Theoretical predicted dynamic model for the Ryan Navion is available in several flight mechanics textbooks [79], [80], NACA reports [81] and USAF DATCOM data which was prepared by Douglass engineers [82]. This dynamic model information relies in manufacturer design data, theoretical estimations, statistics, judgment and experience from the authors. Methods used to predict the longitudinal and Lateral-directional response of small rigid aircraft are available [83]. In 1970 the University of Princeton carried out a series of full-scale wind tunnel test of the Ryan Navion to verify some dynamic data and obtain new parameters. The wind tunnel aircraft model was a full size Navion with an electric motor propulsion system. The flight test was carried out at cruise clean condition and also with flaps deflected 20 degrees.

In January 1971, the Navion theoretical, and wind tunnel dynamic data accuracy was verified by Princeton University for Federal Aviation Administration (FAA); under contract No. FA 68W A-1901. The FAA was testing a new technique for estimating aerodynamic loads and reliable stability derivatives were mandatory for the method to achieve reliable results. The Navion aircraft was instrumented and a series of flight test

carried out. Tables 4 ~ 15 summarize the verified accurate dynamic data and its respective source. Unfortunately, dynamic data with landing gear deployed and flap control derivatives were not published. However, the aim of this research is to capture the dynamic behavior of a particular aircraft dynamic model and to perform air data estimation during sensor failures. Since this flight test data is only a function of the aircraft shape and propulsion system used it will be valid over time and can be used for the purpose of this research.

Table 3. Navion Trim conditions.

Computed Aircraft Trim Conditions		
Free Stream Mach Number	M_o	0.211
Zero Lift Coefficient	C_{L_o}	0.3
Zero Lift Drag Coefficient	C_{D_o}	0.04
Trim Condition Moment Coefficient	C_{m_o}	0.0
Minimum Drag Coefficient	$C_{D_{min,drag}}$	0.3
Span Efficiency Factor (Oswald)	e	0.77

Table 4. Navion longitudinal static stability derivatives.

Longitudinal Static Stability	Abbreviation	Value	Source
Aircraft Lift Curve Slope	C_{L_α}	6.04	Flight Test
Static Stability Derivative	C_{m_α}	-0.715	Etkin

Table 5. Navion stability control derivatives.

Control Derivative	Abbreviation	Value	Source
Drag Due-to-Flap Deflection	$C_{D_{\delta_f}}$	0.0	N/A
Drag Due-to-Tail Incidence	$C_{D_{i_h}}$	0	Fixed Tail
Drag Due-to-Elevator Deflection	$C_{D_{\delta_e}}$	0	N/A
Drag Due-to-Aileron Deflection	$C_{D_{\delta_a}}$	0	N/A
Drag Due-to-Rudder Deflection	$C_{D_{\delta_r}}$	0	N/A
Side Force Due-to-Aileron Deflection	$C_{Y_{\delta_a}}$	0	N/A
Side Force Due-to-Rudder Deflection	$C_{Y_{\delta_r}}$	0.157	NASA STI-TR-176-1
Crosswind Due-to-Aileron Deflection	$C_{C_{\delta_a}}$	$-C_{Y_{\delta_a}}$	Assigned
Crosswind Due-to-Rudder Deflection	$C_{C_{\delta_r}}$	$-C_{Y_{\delta_r}}$	Assigned
Lift Due-to-Flap Deflection	$C_{L_{\delta_f}}$	0.0	N/A
Lift Due-to-Tail Incidence	$C_{L_{i_h}}$	0.0	Fixed Tail
Lift Due-to-Aileron Deflection	$C_{L_{\delta_e}}$	0.355	NASA STI-TR-176-1
Roll Moment due-to-aileron	$C_{l_{\delta_a}}$	-0.152	Wind Tunnel Test
Roll Due-to-Rudder	$C_{l_{\delta_r}}$	0.107	NASA STI-TR-176-1
Moment Due-to-Flap	$C_{m_{\delta_f}}$	0.0	N/A
Tail Effectiveness	$C_{m_{i_h}}$	0.0	Fixed Tail
Elevator Effectiveness	$C_{m_{\delta_e}}$	-1.42	Flight Test
Aileron Yaw	$C_{n_{\delta_a}}$	0.0047	Wind Tunnel
Yaw Control	$C_{n_{\delta_r}}$	-0.075	Flight Test

Table 6. Navion Sideslip stability derivatives.

Sideslip Derivatives	Abbreviation	Value	Source
Side Force	C_{Y_β}	-0.61	Flight Test
Crosswind Force Due-to-Sideslip	C_{C_β}	$-C_{Y_\beta}$	Assigned
Yaw Stability	C_{n_β}	0.08600	Flight Test
Roll Stability	C_{l_β}	-0.067	Flight Test

Table 7. Navion roll rate stability derivatives.

Roll Rate Derivatives	Abbreviation	Value	Source
Side Force Due-to-Pitch Rate derivative	C_{Y_p}	0	N/A
Crosswind Force Due-to-Pitch Rate derivative	C_{C_p}	$-C_{Y_p}$	Assigned
Roll Damping Derivative	C_{l_p}	-0.46	Flight Test
Adverse Yaw Derivative	C_{n_p}	-0.03800	Flight Test

Table 8. Navion yaw rate stability derivatives.

Yaw Rate Derivatives	Abbreviation	Value	Source
Side Force Due-to-Yaw Rate derivative	C_{Y_r}	0.0	N/A
Crosswind Force Due-to-Yaw Rate derivative	C_{C_r}	$-C_{Y_r}$	Assigned
Roll Due-to-Yaw Derivative	C_{l_r}	0.069	Flight Test
Yaw Damping Derivative	C_{n_r}	-0.088	Flight Test

Table 9. Navion pitch rate derivatives.

Pitch Rate Derivatives	Abbreviation	Value	Source
Lift to Pitch Rate Derivative	C_{L_q}	3.8	NASA STI-TR-176-1
Pitch Damping Derivative	C_{m_q}	-13.39	USAF DATCOM

Table 10. Navion speed derivatives.

Speed Derivatives	Abbreviation	Value	Source
Lift Speed Damping	C_{L_u}	0.0	Negligible Compressibility
	C_{L_M}	C_{L_u}/M_0	Computed
Pitch Speed Damping	C_{m_u}	0.0	Negligible Compressibility
	C_{m_M}	C_{m_u}/M_0	Computed
Drag Speed Damping	C_{D_u}	0.0	Vanishes at Trim Condition $\alpha = 0$
	C_{D_M}	C_{D_u}/M_0	Computed

Table 11. Navion pitch rate derivatives.

Pitch Rate Derivatives	Abbreviation	Value	Source
Pitch Angle of Attack Rate Derivative	$C_{m_{\dot{\alpha}}}$	-4.91	Etkin, Seckel
Lift Angle of Attack Rate Derivative	$C_{L_{\dot{\alpha}}}$	0.0	N/A

Table 12. Navion Propulsion System.

Propulsion System	
Reciprocating Engine	Continental Model No. 10520B
Propeller	2 blade - 84 in diameter
Power	285 HP @ 2700 rpm at take off
Maximum Thrust	562 lbs.

Table 13. Navion geometric properties.

Ryan Navion Geometric Properties	
Wing Area	184 ft^2
Wing Span	33.38 ft
Mean Aerodynamic Chord	5.7 ft
Aspect Ratio	6.04
Horizontal Tail Area	43 ft^2

Table 14. Navion mass properties.

Ryan Navion Mass Properties		
Aircraft Mass	m	2948.00000 lbs
Moment of Inertia About x axis	I_{xx}	1284.00000 $slug \cdot ft^2$
Moment of Inertia About y axis	I_{yy}	2773 $slug \cdot ft^2$
Moment of Inertia About z axis	I_{zz}	3235 $slug \cdot ft^2$
Moment of Inertia About xz axis	I_{xz}	0.0

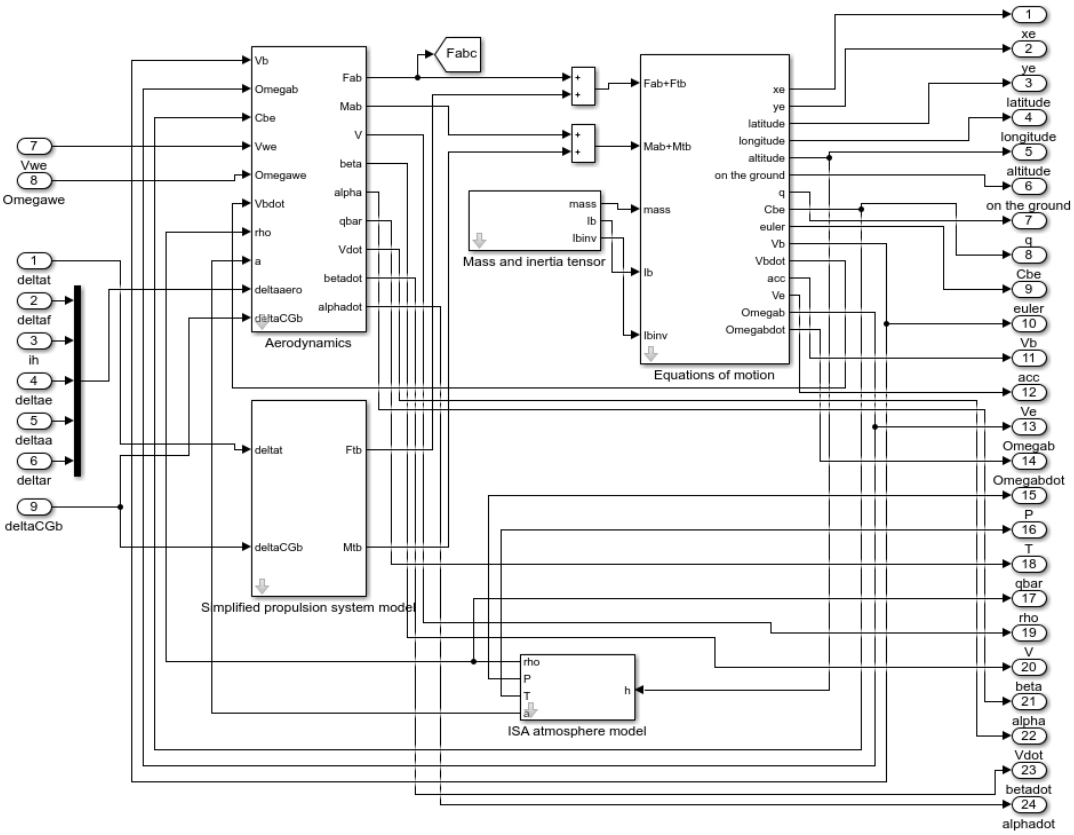

Figure 17. Simulink® Ryan Navion model.

Using the information supplied in Table 3 to Table 14 and the mathematical aircraft dynamic model described in Chapter 3, the Navion aircraft Simulink® model shown in Figure 17 was created [53].

5.3 Sensor Models

The aircraft flight simulation provides true state data to the sensor models: latitude, longitude, altitude, velocity, angular velocity, linear accelerations, dynamic pressure, angle of attack, angle of sideslip, static pressure and static temperature. Sensor models will add noise, bias and dynamics to the true aircraft state variable signals so its behavior mimics the actual sensors. Taking advantage of a flight test instrumentation developed by Dr. Luís

Benigno Gutiérrez Zea for the certification of an agricultural light sport aircraft, actual sensor data was available to create the sensor simulation models. The flight test instrumentation features the following off-the-shelf sensors:

- Xsens MTi G 700 Series sensor suit which contains an IMU (Gyroscopes and Accelerometers), GPS and a Magnetometer.
- Aeroprobe Micro Air Data System V2.0 which contains the Pitot-static probe, and provides measurements of angle of attack and angle of Sideslip. This instrument come with an air data computer.
- Honeywell Linear Position Transducers are used to read the position of Flaps, Ailerons, Elevator, Rudder and Throttle.

Figure 18. Instrumentation for real-time inflight state estimation.

The sensor signals are processed in Fit PC - IPC2 computer with an Intel core i7 processor, solid state disk of 512GB, 16 GB RAM. Figure 18 shows the developed instrumentation suite. The instrument signals were registered during a period of 20 minutes to compute its statistics as shown in Table 15.

The sensor dynamics was modelled as a first order model in state space form as shown in Figure 19 for each sensor signal component "i", the state space dynamic model was written as:

$$\dot{x}_i = Ax_i + B\,u_i \tag{144}$$

$$y = C\,x_i + D\,u_i \tag{145}$$

Table 15. Sensor statistical data.

Instrument	Sensor		Bias Standard Deviation	Noise Standard Deviation	Bandwidth [rad/s]	Sample Time [s]
ADS	Pitot [m/s]		-	0.1	628.32	0.02
	Alpha Vane [deg]		-	0.0001	62.83	
	Beta Vane [deg]		-			
	Bar. Altimeter [m]		-	0.1	62.83	
	Thermocouple [C]		-	0.1	0.6283	
IMU	Accelerometer [m/s^2]		X:0.000016	0.016627	251.328	0.01
			Y: 0.000021	0.021090		
			Z: 0.000098	0.009834		
	Gyroscope [rad/s]		0.00000144	0.001440		
			0.00000154	0.001545		
			0.00000158	0.00158938		
Aiding Sensors	Magnetometer [rad/s]		Lat: 0	Lat: 0.0041	251.328	0.02
			Long: 0	Lon: 0.0069		
			Alt: 0	Alt: 0.0094		
	GPS	Position [m]	H:6.45 e-11	-	-	0.25
			V:0.000764			
		Speed [m/s]	H: 0.041975			
			V: 0.019444			

where

$$A = \begin{bmatrix} -BW & 0 & 0 \\ 0 & -BW & 0 \\ 0 & 0 & -BW \end{bmatrix}, B = \begin{bmatrix} BW & 0 & 0 \\ 0 & BW & 0 \\ 0 & 0 & BW \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (146)$$

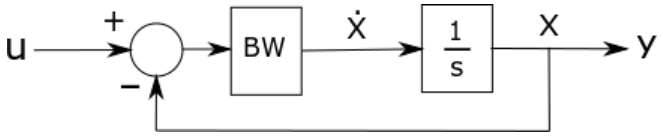

Figure 19. Sensor dynamics in state space form.

5.3.1 Air Data System

The ADS models use the dynamic pressure, angle of attack, angle of sideslip, atmospheric pressure and temperature as input signals and it will add noise and bias were necessary in each sensor model to return the measured values of the dynamic pressure, angle of attack, angle of sideslip, atmospheric pressure and atmospheric temperature. Figure 20 shows the Simulink® block diagram of the ADS. Figure 21 shows the second level block diagram for the ADS. Here the blocks for the Pitot model, angle of attack vane sensor, angle of sideslip vane sensor, barometric altimeter and total temperature probe models are shown. Inputs for Pitot and Angle of attack stuck and additive failures are shown as step signal blocks indicating each the time and duration of the simulated failure.

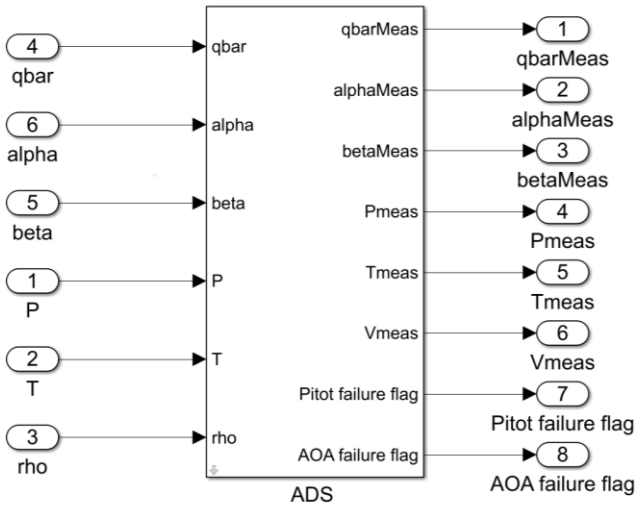

Figure 20. Simulink® model of ADS.

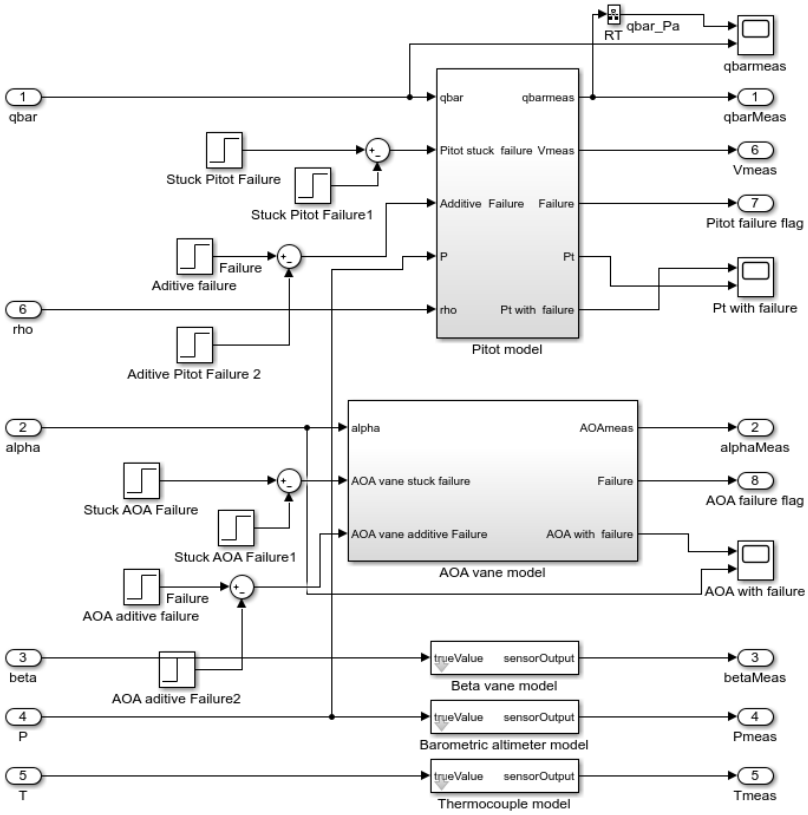

Figure 21. Simulink® ADS second level block diagram.

The fault in the Pitot tube is modelled in two ways. The first method is an additive bias in the airspeed residual signal intended to model physical electric hardware malfunction. The second failure approach, called here stuck failure, reflects the scenario where the Pitot tube dynamic pressure port is blocked and the static port is not affected by the surrounding atmosphere. In the stuck failure approach, when the failure occurs, the dynamic pressure signal is kept constant and the static pressure varies when the aircraft maneuvers to change its altitude. In addition, if the aircraft flies in straight level flight, the stuck failure in the Pitot tube is detected only if the aircraft changes its airspeed as a consequence of a change in wind speed or a variation in thrust.

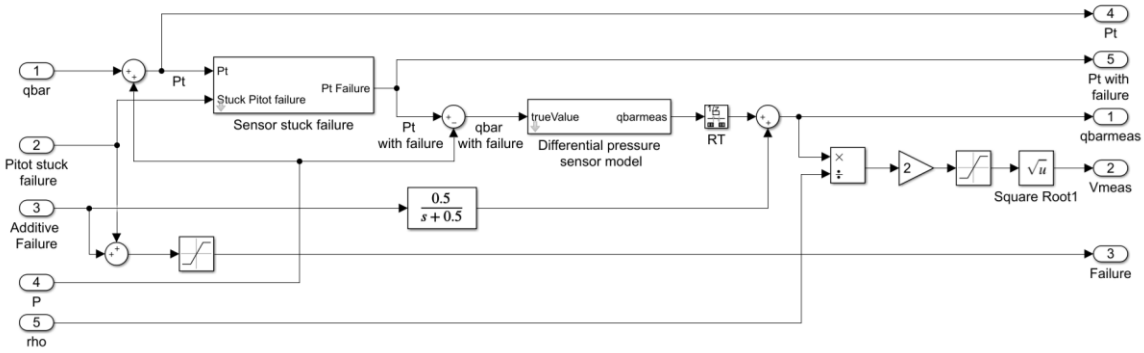

Figure 22. Pitot Simulink® model block.

The faults are introduced in the Pitot sensor model as shown in Figure 22. The airspeed additive bias failure is introduced by means of a step block that injects a dynamic pressure increment at a specified simulation time. Notice that the dynamic pressure increment signal passes through a transfer function before being added to the dynamic pressure so the time that it will take the failure to reach its highest value can be controlled.

The stuck Pitot failure sends a step signal (0 or 1) to a switch block as shown in Figure 23. If the control signal value increases above a threshold, in this case 0.5, the switch block outputs the total pressure value stored in the memory block in that instance until the control signal reverses the fault on the sensor. When the failure in the Pitot tube is present, the total pressure that comes out from the block (Pt Failure) will have a constant value during the flight and the static pressure will be subtracted to obtain the dynamic pressure before entering the differential pressure sensor model block. In this way, the dynamic pressure will vary when the aircraft changes its altitude or if there is a change in airspeed due to the presence of wind airspeed or a change in thrust triggering the fault detection alarm.

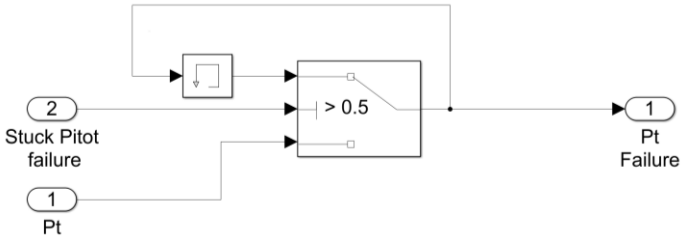

Figure 23. Pitot Simulink® Sensor Stuck Failure block.

The Pitot differential pressure sensor was modelled using as input signal the dynamic pressure passing through a zero-order hold and a discrete state space model to which a white noise and a bias modelled as a random walk is added to obtain the dynamic pressure measured signal. Figure 24 shows the Simulink® block diagram for the differential Pitot pressure sensor.

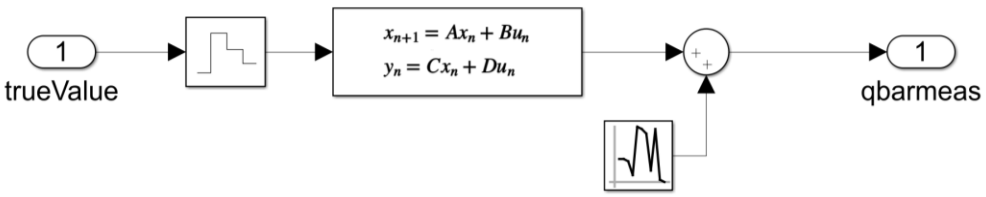

Figure 24. Simulink® model of the Pitot sensor.

A similar approach to the stuck Pitot failure was implemented to simulate the angle of attack vane failure as shown in Figure 25. The stuck angle of attack vane failure uses a step block to send a control signal (0 or 1) to a switch block. If the control signal value increases above a threshold, in this case 0.5, the switch block outputs the angle of attack value stored in the memory block in that instance until a control signal reverses the fault on the sensor as shown in Figure 26. When the failure in the angle of attack vane is present, the angle of attack signal value that comes out from the block (AOA with Failure) will have a constant value during the flight before entering the sensor state space model. The Angle of attack value will vary when the aircraft changes its speed or if there is a control adjustment in attitude.

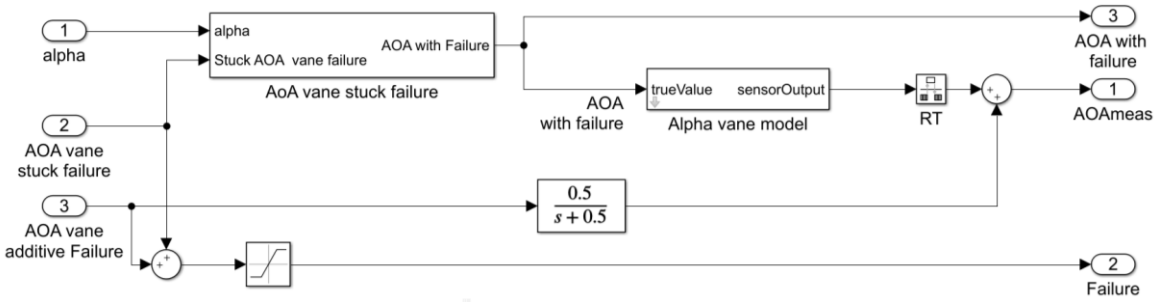

Figure 25. Angle of attack vane Simulink® model block.

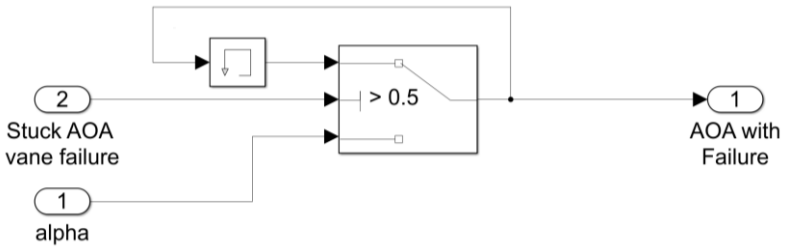

Figure 26. Angle of attack vane sensor Stuck failure Simulink® model block.

The angle of attack vane sensor model was modelled using as input signal the angle of attack to which a white noise and a bias modelled as a random walk is added. The sensor dynamics is modelled as a first order process. The signal enters a zero-order hold before giving as output the measured angle of attack. A similar model is applied to the sideslip angle vane, the Barometric altimeter and the temperature probe sensors but using as inputs signals the sideslip angle, the ambient static pressure and the ambient air temperature respectively. Figure 27 shows the Simulink® block diagram for these sensors.

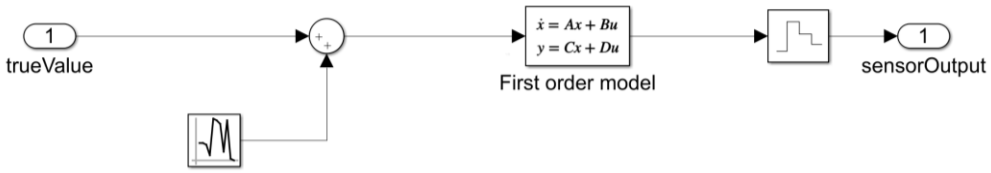

Figure 27. Simulink® model of the angle of attack, sideslip, barometric altimeter and temperature sensors.

5.3.2 IMU

The IMU is composed by the three-axis accelerometer and gyroscope sensors with angular velocity and linear acceleration as inputs respectively. To each signal a white noise and a bias modelled as a random walk is added. The noisy and biased signal of each sensor is subjected to a first order model dynamics and a zero-order hold before the outputs of

measured acceleration and angular velocity and its biases. Figure 28 and Figure 29 shows the IMU Simulink® block diagram and each sensor model respectively.

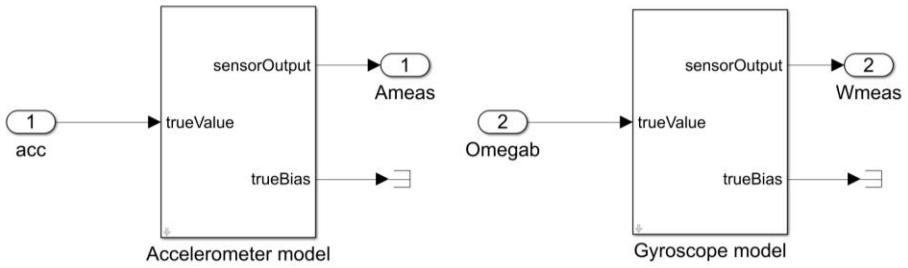

Figure 28. IMU Simulink® model.

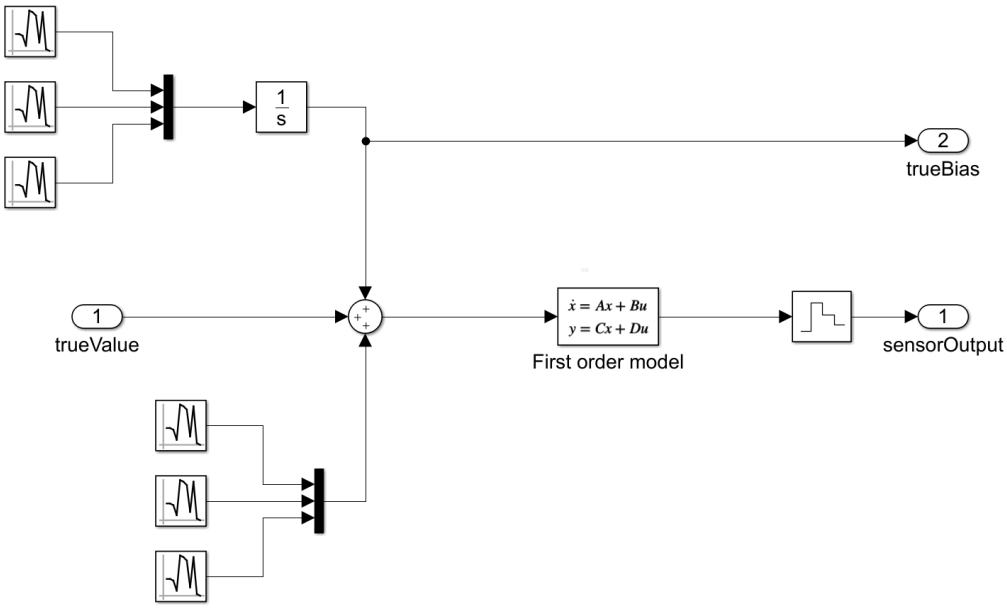

Figure 29. Accelerometers and Gyroscopes Simulink® model.

5.3.3 Magnetometer

The magnetometer was model using the latitude, longitude, altitude and $C_{b/e}$ as inputs and returning the earth's magnetic field vector in the aircraft position and its true bias b_m. The Simulink® model was constructed using a Matlab embedded function block

that runs the “wrldmagm” Matlab function. A white noise and a bias modelled as random walks are added to the wrldmagm output signal. Finally, the noisy and biased signal passes through a first order dynamic model and a zero-order hold. Figure 30 and Figure 31 shows the magnetometer Simulink® block diagram with the embedded Matlab function and the sensor model respectively.

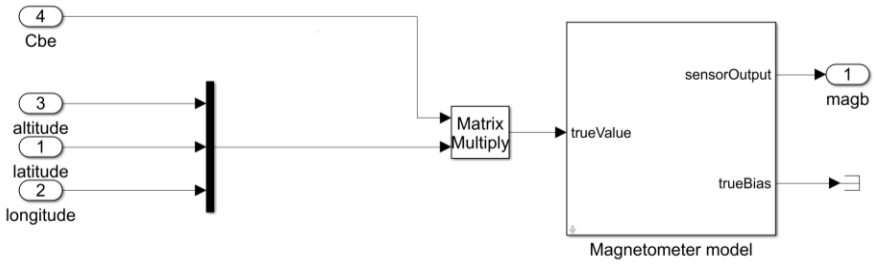

Figure 30. Magnetometer Simulink® model .

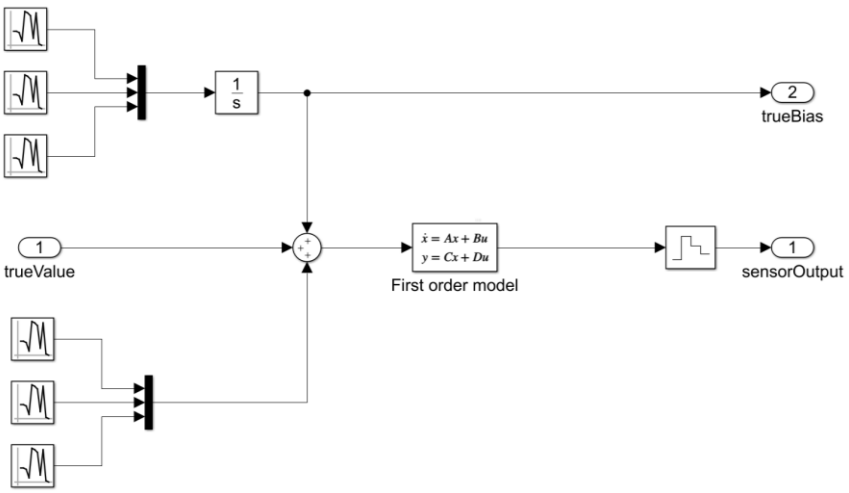

Figure 31. Magnetometer Simulink® model.

5.3.4 GPS

The GPS model takes the simulation values of latitude, longitude, altitude and velocity true values as inputs. A bias modelled as a random walk that is added to each position component. Then a zero-order hold and a unit delay are applied to the biased signal before the output of the GPS measured position. Figure 32 and Figure 33 shows the GPS position Simulink® block diagram and its sensor model respectively.

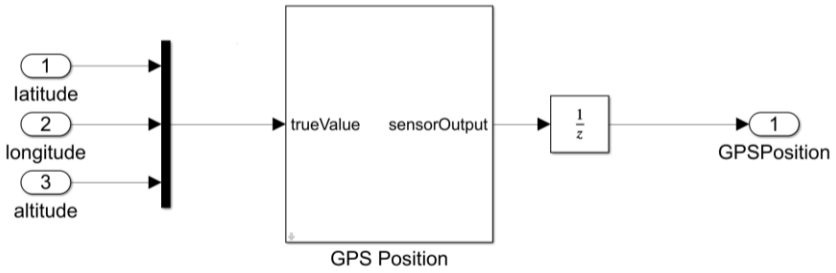

Figure 32. GPS Simulink® model (position).

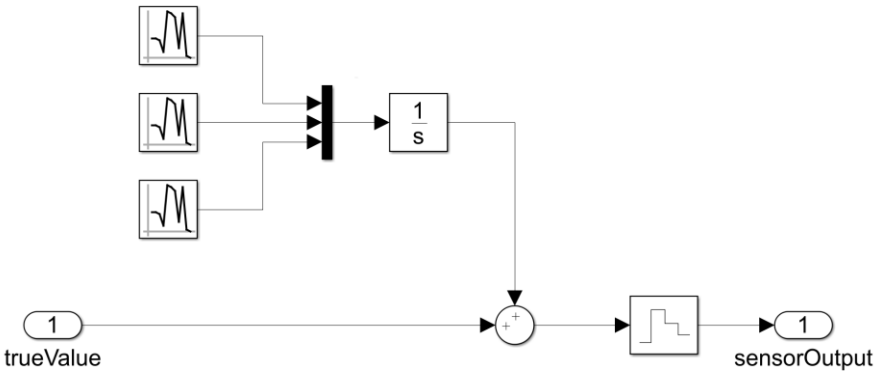

Figure 33. GPS Simulink® model.

The GPS velocity is modelled by adding a bias to each component of the simulation velocity. Then the biased velocity signal is passes through a zero-order hold and a unit

delay before giving the GPS measured velocity as output. Figure 34 and Figure 35 shows the GPS velocity Simulink® block diagram and its sensor model respectively.

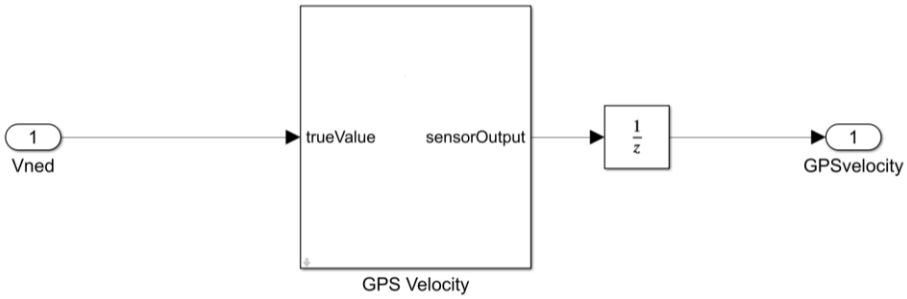

Figure 34. GPS Simulink® model (velocity).

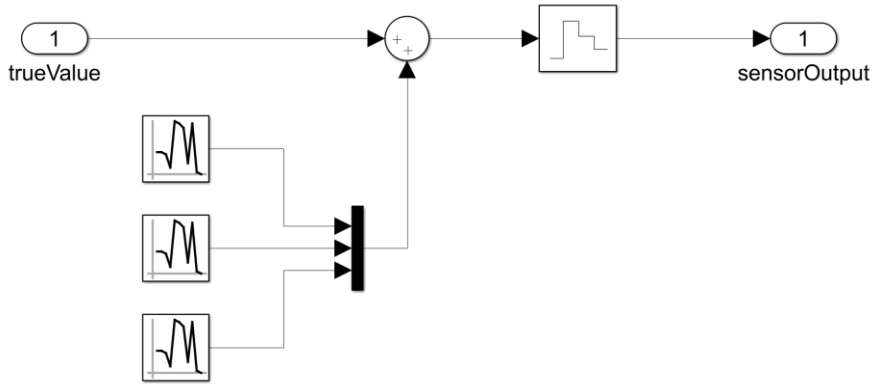

Figure 35. GPS Simulink® model velocity.

The IMU, GPS and Magnetometer work together as follows. The IMU model provides information of linear and angular acceleration $a_{b,meas}$ and $\Omega_{b,meas}$ respectively which is biased, but using a magnetometer to measure the magnetic field $M_{b,meas}$, a GPS to measure the position $P_{e,meas(GPS)}$ and ground speed $V_{e,meas(GPS)}$ as estimations of the navigation filter, a better estimate of the aircraft position P_e, linear acceleration a_b, angular acceleration $\dot{\Omega}_b$, angular rate Ω_b, and the attitude quaternion q and its respective biases

$\hat{b}_{acc}$ and $\hat{b}_{gyro}$ are obtained. The accelerometer and gyroscope estimated bias are used to perform a correction of the measurements performed by these two sensors.

5.4 Sensor Failure Detection

The sensor failure detection block subtracts the estimated airspeed $\hat{V}$ from the measured airspeed V_{meas} to compute the aircraft airspeed residual signal. The residual signal should be constrained between its standard deviation values when the sensor is working correctly. The residual signal standard deviation is used to define thresholds for normal operation by monitoring the behavior of the residual signal mean. The residual signal mean should stay between the thresholds otherwise a sensor failure must be detected. The same method is implemented for the angle of attack vane sensor failure detection.

5.4.1 Residual Signal Mean

Figure 36 shows the Simulink® Pitot sensor threshold computation and failure detection block diagram. Figure 37 shows the residual signal mean and standard deviation block diagram.

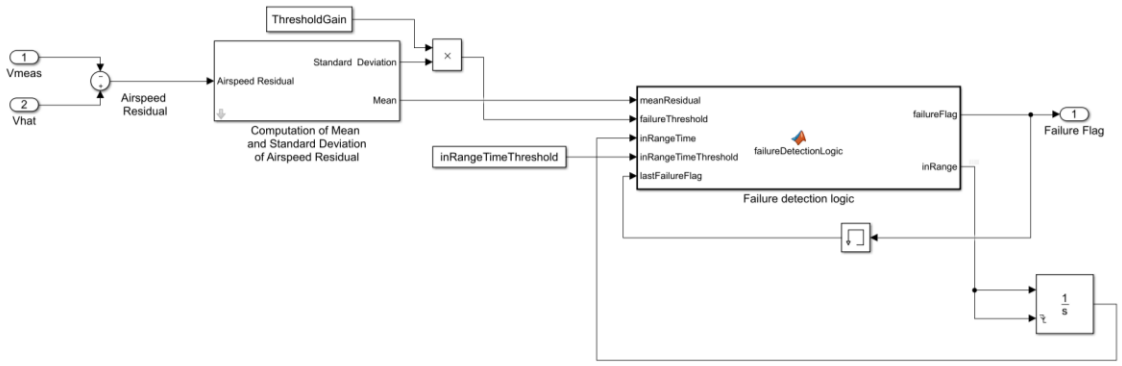

Figure 36. Pitot Sensor threshold computation and failure detection Simulink® block.

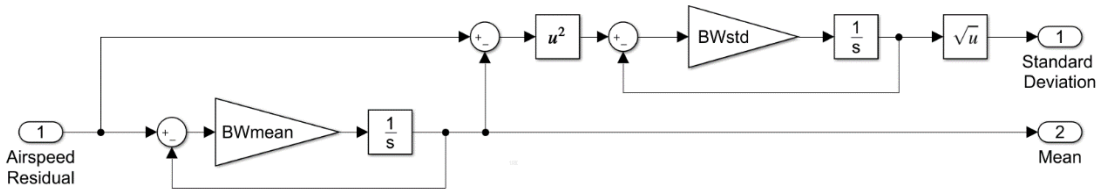

Figure 37. Pitot residual signal mean and standard deviation Simulink® Block Diagram.

5.4.2 Threshold Computation

Airspeed signal upper and lower thresholds are computed by adding and subtracting the standard deviation multiplied by a threshold gain to the mean value of the residual. The standard deviation value is multiplied by a gain to eliminate possible false detections. If the mean crosses any of the thresholds a failure should be detected. If the residual goes beyond the thresholds, then the failure flag is set to one indicating that a failure happened, if the residual is located between the thresholds then the failure flag is set to zero. This will be the input to the covariance modulation block where a failure flag will be created.

5.4.3 Failure Detection Logic

The failure detection logic is carried out by a Matlab custom function that compares the airspeed residual mean with the residual thresholds so a failure flag can be sent to tell the air data estimator to avoid taking into account the failed sensor. To be able to restore the estimation system if the sensor returns to normal operation, an internal flag called "inRange" is used to compute the time that the residual mean value stays between the residual thresholds after a failure. The "inRange" flag is "1" when the residual mean is located between the thresholds and zero if the residual mean crosses the residual thresholds. The "inRange" flag signal feeds an integrator so only when the "inRange" signal has a value of "1" the integrator outputs the time that the sensor has stayed healthy after the failure. This happens since the integrator is automatically reset when the failure stops, that is, when the failure flag changes from one to zero. The function that implements the failure detection logic resets the failure flag to zero when the time that the sensor has been healthy after the failure is greater than certain specified value which in this case is set to 60 seconds. In this way, when a sensor has failures that repeats in short periods of time, the failure flag remains set to indicate the failure, and after the sensor recovers from the failure the system can return to normal operation.

5.5 Simulink® Model of Aircraft Digital Twin

The aircraft digital twin was implemented in Simulink® using the Neuro-Fuzzy Designer toolbox. The first step was to create an initial S-FIS for each parameter using the Fuzzy-Logic designer toolbox®, later an S-ANFIS was trained to cope for with the respective data set. Figure 38 shows the block diagram for the aircraft digital twin and

Figure 39 shows the S-ANFIS for the C_L, wich is identical (except for the number of rules and membership functions) as the other parameters.

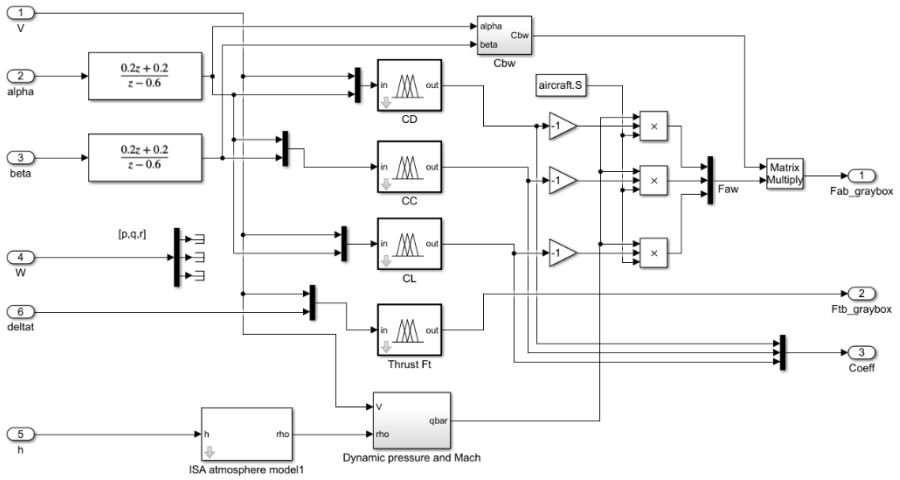

Figure 38. Simulink® Model of Aircraft Digital Twin.

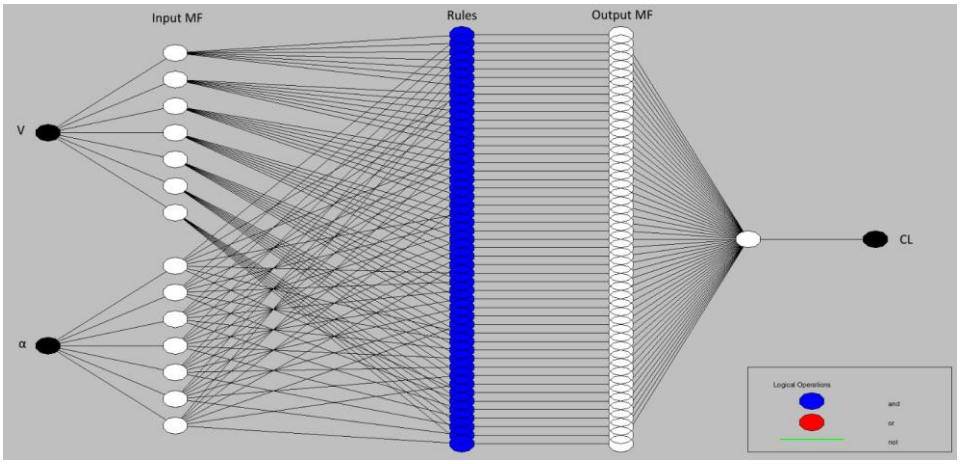

Figure 39. Simulink® model of C_L S-ANFIS

Figure 40 shows the Simulink® C_L block diagram for the S-FIS model where the crisp inputs of V and α are transformed into membership values via membership functions μ shown as Gaussian functions. The membership values are combined using the AND Fuzzy operator to obtain the firing strengths (weights) for the decision-making rules and

the qualified consequent from each rule is generated. The last step consists in aggregation (SUM) of the qualified consequents to produce a crisp output of the C_L. As shown in Figure 41, to cover the speed range of the Ryan Navion aircraft it is necessary to use 49 membership functions.

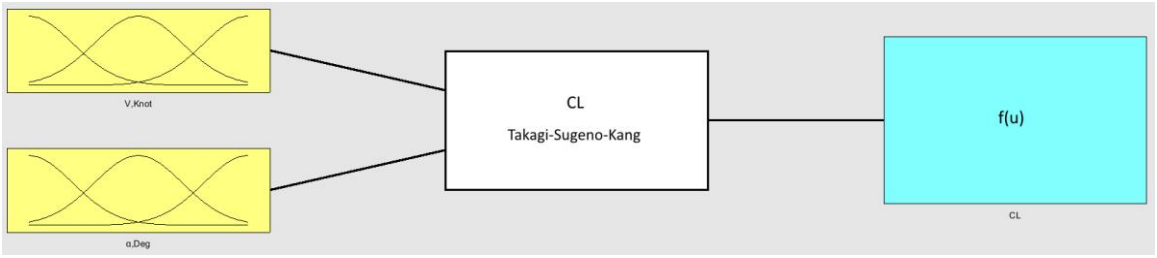

Figure 40. FIS for C_L.

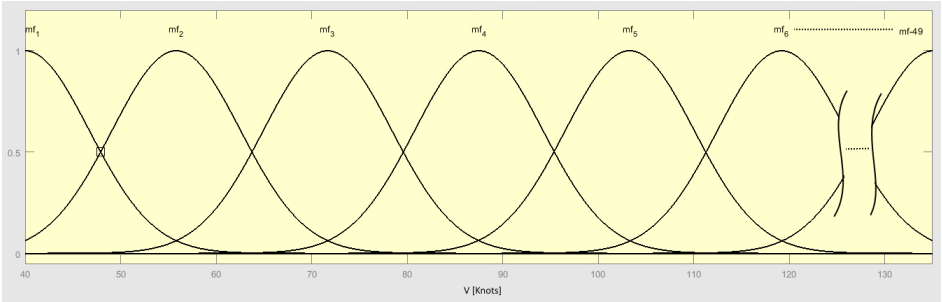

Figure 41. $\mu_A(x)$ for airspeed inputs on the C_L S-FIS.

Each ANFIS in this research was trained using simulation data generated with the Navion aircraft flight simulations. Training data sets were generated for speeds of 50, 65, 80, 95, 100, 110, 125 and 135 knots to cover the aircraft flight envelope. Each speed data set contains eleven parameters; three components of the aircraft velocity V, α, β, three components of the angular rates Ω, the throttle control, C_L, C_C, C_D and F_t, with 16,000 samples required to cover all the aircraft maneuvers during the flight simulation.

The S-FIS's for C_L, C_C, C_D and F_t were initially generated using Gaussian membership functions and by means of the hybrid learning rule each S-ANFIS was trained. During the training process, three epochs were necessary to achieve a training error of zero. The rules used in each S-FIS have the following form

If (V is in μ_1 of input 1) **and** (α is in $\mu_1$of input 2) **then** (C_L is out of μ_1 of input 1)

In this case the specific input values of V and α corresponding to their respective μ_1 are correlated with a specific output value of C_L. Here it is shown that this specific output value of C_L is only correlated to the specific input value of α but not to the specific input value of V. There will be combinations of V and α that yields a specific output value of C_L. The necessary number of rules will be established depending in the necessary input – output combinations required. The number of rules required per S-FIS are 49 for C_L, 36 for C_C, 25 for C_D and 36 for F_t. The number of μ are the same as the number of rules. Figure 42 shows the output space for the C_L S-FIS where the C_L output space is ploted for the crisp inputs V and α. Figure 43, Figure 44 and Figure 45 shows the output space for C_D S-FIS, C_C S-FIS and F_t S-FIS respectively.

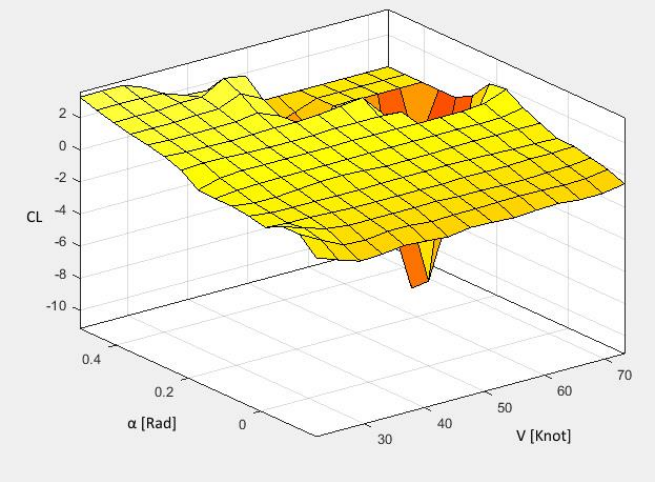

Figure 42. First order Takagi-Sugeno-Kang method of FIS.

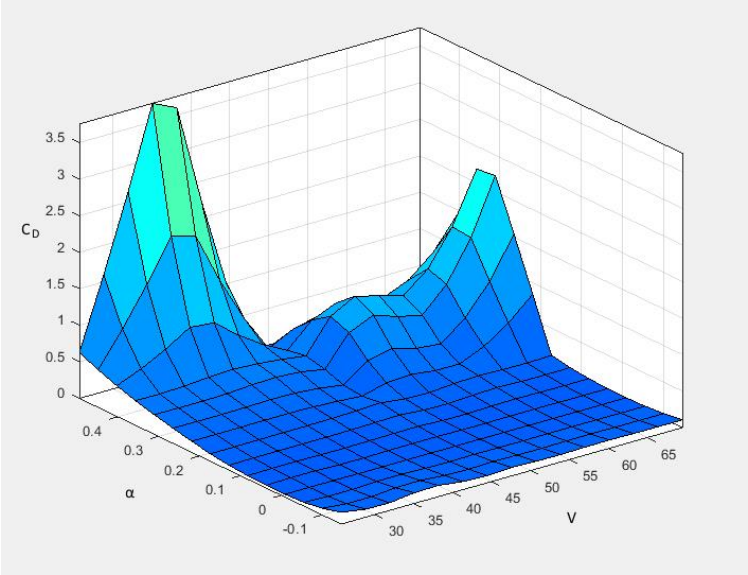

Figure 43. C_D output space for the crisp inputs V and α

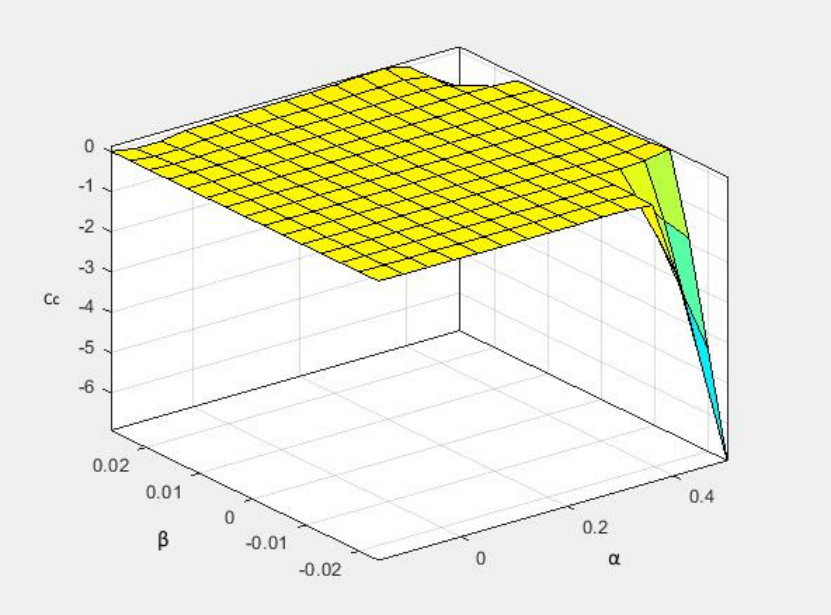

Figure 44. C_C output space for the crisp inputs α and β.

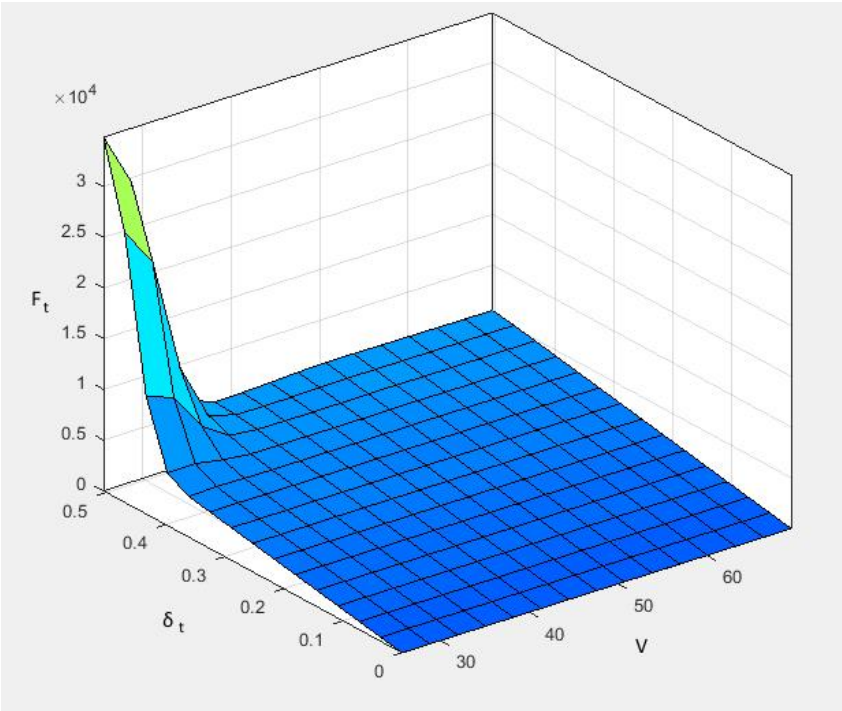

Figure 45. F_t output space for the crisp inputs V and δ_t.

CHAPTER 6. SIMULATION RESULTS

6.1 Introduction

Multiple flight simulations were performed to test the fault tolerant air data estimation system. An aircraft trajectory was carefully selected to include the common maneuvers that general aviation and commercial airplanes perform during flight so Pitot and angle of attack stuck and additive failures effects on the fault tolerant air data system can be analyzed in the complete flight envelope. The set points in the automatic control system were chosen to guide the airplane in cruise, climb, descent, turns, acceleration and deceleration maneuvers as shown in Figure 46. At the start of the flight simulation, the aircraft is cruising north at an altitude of 10.000 ft from simulation time zero until simulation time 15 sec, where the aircraft performs a turn until 45 degrees east heading is achieved. At simulation time of 40 seconds the aircraft start to climb until an altitude of 11.000 feet where the aircraft levels and returns to steady state straight and level flight for 20 seconds. At simulation time of 120 seconds the aircraft start a descent to 10.000 feet where it continues to fly in steady state straight and level flight. During the descent, at flight simulation time of 140 sec, the aircraft turns west until a heading of 20 degrees is achieved. At simulation time 300 the aircraft decelerates from 80 knots to 55 knots but flies at constant altitude. At simulation time of 500 sec the aircraft accelerates again to 80 knots. Finally, the aircraft continues in steady state straight and level flight until the simulation final time of 600 sec.

Stuck and additive Pitot tube and angle of attack vane faults were simulated during each maneuver and also during all the flight envelope. The stuck Pitot tube failure was

generated by “freezing” the total pressure value so the faulty airspeed measurement signal is generated during aircraft climb, descent, deceleration and acceleration maneuvers emulating the Pitot tube failure of recent commercial aircraft accidents. The additive Pitot failure was generated by gradually adding 70 Pascals to the Pitot signal dynamic pressure. The angle of attack vane stuck failure was simulated by freezing the value of the angle of attack signal in a particular simulation time. The angle of attack vane additive failure was generated by gradually adding 10 degrees to the angle of attack signal in a particular simulation time.

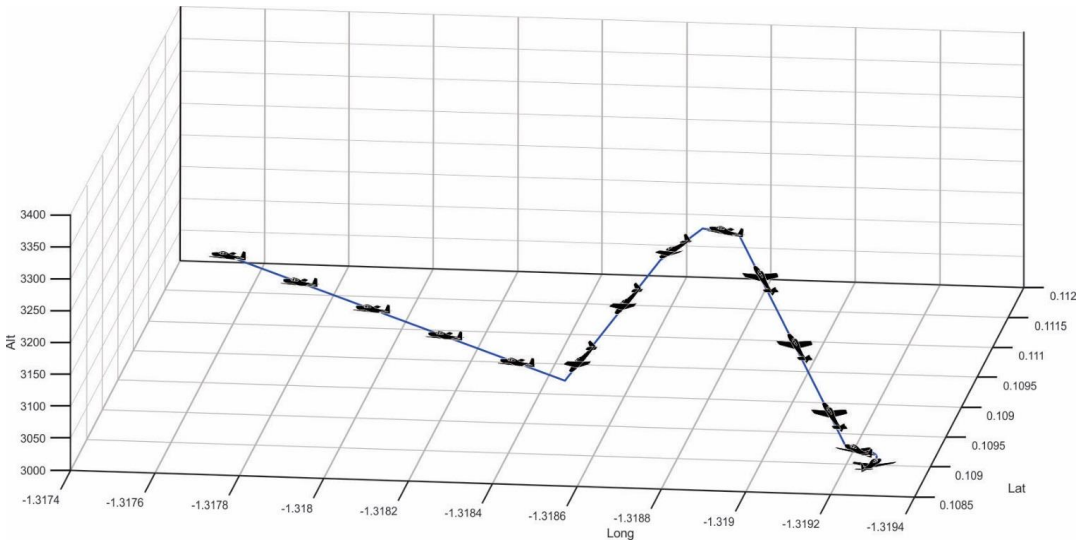

Figure 46. Aircraft trajectory and maneuvers.

The simulation air data estimations, residual signal threshold behavior, fault generation and fault detection results are presented to the reader in a single plot per air data variable. Each plot will have three graphs showing the estimation of the variable of interest, its corresponding residual signal mean and thresholds behavior and the fault generation and detection flags respectively. The estimation plots for V, β and α will show measured signals in blue, estimated signals in red and true signal values in green. The residual signal

mean and thresholds plot shows the upper threshold signal in blue, the lower threshold signal in orange, the residual signal in green and the residual signal mean in purple. The Failure flag plot (shown in red color line) is used to identify the simulation times when the failure starts (flag value of 1) and ends (flag value of 0). This plot also shows for each sensor (in blue color line) the simulation times when the failure is detected and when the sensor returns to normal operation. To get an insight of the precision in the air data estimation variables, the documents cited in Chapter 2 mentions that the airspeed estimation Root Mean Square Error (RMSE) expressed in percentage of the true airspeed value ballparked around 5% during a pitot tube additive sensor failure only. No RMSE for angle of attack vane failure and joint Pitot tube and angle of attack vane failures were encountered. Also, no RMSE data for stuck Pitot failure and angle of sideslip is available.

6.2 Stuck Sensor Failures

6.2.1 Pitot Tube Stuck Failure

Figure 47, Figure 48 and Figure 49 shows the simulation results for a Pitot tube stuck failure that was generated at simulation time of 15 seconds. To explore the behavior of the estimation system when the Pitot tube returns to normal operation, a sensor failure time duration of 185 seconds was established. The airspeed plot shows the influence of stuck pitot failure on the signal behavior. The aircraft is initially flying at steady straight and level flight (constant speed) when the failure is generated at simulation time of 15 seconds. The pitot sensor stuck failure is not detected in this instant because in steady state straight level flight there’s no change in airspeed and altitude. When the aircraft starts climbing the failure is detected because the static pressure starts to decrease with the

increase in altitude and in this instant the airspeed residual mean starts to decrease and trespasses the lower threshold, triggering the fault detection flag, where the air data estimator starts the modulation of the covariance of the airspeed sensor eliminating this measurement from the estimation process. It is worth mention that always at the moment of the failure there is a transient increase in the estimation error which is what allows the detection mechanism to detect the sensor failure. At simulation time of 200 seconds the stuck pitot failure is ended and the sensor's signal residual mean returns to normal condition inside the thresholds. When the residual signal mean reaches 60 seconds after returning within the thresholds the failure detection ends and the estimator covariance modulation stops and the system is restored to normal operation. It can be seen that estimated signals for airspeed, angle of attack and angle of sideslip follows true values accurately in the absence of the pitot. The RMS Pitot measurement error is 30 m/s corresponding to 73% of the airspeed true value, in contrast the RMS airspeed estimation error is 0.27 m/s corresponding to 0.65% of the airspeed true value which reflects and accurate estimation. The RMS Angle of attack vane error is 0.34 degrees corresponding to 4.85% of the angle of attack true value, in contrast the RMS angle of attack estimation error is 0.17 degrees corresponding to 2.4% the angle of attack true value reflecting a very low influence of the Pitot sensor failure in the angle of attack estimation.

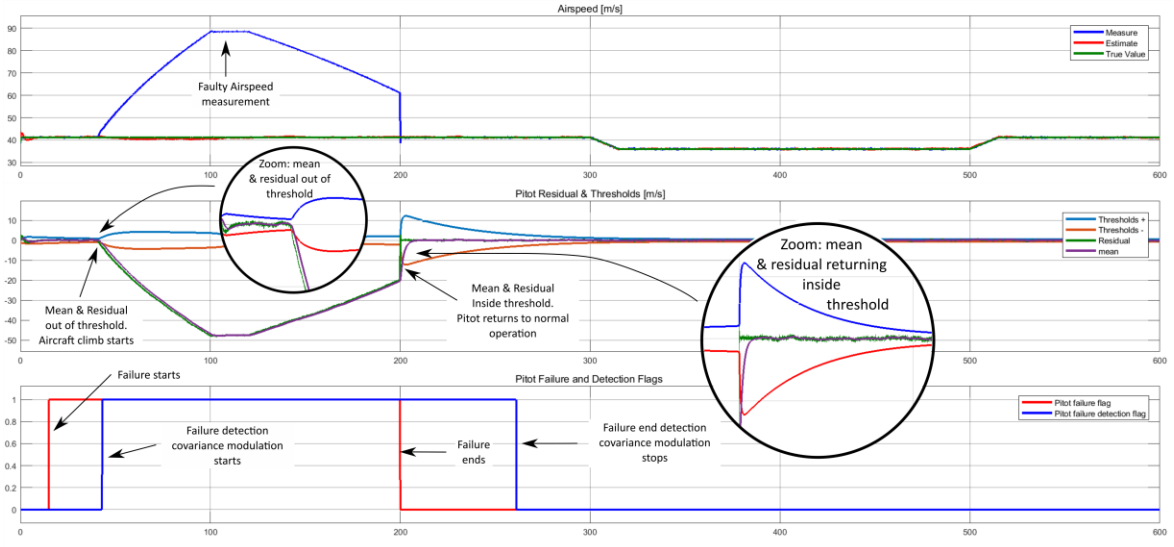

Figure 47. Pitot tube sensor stuck failure.

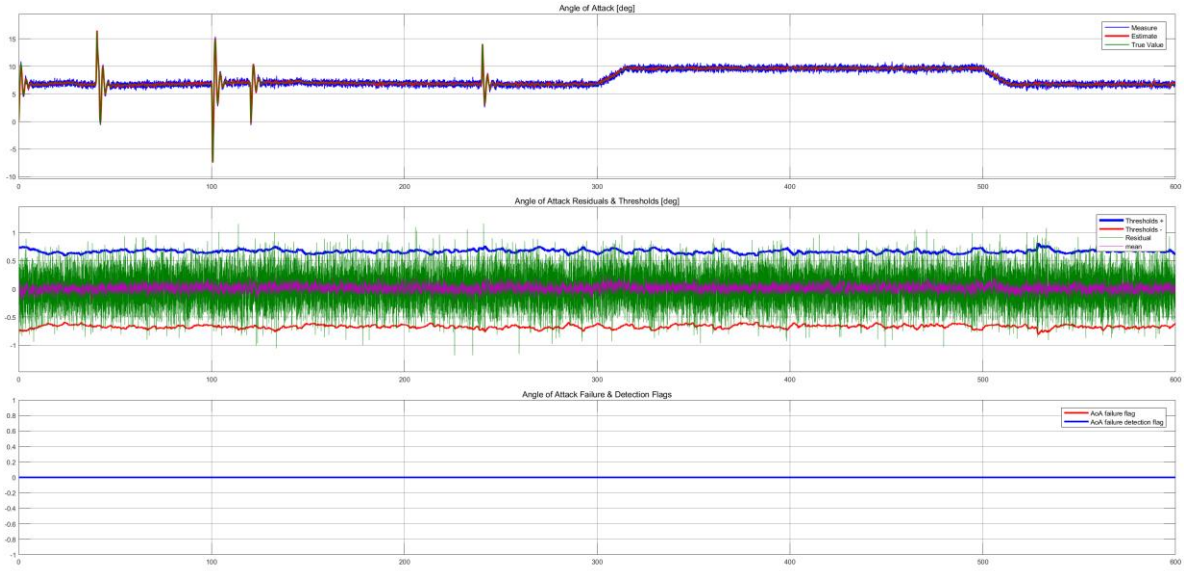

Figure 48. Angle of Attack estimation during Pitot stuck failure.

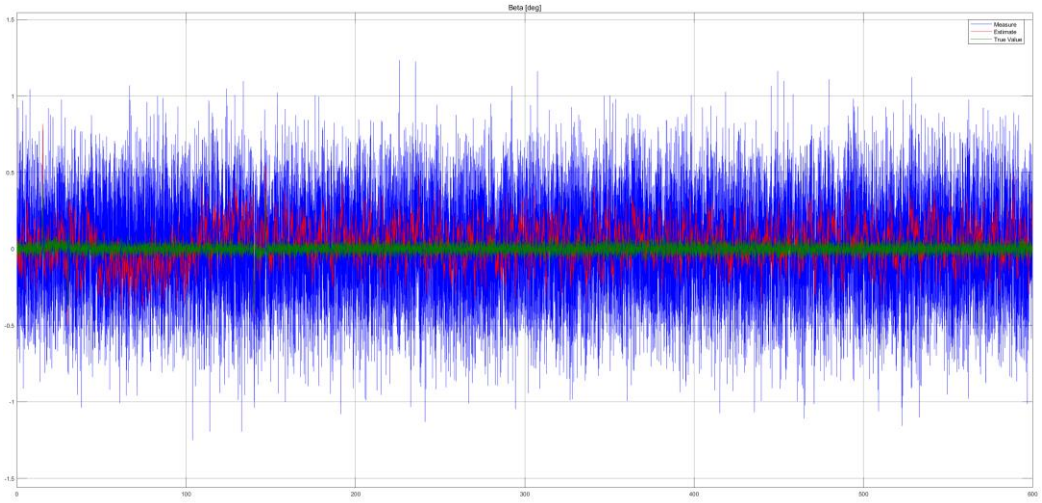

Figure 49. Angle of Sideslip estimation during Pitot stuck failure.

Figure 50, Figure 51 and Figure 52 shows simulation results where the Pitot stuck failure duration is increased to span all the maneuvers presented during flight. The airspeed residual signal mean crosses the lower threshold at the climb onset triggering the estimation covariance modulation and returns inside the threshold at the end of the descent where the aircraft returns to the initial altitude of 10.000 ft, and starts a cruise for more than 60 seconds, so the fault detection ends and the covariance modulation stops. During this cruise, the Pitot sensor is faulty, but provides the correct airspeed signal value, being this the reason for the fault detection to end until the aircraft deceleration maneuver, where the airspeed residual signal crosses the lower threshold again. At simulation time of 515 seconds when the aircraft accelerates to initial airspeed, the airspeed residual signal mean returns inside the threshold so 60 seconds later the covariance modulation stops again restoring the system to normal operation. The RMS Pitot measurement error is 30 m/s corresponding to 73% of the true airspeed value, in contrast the RMS airspeed estimation

error is 0.27 m/s corresponding to 0.65% of the true airspeed value, which is tells that the

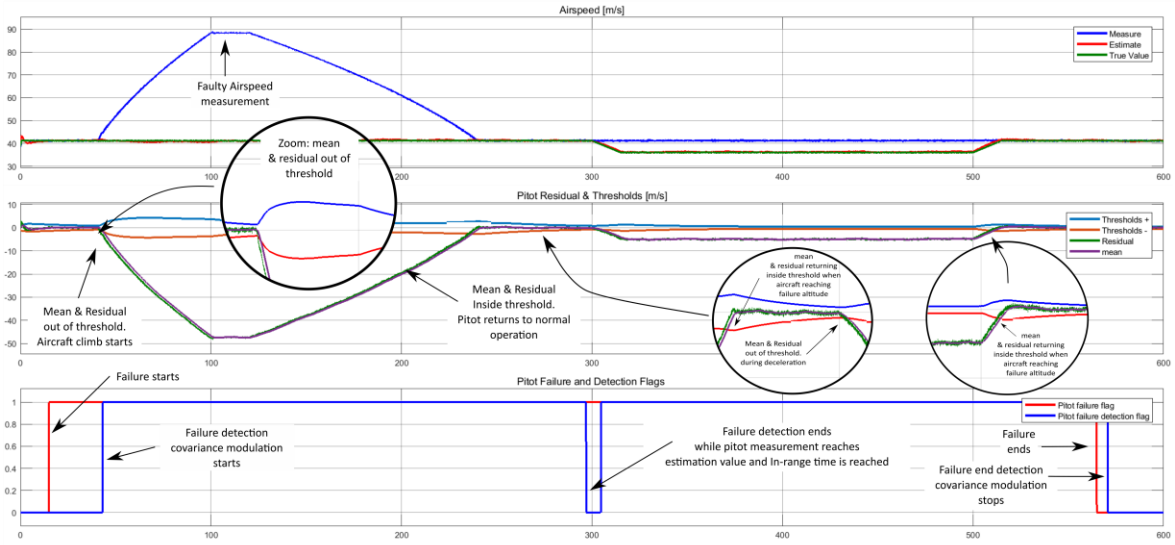

Figure 50. Pitot failure time covering all maneuvers.

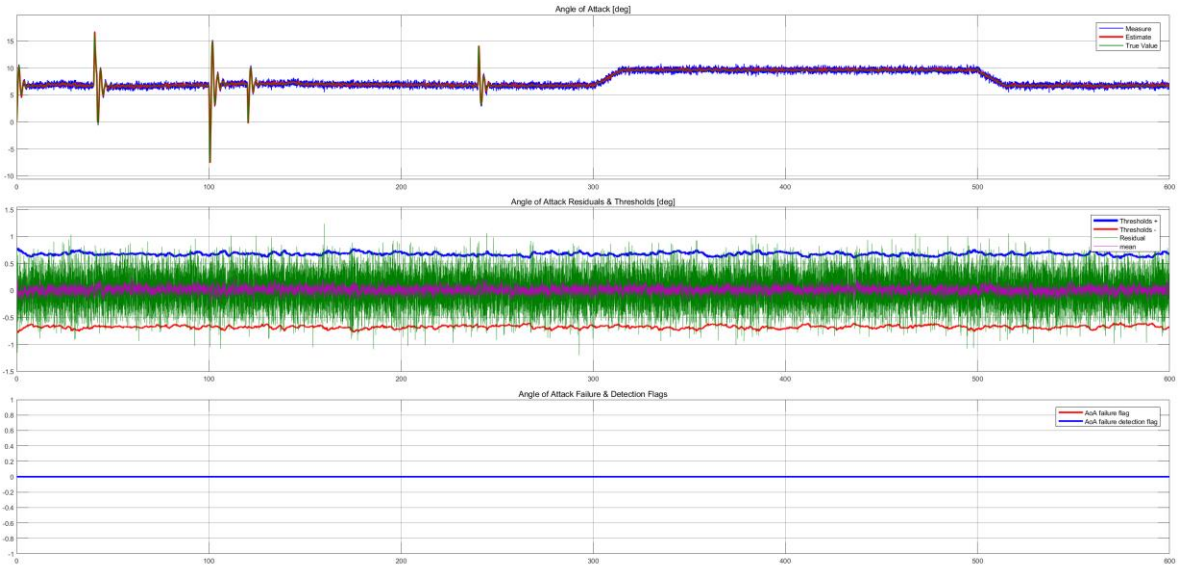

Figure 51. Angle of attack estimation during Pitot stuck failure - all the maneuvers.

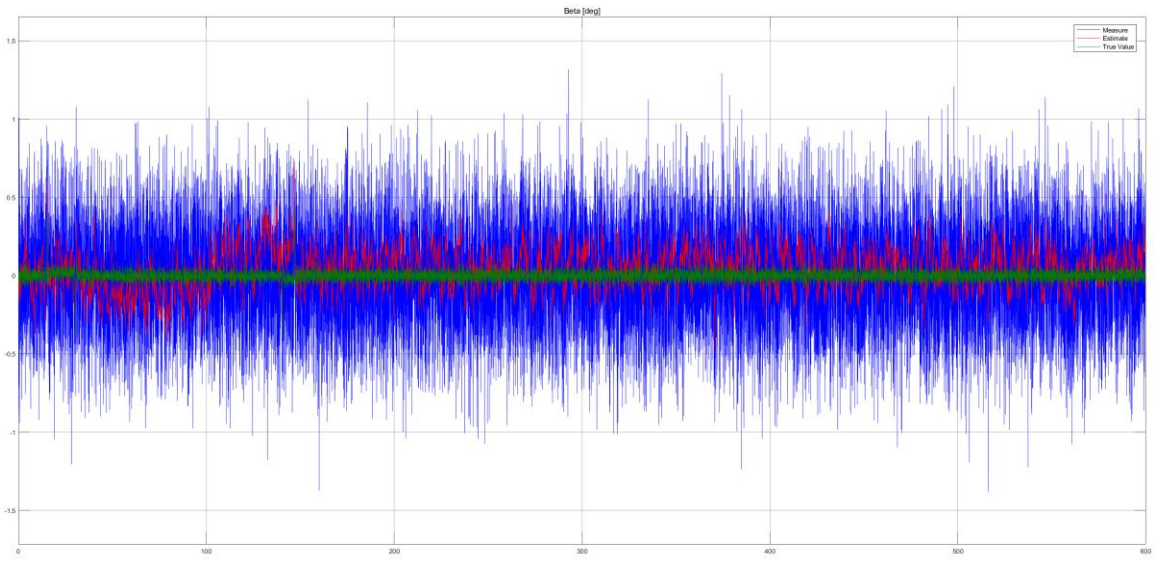

Figure 52. Angle of Sideslip estimation during Pitot stuck failure covering all maneuvers.

estimator is highly accurate in this type of fault. The RMS angle of attack vane measurement error is 0.33 degrees, corresponding to 3.3% of the angle of attack true value, whereas the RMS angle of attack estimation error is 0.19 degrees corresponding to 1.9% of the true angle of attack value, meaning that the pitot tube failure during all the maneuvers have a very low impact on the angle of attack estimation accuracy.

6.2.2 Angle of attack vane stuck failure

Figure 53, Figure 54 and Figure 55, shows an angle of attack vane stuck failure that was generated at simulation time of 300 seconds where the aircraft starts to decelerates increasing its angle of attack to compensate for the loss in coefficient of lift. The failure duration will span 230 seconds to explore the estimation system behavior during the aircraft acceleration where the angle of attack is reduced again until reaching the initial air speed. At the beginning of the maneuver the angle of attack residual signal mean increases slowly

until the simulation time of 350 seconds where the residual signal mean crosses the upper threshold very quickly and the fault detection is activated, triggering the covariance modulation of the angle of attack measurement and superposing the estimated angle of attack signal to the angle of attack true value. At simulation time of 515 seconds the acceleration maneuver reaches the initial simulation airspeed and the angle of attack mean signals returns inside the threshold range. After 60 seconds the covariance modulation stops and the system returns to normal operation. The RMS angle of attack vane measurement error is 6.07 degrees corresponding to 67.44% of the angle of attack true value, whereas the RMS angle of attack estimation error is 1.07 degrees corresponding to 11,88% of the angle of attack true value, showing a high accuracy of the air data estimator.

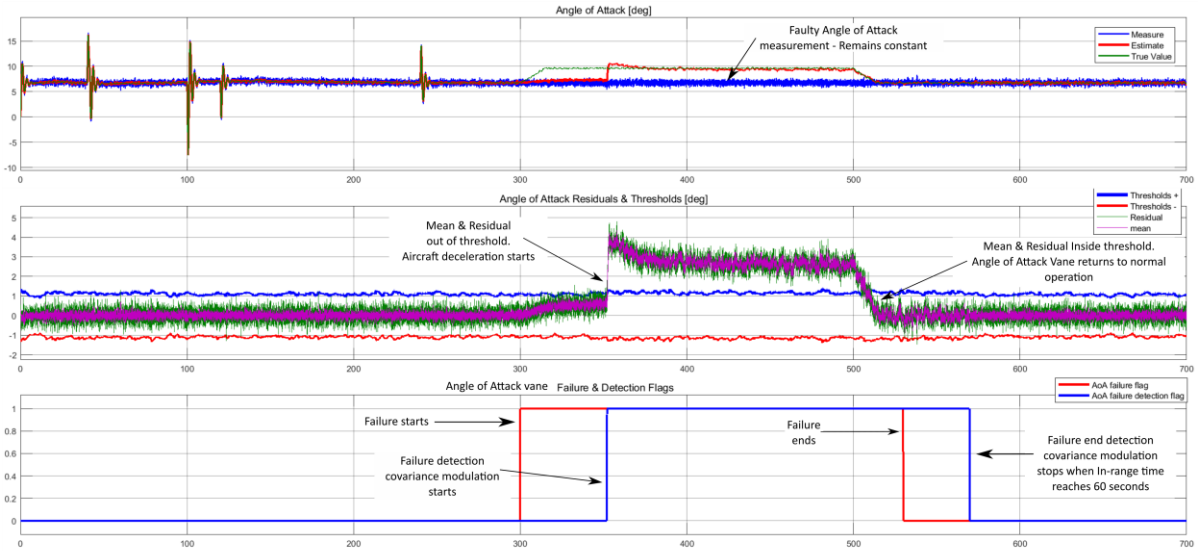

Figure 53. Angle of attack vane stuck failure.

The RMS pitot measurement error is 0.203 m/s corresponding to 0.493% of the airspeed true value, whereas the RMS airspeed estimation error is 0.131 m/s corresponding to 0.131% of the airspeed true value, showing a negligible influence of the angle of attack vane stuck failure on the airspeed estimation.

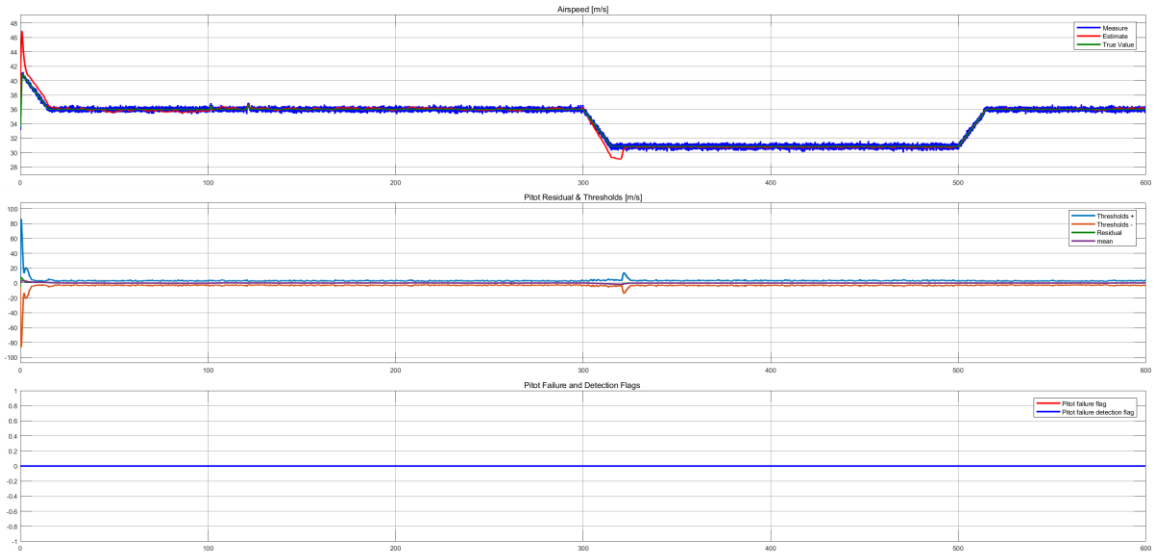

Figure 54. Airspeed estimation during angle of attack vane stuck failure.

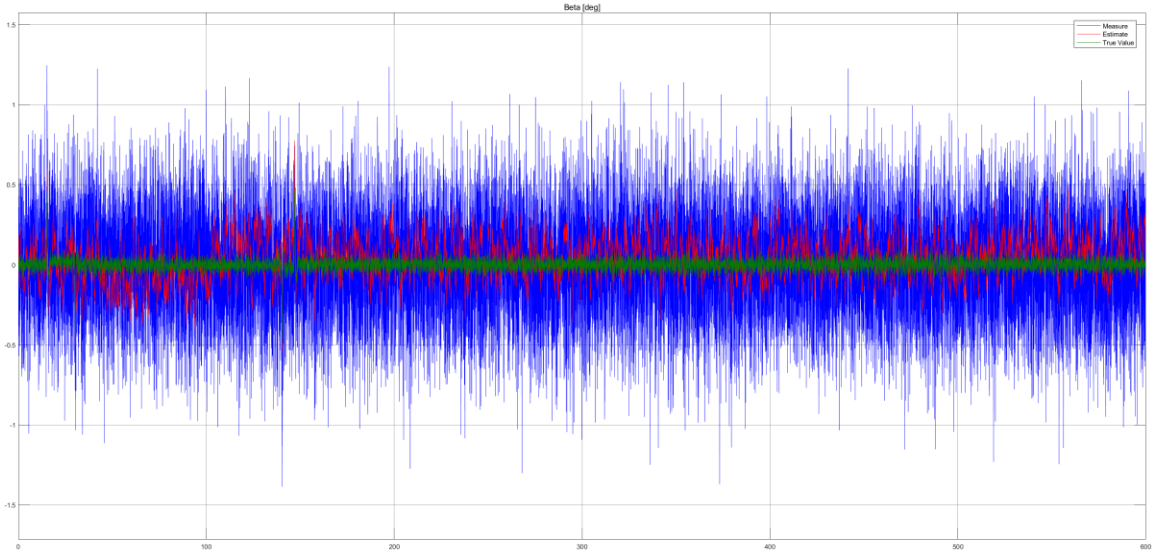

Figure 55. Angle of sideslip estimation during angle of attack vane stuck failure.

6.2.3 Joint Pitot and Angle of Attack Stuck Failures

Figure 56, Figure 57 and Figure 58 shows a joint pitot and angle of attack vane sensor stuck failures that were generated at time simulation of 20 and 70 seconds respectively. To explore the behavior of the estimation system if the sensors return to normal operation, a time duration of the pitot and angle of attack vane failure was established to 200 and 300 seconds respectively.

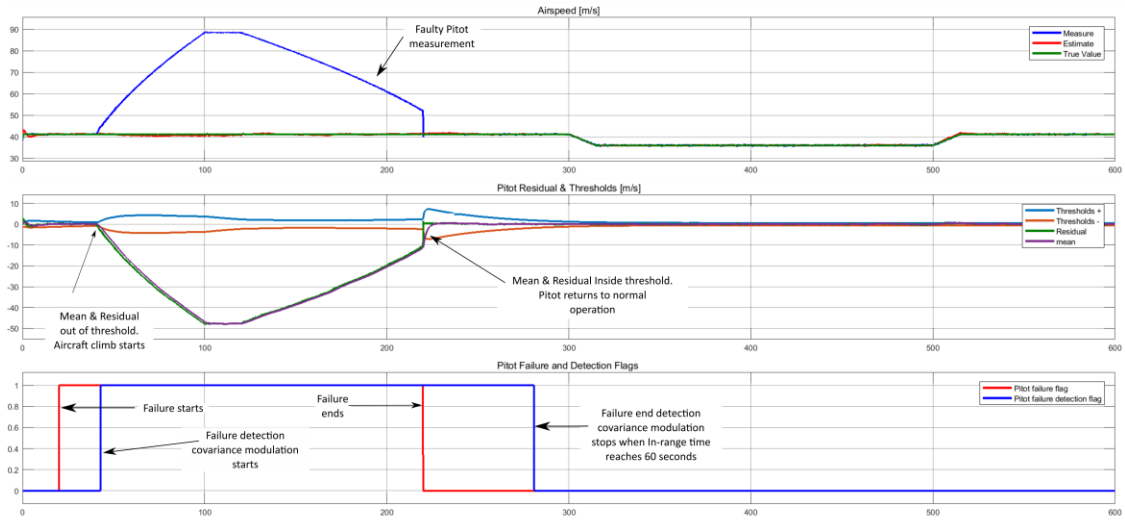

Figure 56. Airspeed estimation during joint Pitot and angle of attack vane stuck failures.

The airspeed plot shows the influence of stuck pitot failure on the signal behavior. The aircraft is initially performing steady turn (constant speed) when the stuck failure is generated at simulation time of 20 seconds. The pitot sensor stuck failure is not detected in this instant because in steady state straight level flight there's no change in airspeed and altitude. When the aircraft starts climbing the failure is detected because the static pressure

starts to decrease with the increase in altitude and here the airspeed residual signal crosses the lower threshold triggering the airspeed covariance modulation.

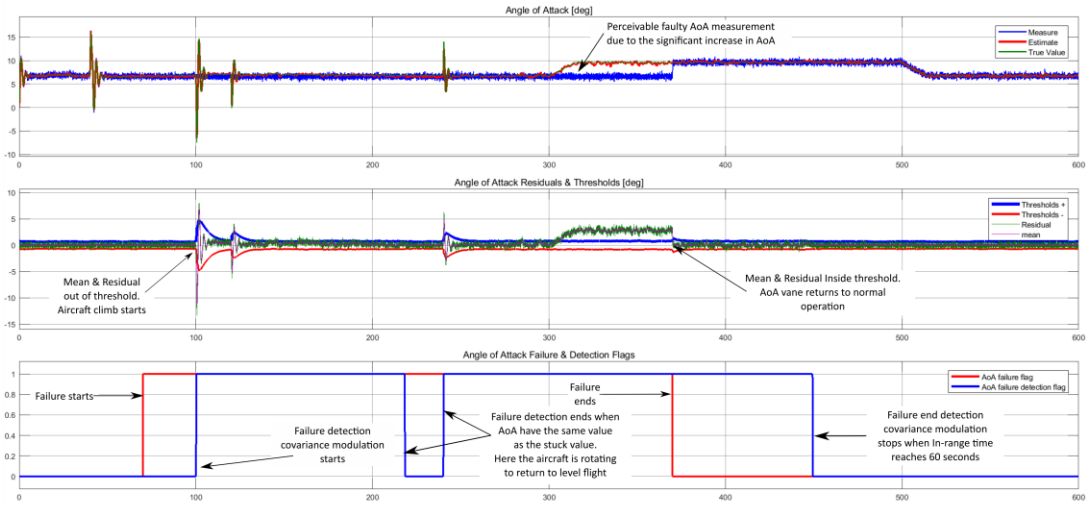

Figure 57. Angle of attack estimation during joint Pitot and angle of attack vane stuck failures.

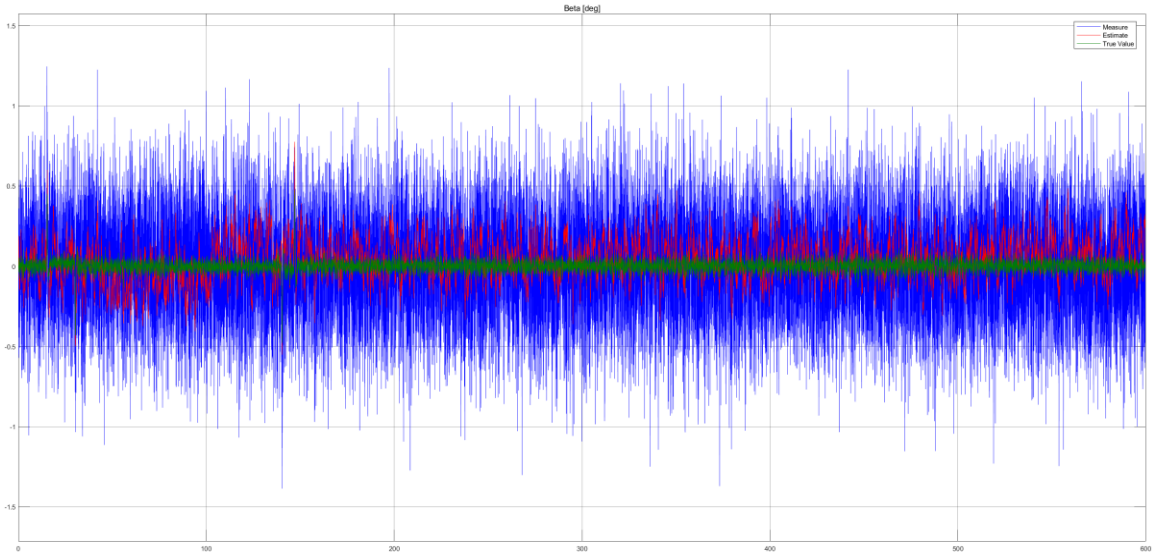

Figure 58. Angle of sideslip estimation during joint Pitot and angle of attack vane stuck failures.

The pitot fault ends at simulation time of 220 seconds and 60 seconds later the airspeed covariance modulation stops and the sensor's signal returns to normal condition. During these aircraft maneuvers there's only a small variation on the aircraft's angle of attack true value from the stuck angle of attack measured signal so the behavior of the angle of attack residual mean is difficult to perceive. In contrast, during the deceleration, the angle of attack residual signal mean crosses the upper threshold triggering the covariance modulation until the failure ends at simulation time of 370 seconds. Sixty seconds later the failure detection stops and the system returns to normal operation. The RMS Pitot measurement error is 29.1 m/s corresponding to 70,7% of the airspeed true value, whereas the RMS airspeed estimation error is 0.44 m/s corresponding to 1.06% of the true airspeed value, showing a high accuracy in the airspeed estimation. The RMS angle of attack vane measurement error is 1.01 degrees corresponding to 11,2% of the angle of attack true value, whereas the RMS angle of attack estimation error is 0.19 degrees corresponding to 2.1% of angle of attack true value, also showing an accurate estimation of the angle of attack.

Figure 59, Figure 60 and Figure 61, simultaneous Pitot and angle of attack vane stuck failures where generated at simulation times of 295 and 300 seconds respectively. The pitot and angle of attack vane stuck failure durations are 230 and 220 seconds respectively. Here the deceleration maneuver takes place and the automatic flight control increases the angle of attack to compensate for the reduction in the aircraft's coefficient of lift, so the difference between measured and true angle of attack signals is clearly perceived. The measured angle of attack signal value is kept constant by the stuck failure of the vane but true value increases. The airspeed measurement signal remains constant because during the deceleration and acceleration maneuver, the aircraft flies at constant

altitude. The airspeed residual signal mean crosses the lower threshold during deceleration triggering the covariance modulation of the airspeed measurement. The airspeed residual signal mean returns inside the threshold once the aircraft accelerates to the initial airspeed. On the other hand, the angle of attack residual mean behaves in a similar fashion to the previous case. The RMS Pitot measurement error is 5.62 m/s corresponding to 15.6% of airspeed true value, whereas the RMS airspeed estimation error is 2.9 m/s corresponding to 8% of airspeed true value, still an accurate airspeed estimation. The RMS angle of attack vane measurement error is 6.34 degrees corresponding to 70% of the angle of attack true value, whereas the RMS angle of attack estimation error is 0.8 degrees corresponding to 8.8% of the angle of attack true value, which is still a reasonable estimation accuracy.

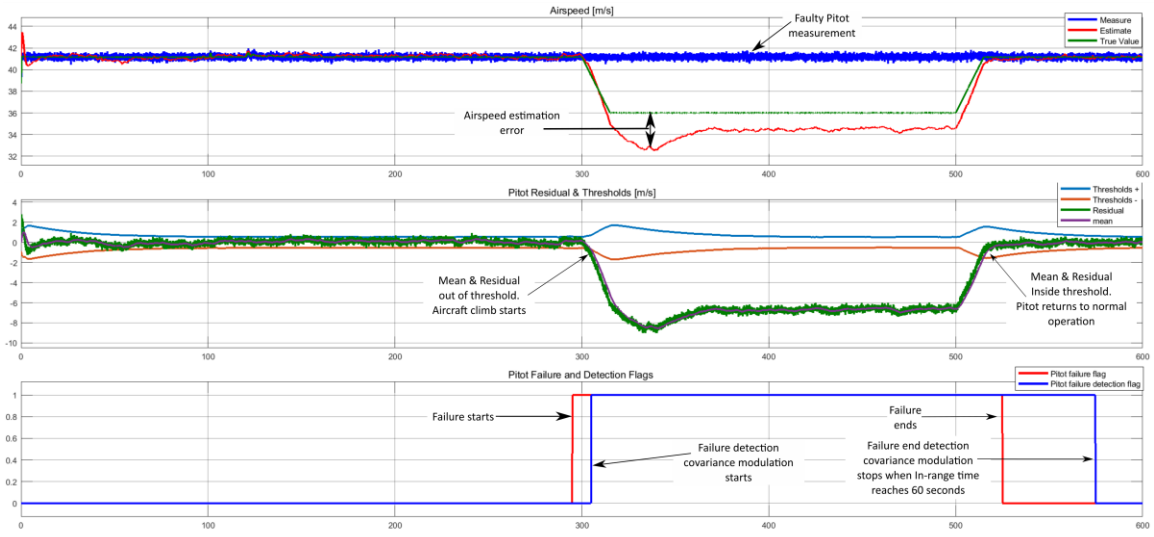

Figure 59. Airspeed estimation during joint Pitot and angle of attack sensor stuck failures during deceleration and acceleration.

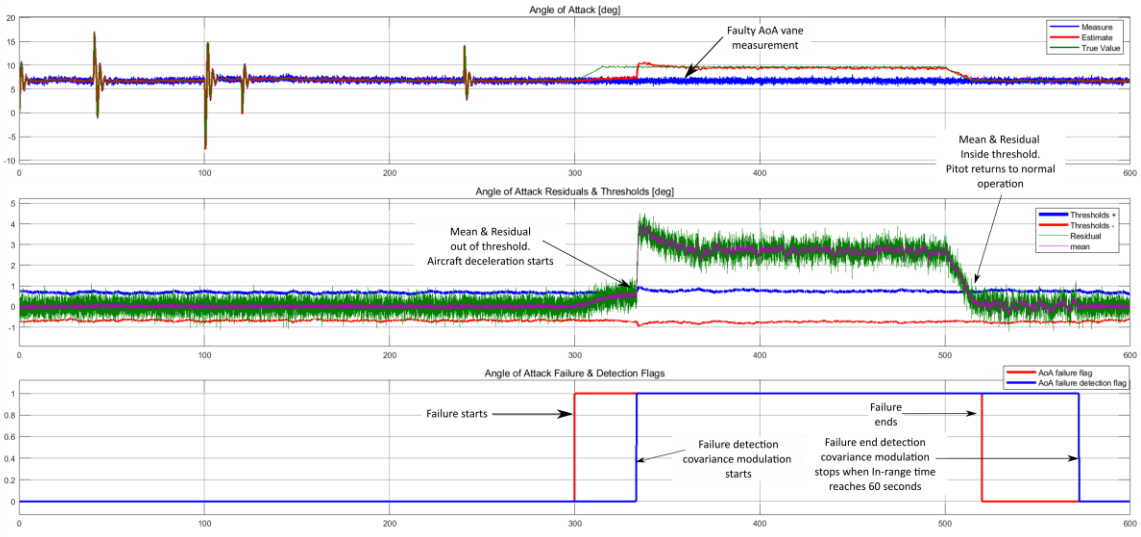

Figure 60. Angle of attack estimation during joint Pitot and angle of attack sensor stuck failures during aircraft deceleration and acceleration.

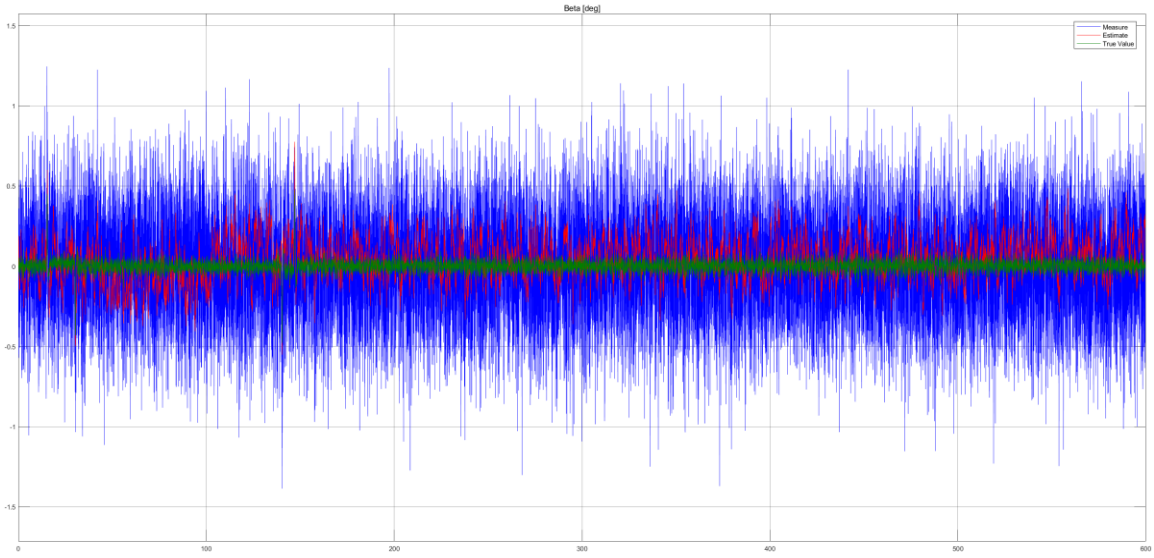

Figure 61. Angle of sideslip estimation during joint Pitot and angle of attack sensor stuck failures during aircraft deceleration and acceleration.

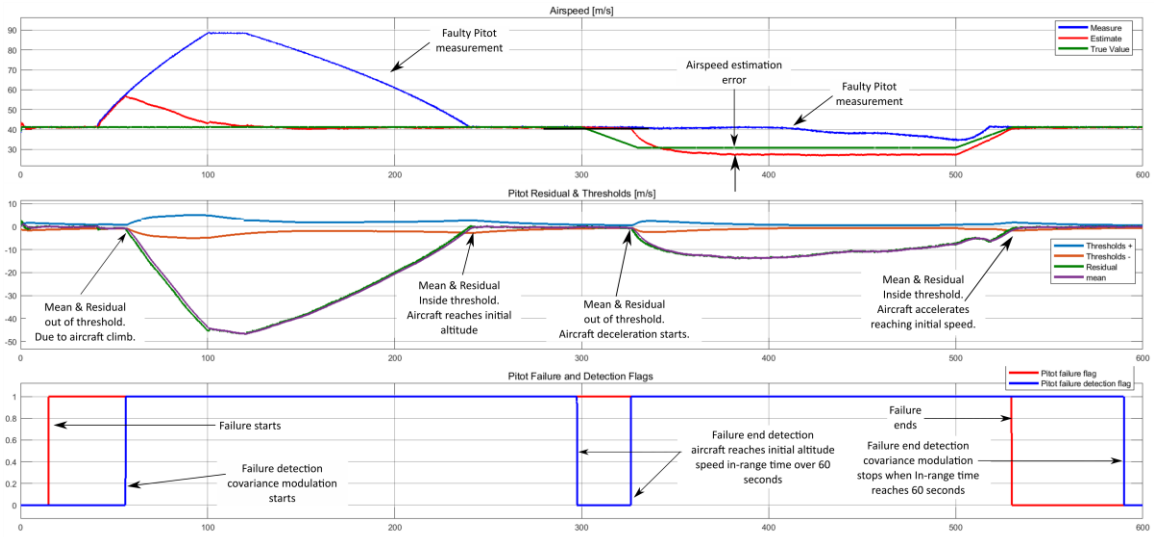

Figure 62. Airspeed estimation in a joint Pitot and angle of attack sensor stuck failures during all maneuvers.

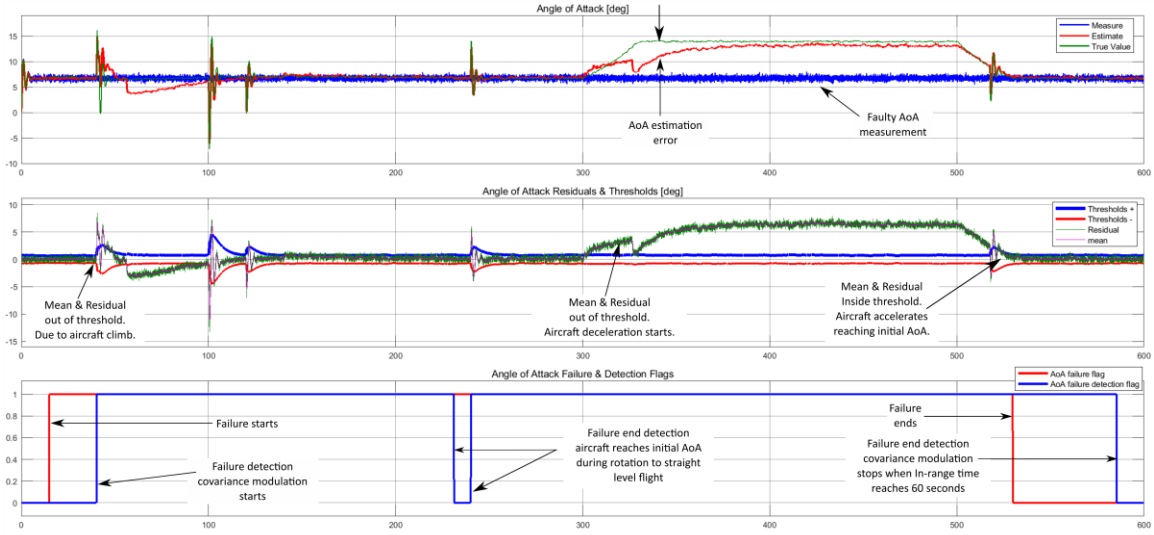

Figure 63. Angle of attack estimation in a joint Pitot and angle of attack sensor stuck failures during all maneuvers.

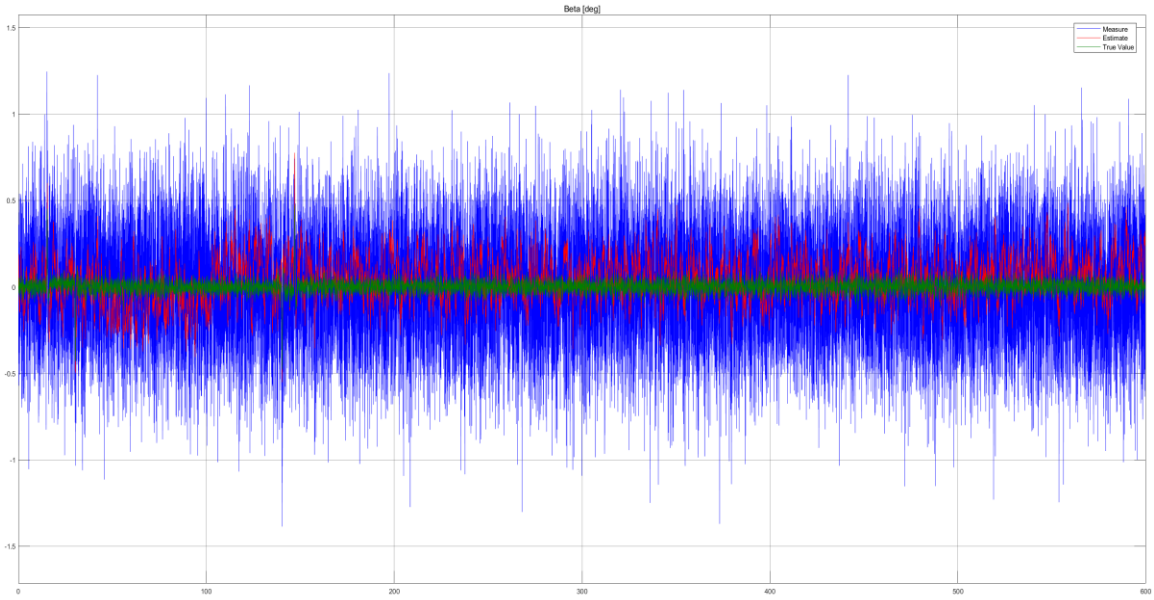

Figure 64. Angle of sideslip estimation in a joint Pitot and angle of attack sensor stuck failures during all maneuvers.

Figure 62, Figure 63 and Figure 64 shows a joint pitot and angle of attack vane stuck failure generated at simulation time of 15 seconds where the aircraft is flying at steady straight flight. In this simulation the aircraft is flying across all the maneuvers and the estimation is being performed without the aid of the pitot and angle of attack vane. The residual signal mean and fault detection for the airspeed and the angle of attack have a similar behavior as explained previously. The RMS Pitot measurement error is 20.08 m/s corresponding to 48.8% of airspeed true value, whereas the RMS airspeed estimation error is 2.33 m/s corresponding to 5.6% of airspeed true value. The RMS angle of attack vane measurement error is 4.61 degrees corresponding to 33% of the angle of attack true value, whereas the RMS angle of attack estimation error is 1 degree corresponding to 7.14% of the angle of attack true value. The air data estimator can predict accurately the air data variable in the absence of pitot tube and angle of attack during stuck failures.

6.2.4 Summary of Stuck Sensor Failure Estimation Results.

Table 16 summarizes the results of the air data estimator during the stuck airspeed and angle of attack vane failures for the different aircraft maneuvers.

Table 16. Summary of stuck failure estimation results.

Stuck Failure									
Sensor Failure	Maneuver	RMSE							
		V_{meas} (m/s)	V_{meas} $\%V$	$\hat{V}$ (m/s)	$\hat{V}$ $\%V$	α_{meas} (deg)	α_{meas} $\%\alpha$	$\hat{\alpha}$ (deg)	$\hat{\alpha}$ $\%\alpha$
Pitot	Cruise, turn, climb	30	72.89	0.270	0.656	0.34	4.857	0.17	2.43
	all	30	72.89	0.270	0.656	0.33	3.3	0.19	1.9
α vane	Acceleration, cruise, deceleration	0.203	0.49	0.131	0.318	6.07	67.444	1.07	11.8
Joint Pitot and α vane	Pitot: cruise, turn, climb, descent. α vane: climb, descent, cruise, deceleration	29.1	70.7	0.44	1.069	1.01	11.222	0.19	2.11
	deceleration, cruise, acceleration	5.62	15.6	2.9	8.056	6.34	70.444	0.8	8.88
	all	20.08	48.7	2.33	5.661	4.61	32.929	1	7.14

6.3 Additive Sensor Failures

6.3.1 Pitot Tube Additive Failure

The Pitot tube additive failure is generated at simulation time of 15 seconds with a duration time of 515 seconds to cover all aircraft maneuvers. Figure 65, Figure 66 and Figure 67 shows the airspeed, angle of attack and angle of sideslip estimation and fault detection plots respectively. The airspeed residual signal mean crosses the lower threshold at simulation time of 60 seconds, triggering the covariance modulation that isolate the pitot tube measurement in the airspeed estimation. The airspeed estimated signal gradually follows the true airspeed signal during all the flight. At simulation time of 530 seconds the Pitot tube failure ends and the system returns to normal operation once the airspeed residual

mean signal stay between the thresholds for 60 seconds. The RMS Pitot measurement error

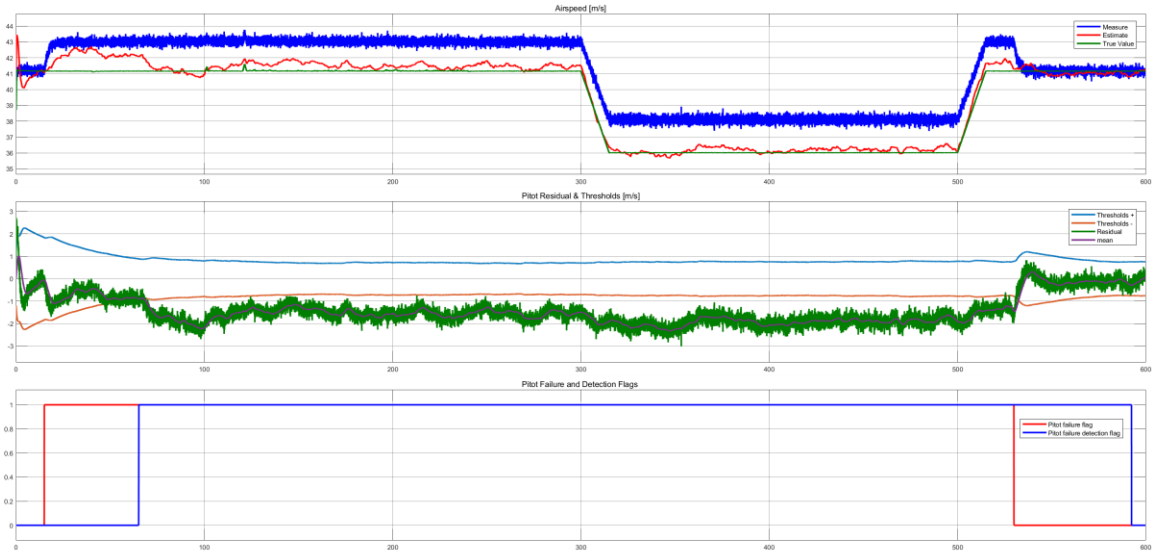

Figure 65. Airspeed estimation during Pitot tube additive failure.

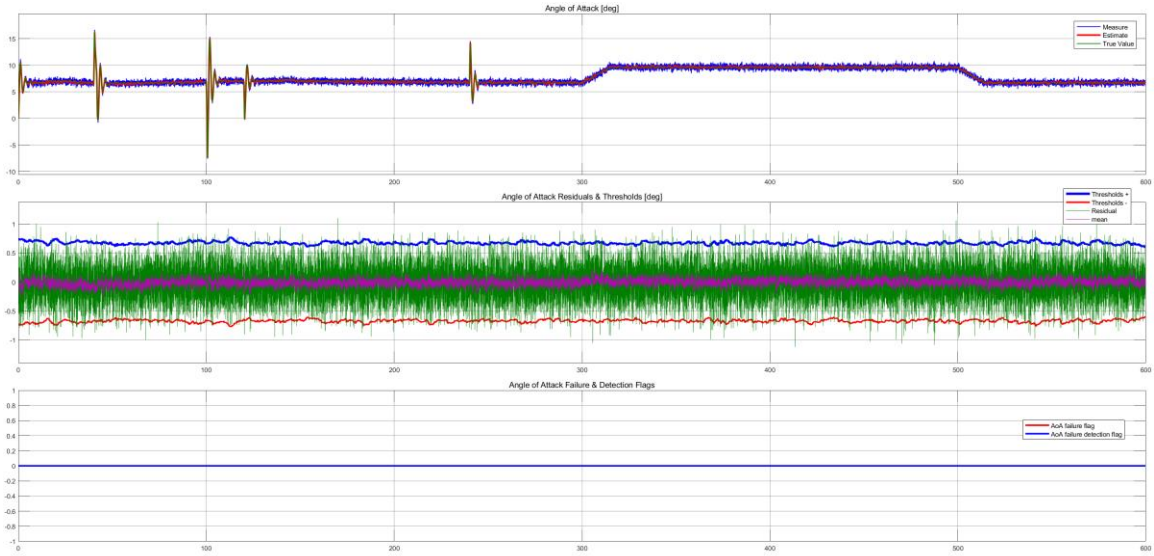

Figure 66. Angle of attack estimation during Pitot tube additive failure.

is 1.8520 m/s corresponding to 4.5% of airspeed true value, whereas the RMS airspeed estimation error is 0.3176 m/s corresponding to 0.77% of airspeed true value, which is a

high estimation accuracy. The RMS angle of attack vane measurement error is 0.3259 degrees corresponding to 4.65% of the angle of attack true value, whereas the RMS angle of attack vane estimation error is 0.1511 degrees corresponding to 2.15% of the angle of attack true value, which represents a high estimation accuracy.

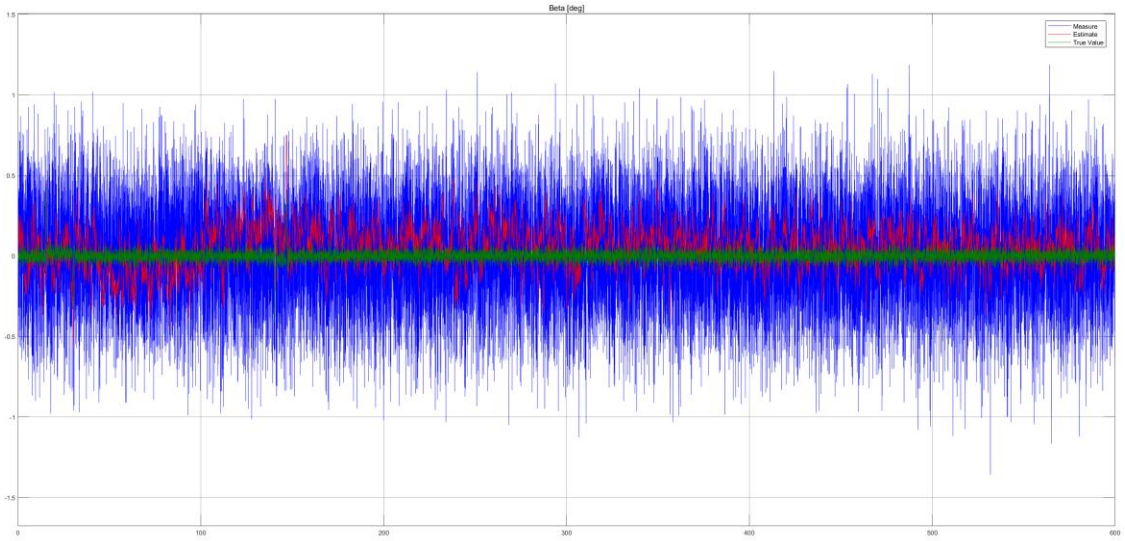

Figure 67. Angle of sideslip estimation during Pitot tube additive failure.

6.3.2 Angle of Attack Vane Additive Failure

In this case, an angle of attack vane additive failure of 10 degrees is generated at simulation time of 300 seconds with a duration time of 230 seconds to cover the aircraft deceleration and acceleration maneuvers. Figure 68, Figure 69 and Figure 70, shows the angle of attack, airspeed and angle of sideslip estimation and fault detection plots respectively. At the beginning of the aircraft deceleration, the angle of attack residual signal trespasses the lower threshold, triggering the sensor fault detection and starting the covariance modulation in the air data estimator. When the aircraft accelerates to the initial flight speed, the angle of attack residual signal mean returns inside the thresholds and 60

seconds later, the fault detection ends and the system returns to normal operation. The airspeed signal, and estimation seems unaffected by the angle of attack failure. The RMS Pitot measurement error is 0.1726 m/s corresponding to 0.42% of airspeed true value, whereas the RMS airspeed estimation error is 0.1784 m/s corresponding to 0.43% of airspeed true value, reflecting a high accuracy of the airspeed estimation. The RMS angle of attack vane measurement error is 8.707 degrees corresponding to 62.2% of the angle of attack true value, whereas the RMS angle of attack vane estimation error is 0.9891 degrees corresponding to 7% of the angle of attack true value, reflecting an acceptable accuracy of the angle of attack estimation.

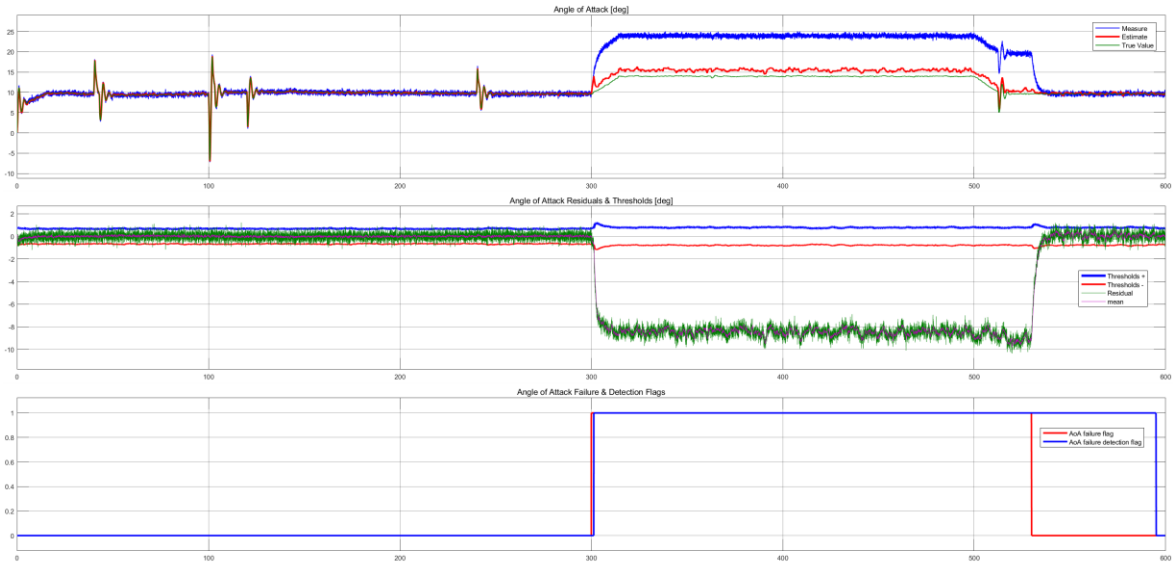

Figure 68. Angle of attack estimation during angle of attack vane additive failure.

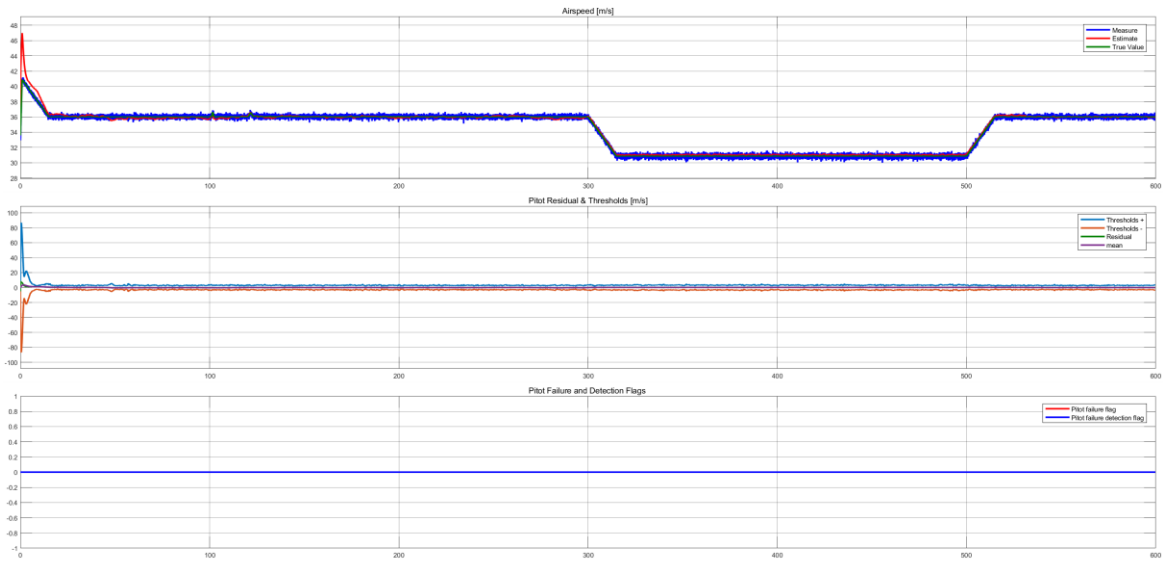

Figure 69. Airspeed estimation during angle of attack vane additive failure.

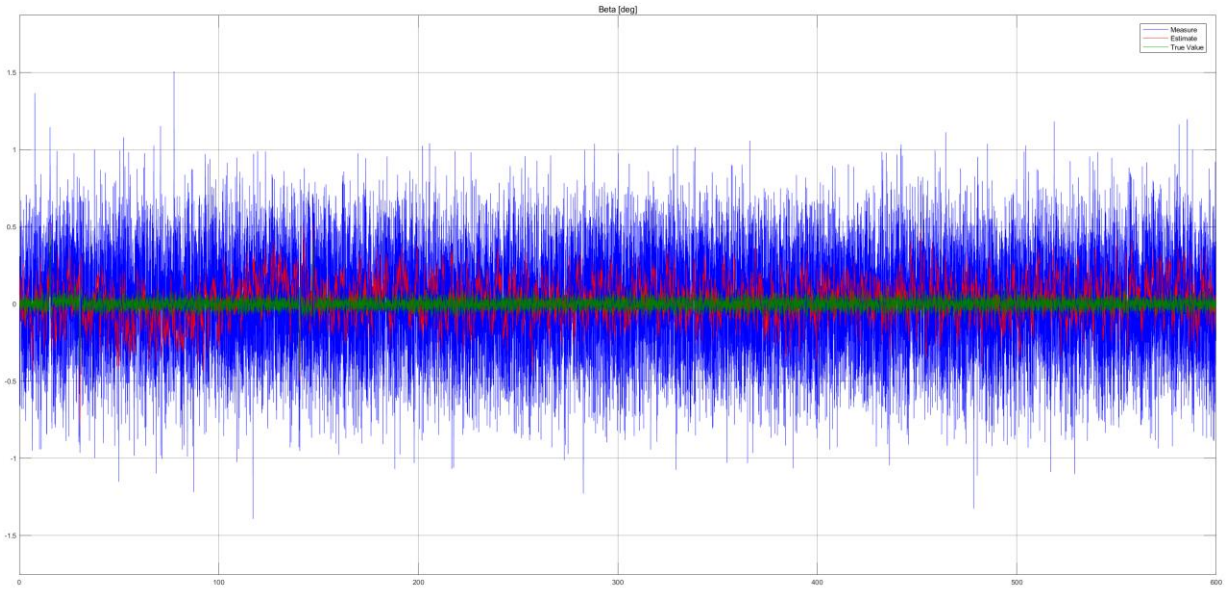

Figure 70. Angle of sideslip estimation during angle of attack vane additive failure.

6.3.3 Joint Pitot and Angle of Attack Vane Additive Failures

In this case, a joint Pitot tube and angle of attack vane additive failures are generated. The Pitot additive failure is generated at simulation time of 15 seconds with a

duration time of 515 seconds whereas the angle of attack additive failure is generated at simulation time of 300 seconds with a duration time of 230 seconds. Figure 71 , Figure 72 and Figure 73 shows the airspeed, angle of attack and angle of sideslip estimation and fault detection plots respectively. The RMS Pitot measurement error is 1.8 m/s corresponding to 4.4% of airspeed true value, whereas the RMS airspeed estimation error is 0.328 m/s corresponding to 0.8% of airspeed true value, showing an accurate airspeed. The RMS angle of attack vane measurement error is 4.38 degrees corresponding to 87.6% of the angle of attack true value, whereas the RMS angle of attack vane estimation error is 0.2319 degrees corresponding to 4.6% of the AoA true value, which reflects an accurate estimate of the angle of attack.

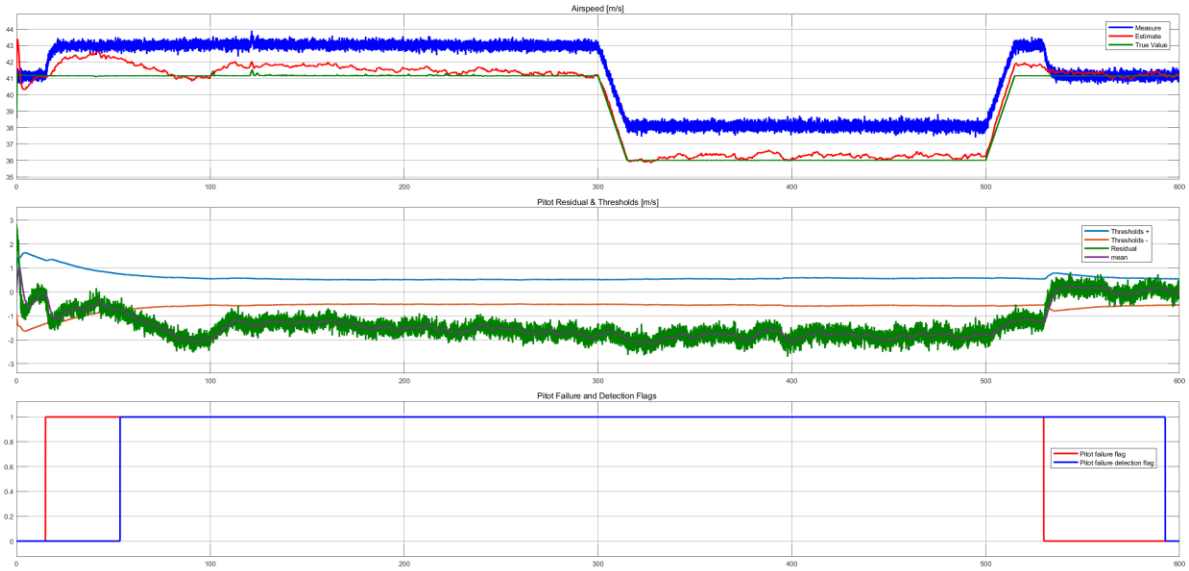

Figure 71. Airspeed estimation during joint Pitot and angle of attack vane additive failures.

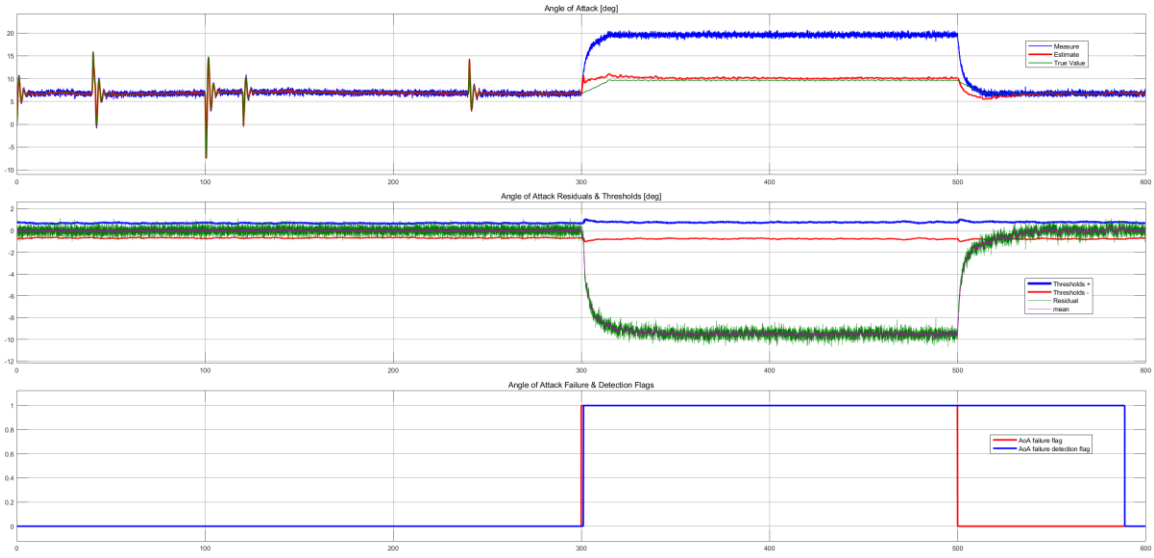

Figure 72. Angle of attack estimation during joint Pitot and angle of attack vane additive failures.

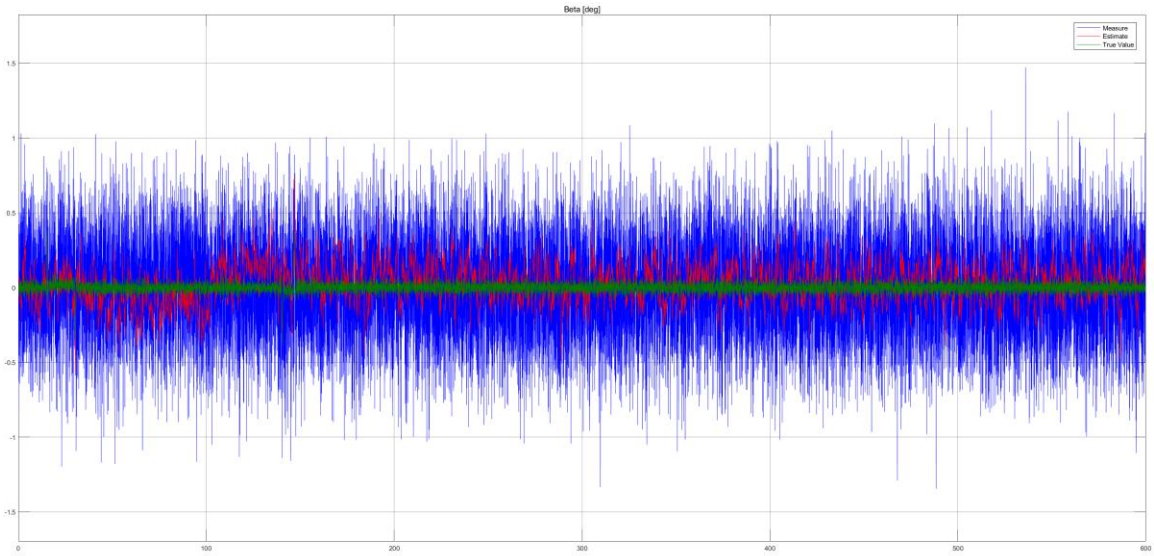

Figure 73. Angle of sideslip estimation during joint Pitot and angle of attack vane additive failures.

6.3.4 Summary of Additive Sensor Failure Estimation Results.

Table 17 summarizes the results of the air data estimator during the additive airspeed and angle of attack vane failures for the different aircraft maneuvers.

Table 17. Summary of additive sensor failure results.

Additive Failure									
Sensor Failure	Maneuver	RMSE							
		V_{meas} (m/s)	V_{meas} $\%V$	$\hat{V}$ (m/s)	$\hat{V}$ $\%V$	α_{meas} (deg)	α_{meas} $\%\alpha$	$\hat{\alpha}$ (deg)	$\hat{\alpha}$ $\%\alpha$
Pitot	All	1.85	4.5	0.32	0.77	0.33	4.66	0.15	2.16
α vane		0.18	0.43	0.18	0.43	8.71	62.19	0.99	7.07
Joint Pitot and α vane		1.8	4.4	0.328	0.8	4.38	87.6	0.2319	4.6

6.4 Observability Analysis During Simulations.

Figure 74 shows the results of the air data estimator observability analysis. This graph is the same for any type and combination of failures presented in this chapter. It can be inferred that the observability matrix H remains in full rank (7 states) during the 600 seconds of the simulation independently of the type and combination of sensor failures.

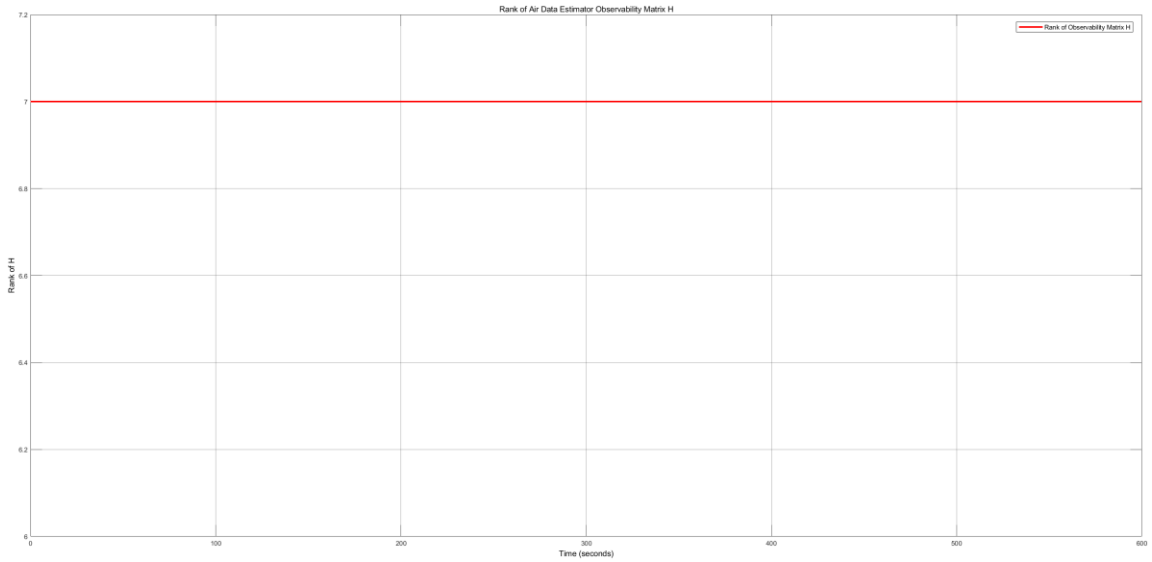

Figure 74. Rank of air data estimatorobservability matrix H.

After analyzing the flight simulation results it can be concluded that the developed fault tolerant air data estimation system is capable of predicting the airspeed and angle of attack with high accuracy when the Pitot and angle of attack vane sensors are subjected to independent and joint stuck and additive failures.

CHAPTER 7. CONCLUSIONS

In this research a novel fault tolerant air data system that is able to cope with stuck and additive Pitot sensor failure was developed. The fault tolerant air data estimation scheme can easily be extended to the rest of the sensors in the air data system as it was done with the angle of attack vane sensor. The system allows the implementation of sensor redundancy; however, this was not accomplished due to the absence of a powerful computer to perform the simulations. A Ryan Navion aircraft flight simulation model constructed with actual flight test and sensor data was used to test the fault tolerant air data estimation system accuracy and the sensor fault detection system during independent and joint Pitot and angle of attack vane sensor stuck and additive failures. The system showed the capability of capturing the aircraft dynamics and detect unhealthy sensors. The air data estimation system is capable of returning to normal operation if the faulty sensors become healthy again as it will be the case if ice covering the pitot or angle of attack melts.

A novel sensor fault detection scheme is developed in this research minimizing false sensor fault detections and undetected sensor failures, being relatively simple to tune. The failure flag provided by the sensor failure detection module feedbacks the air data estimation algorithm were the unhealthy sensor noise covariance is modulated eliminating its effect in the air data estimation accuracy, being a major contribution in the field. A virtual sensor of aerodynamic and propulsion forces was developed to help in the estimation process when the Pitot and angle of attack vane sensors fails. This virtual sensor is based on a grey-box numerical model that captures the aircraft dynamics at the different

flight speeds and during the common flight maneuvers and air turbulence that is presented in flight.

The air data estimation is performed during the common maneuvers encountered in the operation of commercial and general aviation airplanes, allowing the pilot to land safely the aircraft without measurements of airspeed and angle of attack and only with the estimations.

CHAPTER 8. FUTURE WORK

The air data system developed in this research can be improved as follows:

- The number of Pitot and angle of attack sensors can be increased to assess the accuracy of the system whe only one sensor type fails but extra similar sensors remain healty.
- The aircraft digital twin can be augmented to include all the variables that have a strong influence in the aerodynamic and thrust force coefficients and also include the effects in flaps and retractable landing gear. For this to be accomplished, it will be necessary to perform flight test with an instrumented aircraft and collect the necessary data to develop a high fidelity aircraft dynamic model.
- A second upgrade to the system will be to compute the fault detection parameters during each maneouvers to tailor the fault detection behaviour to each phase of the flight.
- The fault tolerant air data estimation system can be adapted to run in real time on an embeeded processor so system tests can be performed during an actual aircraft in flight.
- The effects of the fault tolerant air data system on the way a human pilot safely operates the aircraft can be tested in a flight simulator.
- It will be useful to perform a robustenes and confiability analysis

REFERENCES

[1] Air Safety Foundation, "McDonnell Douglas DC-9-21," Air Safety Network, 30 January 1973. [Online]. Available: https://aviation-safety.net/database/record.php?id=19730130-1. [Accessed 3 6 2020].

[2] National Tansportation Safety Board, "Aircraft Accident Report, Northwest Airlines, Inc, Boeing 727-251, N274US, Near Thielle, New York, December 1, 1974.," NTIS, Washington D.C., 1975.

[3] Aviation Safety Foundation, "Lear Jet 25B," Aviation Safety Network, 28 July 1984. [Online]. Available: https://aviation-safety.net/database/record.php?id=19840728-0. [Accessed 3 6 2020].

[4] Aviation Safety Foundation, "Tupolev Tu-154B-2," Aviation Safety Network, 21 May 1986. [Online]. Available: https://aviation-safety.net/database/record.php?id=19860521-2. [Accessed 3 6 2020].

[5] National Transportation Safety Board, «Aircraft Accident Report, Runway Overrun Following Rejected Takeoff, Continental Aairlines Flight 795, McDonnell Douglas

MD-82, N18835, LaGuardia Airport, Flushing, New York, March 2, 1994.,» NTSB, Washington D.C., 1995.

[6] Junta Investigadora de Accidentes Aereos, «Final Aviation Accident Report, Birgenair Flight ALW-301, Puerto Plata, Dominican Republic.,» JIAA, Puerto Plata, 1996.

[7] Aviation Safety Foundation, "Boeing 757-23A Aeroperu," Aviation Safety Network, 2 Octubre 1996. [Online]. Available: https://aviation-safety.net/database/record.php?id=19961002-0&lang=es. [Accessed 3 6 2020].

[8] Direccion General de Aviacion Civil, «INFORME FINAL DE ACCIDENTE DE AVIACION, McDONNELL DOUGLAS DC - 9-32, AUSTRAL, LINEAS AEREAS CIELOS DEL SUR S.A.,» C.I.A.D.A., Montevideo, 1997.

[9] Aviation Safety Foundation, "Boeing 737-4Q8," Flight Safety Network, 7 April 1999. [Online]. Available: https://aviation-safety.net/database/record.php?id=19990407-0&lang=es. [Accessed 3 6 2020].

[10] Flight Safety Foundation, "McDonnell-Douglas-MD-11," Flight Safety Network, 17 October 1999. [Online]. Available: http://aviation-safety.net/database/record.php?id=19991017-0&lang=es. [Accessed 3 6 2020].

[11] Flight Safety Foundation, "Dornier 328JET-300," Flight Safety Net, 3 June 2006. [Online]. Available: https://aviation-safety.net/database/record.php?id=20060603-1. [Accessed 3 6 2020].

[12] Flight Safety Network, "Boeing 737 Turkish Airlines," Flight Safety Foundation, 25 February 2009. [Online]. Available: https://aviation-safety.net/database/record.php?id=20090225-0&lang=es. [Accessed 03 06 2020].

[13] Flight Safety Foundation, "Airfrance A330," *AeroSafety World,* vol. 5, no. 8, p. 68, 2010.

[14] Federal Air Transport Agency, "On the Results of the Investigation of the Aircraft Accident An-148-100V RA-61704 Saratov Airlines," Federal Air Transport Agency, Moscow, 2018.

[15] D. Koenig, "AP News," AP, 5 November 2018. [Online]. Available: https://apnews.com/0497c5e2bceb42d99d7d7ad1530a71c9. [Accessed 3 6 2020].

[16] Airbus, "Unreliable Airspeed," *Safety First,* no. 5, p. 40, 2007.

[17] Le Bureau d'Enquêtes et d'Analyses (BEA) pour la Sécurité, "On the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro - Paris," BEA, Paris, 2012.

[18] Airbus, «How Airbus research aims to make future flight even safer,» de *Final OPTICS dissemination event*, Brussels, 2017.

[19] Airbus, «FAST-A380 Special edition,» *Airbus technical magazine*, vol. A380 Special Edition, nº 2, p. 64, 2016.

[20] ATSB, «Erratic airspeed indications, Boeing 787-8, VH-VKE,» ATSB, Canberra, 2015.

[21] W. Gracey, «Measurement of Aircraft Speed and Altitude,» Wiley, Ann Arbor, 1981.

[22] SAE, «Air Data Computer - Minimum Performance Standards,» SAE, Evanston, 1996.

[23] G. Seidel, D. Cronin, J. Mette, M. Koosmann, J. Schmitz, J. Fedele and D. Kromer, "Multi-function Air Data Sensing Probe Having an Angle of Attack Vane". US, DE, EP Patent US20040261518A1, 13 9 2005.

[24] Aviation Collector, "F-16 AIR DATA PROBE," 18 7 2018. [Online]. Available: http://www.aviation-collector.be/images/adp/4.jpg. [Accessed 3 6 2020].

[25] Collins Aerospace, "Multi-Function Probes," UTC Aerospace systems, Brunsville, 2020.

[26] C. Belcastro, J. Foster, G. Shah, I. Gregory and D. Cox, "Aircraft Loss of Control Problem Analysis and Research Toward a Holistic Solution," *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS*, vol. 40, no. 4, 2017.

[27] D. W. Hempe and J. D. Seipel, *Approval of Non-Required Angle of Attack (AoA) Indicator Systems, Memorandum No. AIR100-14-110-PM01*, FAA, 2014.

[28] National Aaeronautics and Space Administration, "Advanced Air Data Systems for Commercial Aircraft," Spinoff NASA, 3 6 2020. [Online]. Available: https://spinoff.nasa.gov/Spinoff2006/T_3.html. [Accessed 2020 6 2020].

[29] T. Katsibas, T. Semertzidis, X. Lacondemine and N. Grammalidis, "Signal processing for a laser based air data system in commercial aircrafts," in *2019 27th European Signal Processing Conference (EUSIPCO)*, Laussane, 2008.

[30] NRL, «Optical Air Data System Flight Testing,» NRL, Munich, 2012.

[31] BAE Systems, "BAE Systems develops laser airspeed sensor for aircraft," BAE Systems, 9 8 2016. [Online]. Available: https://www.baesystems.com/en/article/bae-systems-develops-laser-airspeed-sensor-for-aircraft. [Accessed 3 6 2020].

[32] Airbus, "Airbus Group Completes Successful Flight Tests of Laser-Based Airspeed Sensor System," 18 12 2016. [Online]. Available: https://bef7441f-6df7-4fbe-809c-

2a76f8f14d9b.filesusr.com/ugd/d4c014_d7a5922467ff4681a2f62dcb1ba15a40.pdf. [Accessed 4 6 2020].

[33] J. Kurtz, S. Wittig and S. O'Byrne, "A Counter-Propagating Laser Air Speed Sensor System for Aircraft," in *31st AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, Dallas, 2015.

[34] M. Napolitano, S. Cascianelli, S. Gururajan, S. Rhudy and M. L. Fravolini, "Experimental Evaluation of Two Pitot Free Analytical Redundancy Techniques for the Estimation of the Airspeed of an UAV," *SAE International Journal of Aerospace*, vol. 7, no. 2014-01-2163, pp. 109-116, 2014.

[35] M. Napolitano, M. Rhudy, M. Chao, M. Fravolini and S. Gururajan, "Performance Evaluation of Neural Network Based Approches for Airspeed Sensor Failure Accomodation on a Small UAV," in *21st Mediterranean Conference on Control & Automation (MED)*, Platanias-Chania, Crete, Greece, 2013.

[36] S. Hussain, M. Mokhtar and J. Howe, "Sensor Failure Detection Identification and Accommodation Using Fully Connected Cascade Neural Network," *IEEE Transactions on Industrial Electronics*, vol. 62, pp. 1683-1692, 2015.

[37] I. Turkmen and S. Arik, "New Alternative Air Data Computation Method Based on Artificial Neural Networks," *Journal of Aeronautics and Space Technologies,* vol. 10, no. 1, pp. 21-29, 2017.

[38] A. Alireza, P. Aboutalebi, K. Yen and A. Sargolzaei, "Neural adaptive observer-based sensor and actuator fault detection in nonlinear systems: Application in UAV.," *ISA Transactions,* vol. 67, p. 317–329, 2017.

[39] J. Qi, X. Zhao, Z. Jiang and j. Han, "An Adaptive Threshold Neural-Network Scheme for Rotorcraft UAV Sensor Failure Diagnosis," in *4th International Symposium on Neural Networks*, Nanjing, China, 2007.

[40] R. Swischuk and D. Allaire, "A Machine Learning Approach to Aircraft Sensor Error Detection and Correction," in *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, Florida, USA, 2018.

[41] L. Garbarino, G. Zazzaro, N. Genito, G. Fasano and D. Accardo, "Neural Network based architecture for Fault Detection and Isolation in air data systems,," in *AIAA 32nd Digital Avionics Systems Conference (DASC)*, 2013, 2013.

[42] S. Hansen and M. Blanke, "Diagnosis of Airspeed Measurement Faults for Unmanned Aerial Vehicles," *IEEE Trans. on Aerospace and Electronic Systems,* vol. 50, no. 1, pp. 224-239, 2014.

[43] F. A. P. Lie, *Synthetic air data estimation: a case study of model-aided estimation*, Minnesota: University of Minnesota, 2014.

[44] C. Lu, R. Li, J. Liu and T. Lei, "Air Data Estimation by Fusing Navigation System and Flight Control System," *Journal of Navigation,* vol. vol. 71, no. no. 5, p. 1231–1246, 2018.

[45] M. Fravolini, G. d. Core, U. Papa, P. Valigi and M. Napolitano, "Data-Driven Schemes for Robust Fault Detection of Air Data System Sensors," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 234-248, 2017.

[46] M. R. Napolitano, Aircraft Dynamics: from Modeling to Simulation, Hoboken, NJ: John Wiley & Sons, 2011.

[47] H. Felemban, J. Che, C. Cao and I. Gregory, "Estimation of Airspeed Using Continuous Polynomial Adaptive Estimator," in *AIAA Guidance, Navigation, and Control Conference*, National Harbor, MD, 2014.

[48] F. Amato, C. Cosentino, M. Mattei and G. Paviglianiti, "A direct/functional redundancy scheme for fault detection and isolation on an aircraft," *Aerospace Science and Technology*, vol. 10, no. 4, pp. 338-345, 2006.

[49] S. Gururajan, M. Rhudy, M. Fravolini and M. Napolitano, "Failure Detection and Accommodation Approaches for the airspeed sensor on a small UAV," in *Fault Detection: Classification, Techniques and Role in Industrial Systems*, Birmingham, Nova, 2013, p. 266.

[50] S. Gururajan, M. Fravolini, M. Rhudy, A. Moschitta and M. Napolitano, "Evaluation of Sensor Failure Detection, Identification and Accommodation (SFDIA) Performance Following Common-Mode Failures of Pitot Tubes," in *SAE 2014 Aerospace Systems and Technology Conference*, Cincinnati, 2014.

[51] F. Balzano, F. Fravolini, M. Napolitano, S. d'Urso, M. Crispoltoni and G. d. Core, "Fault Detection With An Augmented Floating Limiter.," *IJAE - International Journal of Aerospace Engineering*, 2018.

[52] M. Crispoltoni, M. Fravolini, F. Balzano, S. D'Urso and M. Napolitano, "Interval Fuzzy Model for Robust Aircraft IMU Sensors Fault Detection," *Sensors*, vol. 18, no. 8, 2018.

[53] L. G. Zea, *Flight Dynamics Lecture Slides, Topic: Equations of motion for fixed wing aircraft*, Medellin, Antioquia: Universidad Pontificia Bolivariana, 2019.

[54] T. Nakamura and E. Johnson, "Trade Studies on Implementation of Extended Kalman Filters for sUAS Navigation," in *AIAA Scitech 2019 Forum*, San Diego, California, 2019.

[55] S. Oh and E. Johnson, "Development of UAV Navigation System Based on Unscented Kalman Filter," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, 2006.

[56] F. L. L. E. J. B. L. Stevens, Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, Third edition ed., Hoboken, New Jersey: Wiley-Blackwell, 2015.

[57] R. M. Rogers, Applied Mathematics in Integrated Navigation Systems, Reston, VA: AIAA Education Series, 2003.

[58] L. Wenger and D. Gebre-Egziabher, "System Concepts and Performance Analysis of Multi-Sensor Navigation Systems for UAV Applications," in *Aiaa guidance, navigation, and control conference*, Austin, Texas., 2003.

[59] J. Wendel, A. Maier, J. Metzger and G. Trommer, "Comparison of Extended and Sigma-Point Kalman Filters for Tightly Coupled GPS/INS Integration," in *AIAA guidance, navigation, and control conference*, San Fancisco, California., 2006.

[60] C.F. Lin, Modern Navigation, Guidance, and Control Processing, Englewood Cliffs, NJ: Prentice Hall, 1991.

[61] R. G. Brown and P. Hwang, Introduction to Random Signals and Applied Kalman Filtering, New York: John Wiley & Sons, 1997.

[62] B. Anderson and J. Moore, Optimal Filtering, Mineola, NY,: Dover Publishings, 2005.

[63] C. Chui and G. Chen, Kalman Filtering with Real-Time Applications, New York: Springer, 1999.

[64] R. F. Stengel, Optimal Control and Estimation, New York: Dover Publishings, 1994.

[65] NOAA, "The World Magnetic Model," NOAA, 10 12 2019. [Online]. Available: https://www.ngdc.noaa.gov/geomag/WMM/. [Accessed 18 1 2020].

[66] L. G. Zea, *Flight Dynamics Lecture Slides, Topic: Dynamic_behavior_of_a_fixed_wing_aircraft_handout,* Medellin, Antioquia: Universidad Pontificia Bolivariana, 2019.

[67] M. Anguelova, Printed at the Department of Mathematical Sciences, Göteborg, 2007.

[68] P. C. a. H. C. X. Wang, "Local observability of nonlinear systems based on linearization," in *2008 2nd International Symposium on Systems and Control in Aerospace and Astronautics*, Shenzhen, 2008.

[69] S. S. Sastry and C. A. Desoer, "The Robustness of Controllability and Observability of Linear Time Varying Systems," in *IEEE Trans. Automatic Control*, 1982.

[70] J. JANG and C. SUN, "Neuro-Fuzzy Modeling and Control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378 - 406, 1995.

[71] J. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665 - 685, 1993.

[72] J. Jang and N. Gulley, "Gain scheduling based fuzzy controller design," in *NAFIPS/IFIS/NASA '94. Proceedings of the First International Joint Conference of The North American Fuzzy Information Processing Society Biannual Conference. The Industrial Fuzzy Control and Intellige*, San Antonio, TX, USA, 1994.

[73] J. Jang, C. Sun and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Upper Saddle River, NJ: Prentice-Hall Inc, 1997.

[74] Mathworks Inc, *Matlab Fuzzy Logic Toolbox™ User's Guide,* Natick, MA: Mathworks Inc, 2018.

[75] L. Gutierrez, *Adaptive Mode Transition Control Architecture with an Application to Unmanned Aerial Vehicles,* Atlanta: Georgia Institute of Technology, 2004.

[76] R. Stengel, "Robert F. Stengel," 7 1 2020. [Online]. Available: http://www.stengel.mycpanel.princeton.edu/FRL.pdf. [Accessed 5 6 2020].

[77] P. Hamel, Inflight Simulators and Flight-by-Wire / Light Demonstrators, Brunswick, Germany: Springer, 2017.

[78] E. Seckel and J. Morris, "Full Scale Wind Tunnel Tests of a North American Navion Airframe with Positive and Negative Propeller Thrust and Up and Down Flap Deflection," Princeton University AMS, Princeton, 1970.

[79] C. D. Perkins and R. E. Hage, Airplane Performance Stability and Control, New York,: John Wiley and Sons, Inc.,, 1949.

[80] B. Etkin, Dynamics of Flight, New York: John Wiley and Sons, Inc, 1959.

[81] E. Seckel and J. Morris, "The Stability Derivatives of the Navion Aircraft Estimated by Various Methods and Derived from Flight Test Data," Princeton University, Princeton, 1971.

[82] D. Ellison, USAF Stability and Control Handbook (DATCOM), Ohio,: WrightPatterson Air Force Base, 1968.

[83] J. Campbell and M. McKinney, "Summary of Methods for Calculating Dynamic Lateral Stability and Response and for Estimating Lateral Stability Derivatives," NACA, Langley, VA, 1952.