AIRDATA SENSOR BASED POSITION ESTIMATION AND FAULT DIAGNOSIS IN AERIAL REFUELING

by

HAKKI ERHAN SEVIL

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2013

To my parents, Erol and Zehra,

and my sister Yasemin.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Atilla Dogan for the opportunity to conduct my doctoral studies with him and for his endless support during my years at the University of Texas at Arlington (UTA). It has been a rewarding experience to have Dr. Dogan's guidance throughout my doctoral education, and it is my privilege to have Dr. Dogan as my advisor.

I would like to thank the members of my dissertation committee, Dr. Kent Lawrence, Dr. Kamesh Subbarao, Dr. Frank Lu and Dr. Brian Huff for their interest in my research, for their valuable comments and suggestions, and for taking time to serve in my dissertation committee.

I would also like thank the Mechanical and Aerospace Engineering department, which provided me travel support for the academic conferences. Thanks to all of UTA's professors who are genuine and great educators without any doubt, for helping me to find my drive and passion through course work. Also thanks to all the administrative staff for their endless patience and for guiding me when I did not know what to do. I have been lucky to have my colleagues and labmates at the Computer Aided Control System Design Laboratory and the Autonomous Vehicles Laboratory. I have benefited a lot from our discussions, brainstorming and studies.

I am grateful to my professors at Izmir Institute of Technology, especially to my previous advisor Dr. Serhan Ozdemir without whom I would not be able to be here. I would like to thank them all for their support, encouragement and for being extraordinary mentors and excellent instructors. I cannot thank enough my labmates at the Control, Artificial Intelligence & Design Laboratory and the Mechatronics

Laboratory, with whom I spent precious years together. They have been more than friends to me, and I have learned a lot from their vast of knowledge.

I would like to thank my friends in Izmir and Istanbul for their continuous support, encouragement and patience. They always encouraged me to trust my instincts and to pursue my dreams. They constantly express their belief in me. I have been privileged to have them in my life.

I would like to express my deepest gratitude to my parents, and to all members of the Sevil family. Special thanks to my father Erol who is an inspirational role-model for me, to my mother Zehra who supports me endlessly, and to my sister who always encourages me to aim higher. The love of my parents and my sister gave me power to reach the heights I sometimes doubt that I can achieve.

July 22, 2013

ABSTRACT

AIRDATA SENSOR BASED POSITION ESTIMATION AND FAULT DIAGNOSIS IN AERIAL REFUELING

HAKKI ERHAN SEVIL, Ph.D.

The University of Texas at Arlington, 2013

Supervising Professor: Atilla Dogan

Aerial refueling is the process of transferring fuel from one aircraft (the tanker) to another (the receiver) during flight. In aerial refueling operations, the receiver aircraft is exposed to a nonuniform wind field induced by the tanker aircraft, and this nonuniform wind field leads to differences in readings of airdata sensors placed at different locations on the receiver aircraft. There are advantages and disadvantages of this phenomenon. As an advantage, it is used as a mechanism to estimate relative position of the receiver aircraft inside the nonuniform wind field behind the tanker. A model of the nonuniform wind field is used to construct maps of the wind field as functions of relative position in terms of airspeed, side slip angle and angle of attack. Several algorithms are developed that use measurements from three airdata sensors placed at different locations on the receiver aircraft and the inverse of the maps to estimate the relative position of the receiver with respect to the tanker.

The disadvantage of the phenomenon is that the differences in the readings of airdata sensors cause false fault detections in a redundant-sensor-based Fault Detection and Isolation (FDI) system developed based on the assumption of identical

sensor readings from three airdata sensors. Such an FDI algorithm successfully performs detection and isolation of sensor faults when the receiver aircraft flies solo or outside the wake of the tanker aircraft. However, the FDI algorithm yields false fault detection when the receiver aircraft enters the tanker's wake. This problem is resolved by modifying the FDI algorithm. For robustness, the expected values of the sensor measurements are incorporated in the FDI algorithm, instead of the assumption of identical measurements from the sensors. The expected values, which depend on the position of the receiver relative to the tanker, are obtained from the maps of the nonuniform wind field as functions of the relative position. The new robust FDI detects and isolates sensor faults, as well as it eliminates the false fault detection in the nonuniform wind field induced by the tanker aircraft.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Statement

An aircraft flying in air induces a nonuniform wind field around its surroundings through various mechanisms such as trailing vortices emanated from the wings and the tail (Fig. 1.1). In aerial refueling operations, the receiver aircraft needs to fly in this nonuniform wind field. In this wind velocity field, the flow has different magnitude and direction depending on the position relative to wind-generating aircraft.

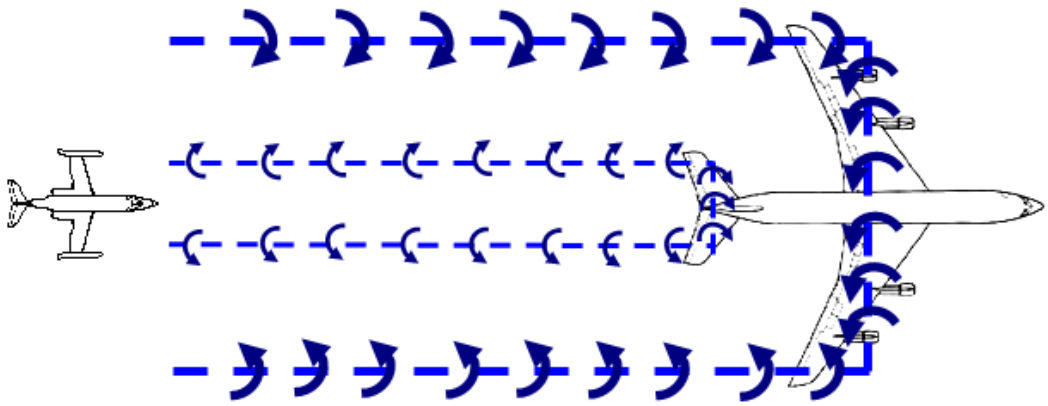


Figure 1.1. The Wing and Horizontal Tail Vortices Induced by Tanker Aircraft [1].

Airdata measurements provide aircraft velocity vector information with respect to the local air. An airdata sensor on an aircraft, which is the receiver aircraft in aerial refueling, flying within the nonuniform wind field will measure different airspeed, side slip angle, and angle of attack values depending on its position relative

to the tanker aircraft. At a fixed position of the receiver aircraft relative to the tanker, airdata sensors placed at different positions on the receiver will also measure different airspeeds, side slip angles and angles of attack. This is because the nonuniform wind field has significant velocity variation over the span of receiver aircraft (Fig. 1.2). This phenomenon can be an advantage or a disadvantage for the receiver flight control system depending on what aspect of the aerial refueling problem is considered. An important advantage is to consider the nonuniform wind field as a signature of the tanker aircraft in the air and thus use it to determine the relative position with respect to the tanker. If a map of the wind field is available in terms of the magnitude and direction as a function of the relative position, the inverse of this map can be used to estimate the relative position given the wind velocity vector measurement. Airdata sensor measurements (airspeed, side slip angle and angle of attack) can be used to determine where the airdata sensor is relative to the tanker aircraft. For this idea to work, there should be a one-to-one mapping from relative position to airdata sensor measurements. This would permit inverse maps to be generated that use airdata measurements to predict the relative position without any ambiguity.

A disadvantage of flying in nonuniform wind field appears for the Fault Detection and Isolation (FDI) system of the receiver aircraft in the form of differences in measurements from multiple airdata sensors placed on different locations on the aircraft. In an FDI system relying on redundant measurements from identical sensors, the sensor readings are compared to each other and after a decision making process, a sensor that has a fault can be identified. However, if the difference in sensor readings is caused by nonuniform wind field instead of a fault in a sensor as in aerial refueling, this leads to a false fault detection. In other words, since the receiver aircraft flies within the nonuniform wind field in an aerial refueling operation, the FDI algorithm (based on the assumption that multiple sensors provide measurements with identi-

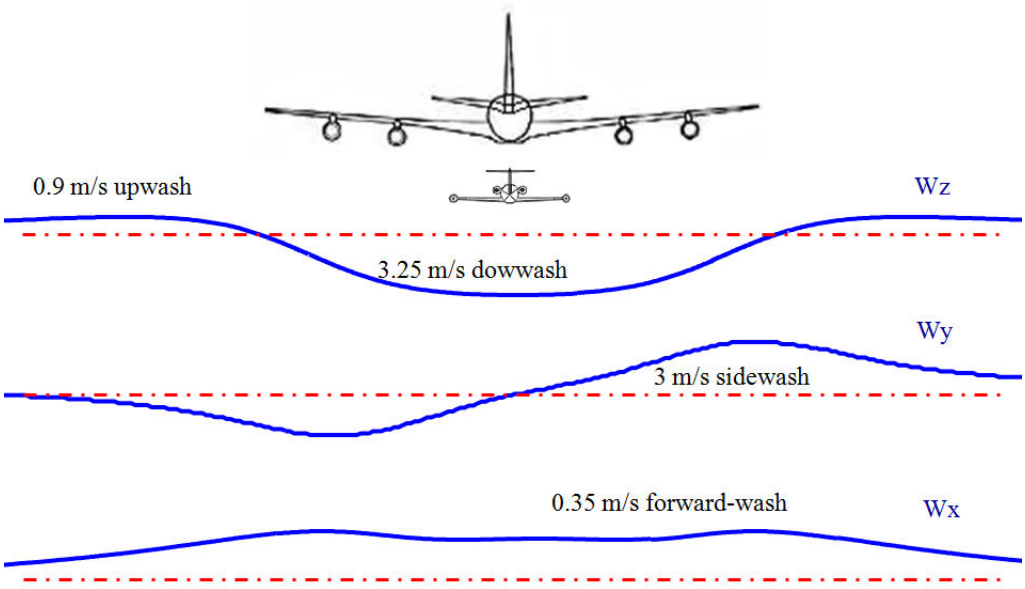


Figure 1.2. Nonuniform Wind Distribution that Receiver Aircraft is Exposed during Aerial Refueling [1].

cal mean in no-fault conditions) tends to experience a high probability of false fault detections. Thus, such an FDI system should be improved to become robust to operate satisfactorily based on multiple airdata sensors placed at different locations on receiver aircraft in an aerial refueling operation as well as in all other flight phases.

This research addresses both the advantage and the disadvantage of flying in the nonuniform wind field for the receiver aircraft in aerial refueling operation

- By utilizing the variation of the induced wind vector field relative to the tanker aircraft, a position estimation algorithm can be developed that calculates the position of the receiver relative to the tanker based on multiple airdata sensor.
- After confirming that the wake of the tanker causes false fault detection in a hardware-redundant FDI system on the receiver aircraft, the maps of the nonuniform wind field is used to provide robustness to the FDI system against the exposure to nonuniform wind field.

1.2 Literature Survey

This section provides a literature review of related topics relevant to the research presented in this dissertation. Specifically, the literature review focuses on relative position estimation techniques and FDI methods. Additionally, the papers related to modeling and control system design for aerial refueling, aerodynamic effects of the tanker over receiver in aerial refueling, and several reports related to accidents associated with the failure of airdata sensors are also presented in various subsections.

1.2.1 Modeling and Control System Design for Aerial Refueling

Recent research has been conducted in the areas of modeling, guidance, and control to support the addition of aerial refueling capabilities to both piloted and unmanned aircarft. Ro and Kamman [2] present a finite-segment approach to modeling dynamic equations for hose-paradrogue aerial refueling systems. Waishek et al. [3] introduce the derivation of the dynamics equations of receiver aircraft in aerial refueling that include time varying mass and inertia caused by the fuel transfer. In that study, the speed and direction of the fuel flow and the location of the receptacle in the receiver are taken into consideration in the dynamic equations of the receiver aircraft. An adaptive flight controller for unmanned combat aerial vehicle and aerial refueling is presented by Wang et al. [4]. Enomoto et al. [5] introduce an automatic chase guidance and control system with a dynamic inversion approach. In Ref. [6], control algorithms for relative motion during aerial refueling in racetrack maneuvers are presented. The controllers designed for both lead and follower aircraft are based on gain-scheduling according to the commanded speed and yaw rate. In that study, linear quadratic regulator (LQR)-based multi-input/multi-output (MIMO) state feedback and integral control techniques are used for both aircraft. The first control system is used to track commanded speed, altitude, and yaw rate for the lead aircraft. The sec-

ond controller is used to track commanded trajectory expressed in the lead aircraft's body frame.

1.2.2 Aerodynamic Coupling in Aerial Refueling

One important topic in aerial refueling is the aerodynamic effects of the tanker aircraft on the receiver aircraft. The tanker aircraft induces a significant nonuniform wind gradient over the receiver aircraft's span, length, and height depending on its position relative to the tanker aircraft. Aerodynamic effects in aerial refueling have been widely investigated and wake vortex models are available in the literature [7, 8, 9, 10]. Han et al. [11] present a study of the wake shape behind the wings in close formation using a discrete vortex method. In that study, unsteady wake vortex evolution is investigated for the wings, and load distributions along the lifting wings are represented with point vortices with finite circulation. Based on the assumption of point vortices that deform freely with a force-free position, trailing wakes from each wing are simulated in close formation flight. A new technique for the vortex effect modeling is presented by Venkataramanan et al. [7]. They use a Helmholtz profile based modified horseshoe vortex model that is incorporated in a six-DOF analysis of a close formation flight. This model includes the effects of geometric parameters and the relative orientation between the lead and follower aircraft, resulting in a vortex-induced-velocity field that can be used directly to obtain the effect on the dynamics of the trailing aircraft. A model for wake-vortex induced wind with turbulence is presented by Dogan et al. [9]. In that study, VEMT (Vortex Effect Modeling Technique) is used to model the dynamic effect of the nonuniform wind induced by the lead aircraft wake vortices on the follower. Also, the turbulence is modeled by either identifying turbulence components from flight data, or by using the Dryden translational and rotational turbulence model. In another study, Dogan et al. [8]

present methods for modeling the three sources of wind, i.e. prevailing wind, wake-vortex-induced wind, and turbulence that the aircraft are exposed to in a formation flight. In that study, a direct computation of the winds experienced by the lead and follower aircraft are obtained from the available flight data. The wind and turbulence modeling techniques are validated by comparing the simulation results with the flight data in terms of the power spectral densities and mean variations.

1.2.3 Relative Position Estimation

In aerial refueling operations, the receiver aircraft's position should be controlled relative to the tanker aircraft. This requires the relative position information to be provided to the pilot or controller of the receiver aircraft. In a piloted aircraft, this is accomplished by the visual senses of the pilot. In order to reduce the pilot workload as well as enable Unmanned Aerial Vehicles (UAVs) with aerial refueling capability, recent efforts focus on automating the aerial refueling operation for receiver aircraft [12, 13]. Automated aerial refueling (AAR) is important especially for UAVs because, with the human pilot board, they can operate for a long time without having to land for refueling [14, 15]. DARPA recently announced the results of test flights from their Autonomous High-Altitude Refueling (AHR) program [16]. Their reported results indicate that 60% of their automated aerial refueling tests achieved successful contact.

In order to develop an automated aerial refueling system, relative position of the receiver aircraft should be accurately and reliably measured. Global Positioning System (GPS) data along with advanced vision-based methods are most commonly considered for this purpose, and they are presented in the literature. These methods can be grouped in two major categories. The first category consists of the methods which are based on camera vision, pixel motion tracking, radar/infrared/laser/lidar signals or optical systems. In this category, modifications are needed to be done only

on trailing aircraft, and the methods in this category do not require any communication between the receiver and the tanker. The second category includes all the methods that require a data-link between lead and follower aircraft, or require that equipment be mounted on both aircraft. The methods in this category require a communication link between the receiver and the tanker, and modifications are needed to be done on both aircraft. There are also hybrid methods using some combination of methods from each category.

One of the methods from the first category is presented in Ref. [17] that uses a visual snake optical sensor. In that system, an algorithm is used to segment the image of the target by having a closed, nonintersecting contour. Then, the controller receives relative position measurements from the optical sensor and associated relative navigation algorithms, and performs a boom-receptacle autonomous air refueling task, within specifications of docking accuracy and maximum docking velocity. Tangale et al. [18] introduce a reference-observer-based tracking controller using a vision-based relative navigation sensor for hose-drogue aerial refueling. In that study, the system on the receiver aircraft generates a smooth reference trajectory for the sensor measurement of relative drogue position, then feedforward control and the reference states to be tracked are estimated by an output injection observer. In another study [19], a vision based navigation (VisNav) system that enables probe-drogue autonomous aerial refueling for unmanned aerial vehicles is introduced. The VisNav consists of a new kind of optical sensor combined with structured active light sources (beacons) to achieve a selective vision, and it provides relative position measurements for an optimal proportional-integral-filter nonzero setpoint controller. A semi-autonomous docking problem for UAVs is investigated by DellAquila et al. [20]. They introduce a real-time machine-vision position sensing system for boom-receptacle aerial refueling operations. Johnson et al. [21] present two Extended Kalman Filter (EKF)

based methods for relative navigation that are named Center Only Relative State Estimation (CORSE) and Subtended Angle Relative State Estimation (SARSE). In these methods, the idea is to use only information from a single camera mounted on the follower aircraft. Image processing techniques are used to output either only the center of the target, or locations of the wingtips of the target in addition to the center of the target. This information is then fed to the EKF in order to estimate both the velocity and position of the aircraft relative to the target. A vision-based sensor and navigation system is introduced for UAVs by Valasek et al. [22], in order to achieve precise and reliable probe-and-drogue autonomous aerial refueling. Recently, two algorithms based on a scanning Light Detection and Ranging (LiDAR) sensor are presented by Curro [23] for determining the position of the receiver aircraft relative to the tanker aircraft. The first algorithm finds the closest points between a tanker model and the LiDAR measurements by comparing the measured values with the model of the aircraft. The second algorithm, on the other hand, tries to match the actual LiDAR measurements with a perturbed position of the tanker. One of the drawbacks of all the systems in this category is that they are highly dependent on environment conditions. For instance, the camera systems can be blinded by the direct sun light. The systems that use pixel-tracking methods or optical devices are not suitable for all weather conditions. Especially, cloudy or foggy weather can cause problems with the performance of these systems. Additionally, if the view or objective area of the sensors is blocked, the systems in this category cannot perform at all.

Various other approaches and methods have been developed that fall into the second category. A method of relative position algorithm is presented by Jiang and Chen [24] that is based on the error model of an internal navigation system (INS). They use the combination of a least squared estimation and the Taylor series method

to obtain relative position of multiple aircraft. Similarly, a vision algorithm that includes optimal marker labeling is presented by Pollini et al. [25] for probe-and-drogue refueling. This approach allows estimation of the relative displacement via using localization of infrared light emitting diode (LED) markers placed on the body to be tracked. Mahboubi et al. [26] introduce a method that is based on the same principal. In their study, the follower aircraft uses a camera to capture the images of the high-intensity LEDs that are mounted on the lead aircraft. Then, relative position and orientation of the follower aircraft with respect to the lead aircraft is determined by using a computer vision algorithm. Smith [27] introduces a proportional navigation method with adaptive terminal guidance for midair constant-altitude aircraft rendezvous. In that study, the proposed navigation system coupled with a velocity control system uses airspeed, ground speed, velocity, heading, and position of the lead aircraft in its equations. Moreover, a high-accuracy and redundant navigation system is applied to boom-receptacle aerial refueling to estimate position, velocity, and attitude of aircraft is presented by Williamson et al. [28]. In that study, an Extended Kalman Filter based sensor fusion technique is introduced, which is composed of electro-optic (EO) image sensors, a wireless communication system, and a differential carrier-phase (DCP) GPS receiver. Navigation of autonomous airborne refueling is analyzed and optimal navigation architecture is defined by Khanafseh and Pervan [29]. They developed a high-fidelity dynamic sky-blockage model for GPS blockage problem by the tanker aircraft during boom-receptacle type aerial refueling. The methods that are classified in this category have an important and common drawback. These systems are not self contained systems. Any disconnection of the datalink required by these systems will make them nonfunctional. Another drawback of these systems is that they are subject to jamming which affects the performance of the system.

In Ref. [30], a hybrid system that includes an Extended Kalman Filter based relative position estimation method is presented. In this approach, the states of the lead aircraft are assumed to be known and the relative states are estimated using (i) line-of-sight measurements between the vehicles, and (ii) acceleration and angular rate measurements of the follower. Visual navigation beacons are placed on the follower for the line-of-sight measurements. Campa et al. [31] present a combination of GPS and Machine Vision (MV) guidance for automated boom-receptacle aerial refueling in which a sensor fusion between MV based measurements and GPS signals is proposed. Naturally, this system along with all the systems that can be titled as hybrid systems have drawbacks from the first or second categories. Hybrid systems are not self contained and are not suitable for all weather use.

1.2.4 Fault Detection and Isolation

FDI algorithms are used to identify any occurrence of a fault in a system (detection) and to pinpoint the type and location of the fault (isolation). Analytical and hardware redundancy are the two main FDI methods. The methods of analytical redundancy are based on the comparison of estimated states against the measurements. Anwar and Chen present an observer-based analytical FDI algorithm that uses a full-state observer for vehicle-body sideslip angle [32]. A bank of residual generators and a discrete-event system-based actuator fault diagnoser is presented by Meskin et al. for a network of multiple quad-rotors [33]. In another study, Tousi et al. introduce an observer-based fault detection method for a team of unmanned aerial vehicles [34]. Unknown input observers are also used for fault detection of aircraft sensor and actuator faults. This method is implemented on a large cargo jet aircraft model [35]. Hajiyev presents a sensor fault detection method based on testing the innovation covariance of the Kalman filter using Tracy-Widom Distribution [36]. A model-based

approach to fault detection for an air data system is proposed by Freeman et al., who use H_∞ synthesis of a robust fault detection filter [37].

The methods of hardware redundancy are based on the comparison of measurements from multiple sensors of the same kind. Shaoping et al. present a redundant-sensor-based fault reasoning [38] that uses a logic algebra reasoning technique based on sequential variables for the fault diagnosis of magnetic bearing systems. A fault detection algorithm using the double exponential smoothing method that is applied to a brake pedal signal is given by Kim et al. [39]. Dajani-Brown et al. present a sensor voter algorithm in order to manage three redundant sensors [40]. Another voter-based fault detection system is presented by Kerr that is implemented on multiple simultaneous sensor subsystems such as GPS, INS, and Doppler AHRS [41]. An FDI method using singular value decomposition (SVD) is presented by Shim and Yang for INS sensor redundancy [42]. Berdjag et al. present an application of FDI on redundant aircraft sensors using Fuzzy logic approach [43]. Eubank et al. implement a fault detection algorithm for air-data systems of an unmanned autonomous seaplane [44].

There are also hybrid FDI systems that use a combination of hardware and analytical redundancy. Kim et al. present a FDI algorithm for an UAV inertial sensor that is based on hardware redundancy combined with an analytic redundancy by utilizing the unscented Kalman Filter [45]. In another study, Kim et al. propose another hybrid algorithm that is based on an Extended Kalman Filter [46]. A statistical overlap test based approach is presented by Brumback and Srinath that is applied to a dual-inertial/Doppler radar navigation system [47]. Implementations of hybrid FDI on air data systems are also presented in the literature. An approach that is based on a bank of extended H_∞ observers is proposed by Mattei and Paviglianiti for a redundant airdata system for a small commercial aircraft [48]. Wheeler et al.

applied a probabilistic performance metrics based fault detection scheme to air-data sensors [49].

One of the well known and widely used approach in both analytical and hardware redundant systems is the parity space approach with roots going back to early 1980s. In parity space based fault detection, the idea is to create residuals that represent the difference between the actual system and the model outputs [50]. Gertler and Monajemy present a method for generating directional residuals using dynamic parity relations [51], and Gertler presents the design of dynamic parity relations for the detection and isolation of both additive and multiplicative faults [52]. Generalization of parity space for FDI to the continuous-time case is presented by Medvedev [53]. Kratz et al. show a residual generator using parity relations for time-delay systems [54]. A Kalman filter based FDI algorithm that provides estimates of redundant strapdown inertial measurement units (IMUs), that affect the parity vector is proposed by Hali et al. [55]. Chow and Willsky also present a design for residual generation using analytical redundancy and parity space [56]. A fault detection algorithm based on the parity approach and chi-square hypothesis testing is presented by Sturza for redundant strapdown inertial navigation systems, and GPS navigation sets [57]. Oliveira et al. present a FDI system based on χ^2-CUSUM that uses parity space projection and Wavelet Packets that is applied to redundant IMUs [58]. A comparative review of three FDI approaches (the parity space method, the Mahalanobis distance method and its direct robustification) is presented by Guerrier et al. for multiple MEMS-IMU configurations [59].

1.2.5 Flight Accidents attributed to the Failure of Airdata Sensors

In this study, a FDI system is developed for multiple airdata sensors. Several accidents associated with the failure of airdata systems are discuessed below to demonstrate the importance of reliable airdata measurements.

On September 29, 2004, an F/A-22 suffered a Class A mishap near Edwards Air Force Base [60]. As the F/A-22 passed behind an F-16 during an air-to-air tracking flight, wake vortices from the F-16 caused extremely large perturbations in the airdata sensor measurements on the F/A-22. These caused a divergent oscillation in the pitch axis of the F/A-22, which resulted in the aircraft exceeding both angle of attack and structural limits, even though there were limiters in the flight control system. On September 17, 2007, an F-16 was lost in Italy when icing conditions froze the air data probe, resulting in inaccurate angle of attack and airspeed measurements [60]. When the aircraft attempted to climb out of the icing conditions, the airspeed reduced and angle of attack increased. The faulty sensor prevented the flight control system from initiating inputs to reduce the angle of attack. The aircraft eventually departed from controlled flight and crashed. On February 23, 2008, a B-2 Spirit crashed after takeoff from Andersen Air Force Base, Guam. According to the accident report [61], the aircraft's flight control computers calculated an inaccurate airspeed, low altitude/low airspeed, and a negative angle of attack, which led to an uncommanded 30 degree nose pitch-up on takeoff, causing the aircraft to stall. The reason for the inaccurate airspeed calculation was moisture in the aircraft's Port Transducer Units during air data calibration. On October 7, 2008, an Airbus A330-303 aircraft had an in-flight upset at 154 km west of Learmonth, WA, while it was flying from Singapore to Perth, Australia with 303 passengers, nine cabin crew and three flight crew onboard. During the flight, the autopilot disconnected, followed by various aircraft system failure indications. After that, the aircraft suddenly pitched nose-down. Approximately 3

minutes later, the aircraft started a second uncommanded pitch-down maneuver. In both events, the aircraft reached maximum pitch angles, 8.4 degrees nose-down and 3.5 degrees nose-down, respectively. According to the investigation report [62], two safety factors were identified associated with pitch-down movement. One of them is the airdata inertial reference units (ADIRUs), which provided erroneous data (spikes) on many parameters to other aircraft systems. The cabin crew declared MAYDAY and the aircraft landed at Learmonth. Eleven passengers and 1 flight attendant were injured during this incident. On June 1, 2009, Air France Flight 447 crashed into the Atlantic Ocean, killing all passengers and crew [63]. A preliminary analysis of the recently recovered flight data recorder suggests that a problem with the plane's pitot tubes giving inconsistent speed readings might have contributed to the crash [63].

1.3 Organization of the Dissertation

The dissertation is organized as follows. Chapter 2 describes the modeling of both tanker and receiver aircraft dynamics, their controller design, and the modeling of the wind sources and their dynamic effect on receiver aircraft. The equations of motion of receiver and tanker aircraft are given in Chapter 2, as well as equations for prevailing wind, turbulence, and wake vortex induced wind. The effect of the nonuniform wind field on the airdata sensor measurements is discussed in Chapter 3. Also, a general information about the airdata sensors and positions of receiver aircraft during aerial refueling operation is presented in Chapter 3. Definitions of the relative estimation algorithms developed in this research and performance metrics defined for evaluating the algorithms are presented in Chapter 4. Chapter 5 describes the FDI algorithm developed using a statistical approach. The robust FDI algorithm is also presented in Chapter 5, as well as the equations used for the measurement correction due to the rigid body motion of the aircraft. Metrics for evaluating the FDI algorithm

are also prensented. Chapter 6 contains the simulation results and discussions of the performance of the both the position estimation and FDI algorithms. Conclusions and suggestions for future work are presented in Chapter 7.

1.4 Original Contributions

The original contributions of this dissertation can be separated in two parts. For the relative position estimation component of this work, airdata sensor measurements (airspeed, side slip angle, and angle-of-attack) of a trailing aircraft are used to estimate its relative position with respect to the lead aircraft which generates bound and tip vortices. The following contributions have been made in this part:

- Reference airspeed, side slip angle and angle of attack measurements (or calculations) from three airdata sensors placed at different locations on the aircraft for various relative positions with respect to the lead aircraft are organized as maps. Absence or the number of intersection of the contour lines of the maps determines whether the position estimation is possible with the minimum number of sensors.
- Various logical rule based algorithms are developed that estimate the relative position of the trailing aircraft with respect to the lead aircraft using the measurements of airspeed, side slip angle and angle of attack by single or multiple sensors. This is done by determining the intersections of the contour lines of the preformed maps of the reference airspeed, side slip angle and angle of attack. The contour lines are selected based on the measurements of the respective airdata variable, airspeed, side slip angle and/or angle of attack.
- A procedure is developed that quantify the efficiency of a given algorithm in terms of its accuracy in the relative position estimation and its ability to produce a relative position estimation. This procedure is also used to quantify the

robustness of a given algorithm against the level of turbulence and the measurement noise.

In the fault detection and isolation section of this research, a redundant sensor based FDI system is developed and applied to the FDI of multiple airdata sensors on a receiver aircraft in aerial refueling. The FDI system uses the parity space approach along with the chi-square test hypothesis for detection and isolation purposes. The following contributions have been made:

- The effect of a nonuniform wind distribution that is induced by the tanker aircraft over receiver aircraft on such an FDI system is demonstrated through simulation.
- The FDI algorithm is improved to eliminate false fault detections caused by the assumption of identical sensor readings in the nonuniform wind field. The new robust FDI algorithm developed in this research accounts for the variations of the airdata variables in a nonuniform wind field during aerial refueling.
- The expected values of airdata measurements are obtained from a model of the nonuniform wind field and organized as maps of the airspeed, side slip angle, and angle of attack as functions of the relative position.
- The expected values of the sensor measurements are incorporated in the FDI algorithm, and the FDI system is improved against the measurement variation among the multiple sensors due to the nonuniform wind field.

CHAPTER 2

MODELING OF AIRCRAFT DYNAMICS AND WIND, AND CONTROLLLER DESIGN

This chapter presents full 6-DOF nonlinear equations of motions including the wind effects for receiver and tanker aircraft, their controllers as well as the aerodynamic coupling, prevailing wind and turbulence equations used for simulating Learjet 25 aircraft flying behind a KC-135R tanker aircraft in an aerial refueling flight. These equations were developed in previous research efforts [1, 6, 8, 9, 64, 65, 66] and they are repeated here. Three different frames are used in describing the equations; inertial frame, body-fix frame and wind-frame.

2.1 Equations of Motion of the Tanker Aircraft

The equations of motion for a tanker aircraft are used to model a KC-135R aircraft flying in an aerial refueling operation. The translational kinematics equation of the tanker aircraft is written in terms of the position vector of the tanker with respect to an inertial frame. The translational kinematics equation in matrix form is

$$\dot{r}_{B_T} = \mathbf{R}^{\mathbf{T}}_{\mathbf{B_T I}} \mathbf{R}_{\mathbf{B_T W_T}} V_{w_T} + W_{I_T} \tag{2.1}$$

where $\dot{r}_{B_T}$ is inertial-frame representation of the velocity of the tanker aircraft relative to the inertial frame, $\mathbf{R}_{\mathbf{B_T I}}$ is the rotation matrix from the inertial frame to the tanker's body frame, $\mathbf{R}_{\mathbf{B_T W_T}}$ is rotation matrix from tanker's wind frame to tanker's body frame, V_{w_T} is wind-frame representation of the velocity of the tanker relative to

the surrounding air, and W_{I_T} is the representation of wind in the inertial frame. The integration of the velocity is the position vector.

The rotational kinematics equation in matrix form is

$$\mathbf{R_{B_TI}}\dot{\mathbf{R}}_{\mathbf{B_TI}} = -\mathbf{S}(\omega_{\mathbf{B_T}}) \tag{2.2}$$

where $\omega_{\mathbf{B_T}}$ is the body-frame representation of the angular velocity vector of the tanker relative to the inertial frame and is written in terms of the components as

$$\omega_{\mathbf{B_T}} = \begin{bmatrix} p_T \\ q_T \\ r_T \end{bmatrix} \tag{2.3}$$

The rotational motion of the tanker aircraft is represented in terms of following Euler angles

$$\begin{aligned} \dot{\psi}_{_T} &= (q_{_T} \sin\phi_{_T} + r_{_T} \cos\phi_{_T}) \sec\theta_{_T} \\ \dot{\theta}_{_T} &= (q_{_T} \cos\phi_{_T} - r_{_T} \sin\phi_{_T}) \\ \dot{\phi}_{_T} &= p_{_T} + (q_{_T} \sin\phi_{_T} + r_{_T} \cos\phi_{_T}) \tan\theta_{_T} \end{aligned} \tag{2.4}$$

where $(\psi_{_T}, \theta_{_T}, \phi_{_T})$ are the Euler angles relative to the inertial frame. The translational dynamics are

$$\begin{aligned} \begin{bmatrix} \dot{V}_{_T} \\ \dot{\beta}_{_T} \\ \dot{\alpha}_{_T} \end{bmatrix} &= \varepsilon_{\mathbf{T}}^{-\mathbf{1}} \mathbf{S}(\omega_{B_T})(\mathbf{R_{B_TW_T}} V_{w_T}) - \varepsilon_{\mathbf{T}}^{-\mathbf{1}} \mathbf{R_{B_TI}} \dot{W}_{I_T} \\ &+ \frac{1}{m_{_T}} \varepsilon_{\mathbf{T}}^{-\mathbf{1}} (\mathbf{R_{B_TI}} M_T + \mathbf{R_{B_TW_T}} A_T + P_T) \end{aligned} \tag{2.5}$$

where the expressions for $\varepsilon_{\mathbf{T}}^{-\mathbf{1}}$ and V_{w_T} are expressed as

$$\varepsilon_{\mathbf{T}}^{-\mathbf{1}} = \begin{bmatrix} \cos\alpha_T \cos\beta_T & \sin\beta_T & \cos\beta_T \sin\alpha_T \\ -\frac{1}{V_T}\cos\alpha_T \sin\beta_T & \frac{1}{V_T}\cos\beta_T & -\frac{1}{V_T}\sin\alpha_T \sin\beta_T \\ -\frac{1}{V_T}\sec\beta_T \sin\alpha_T & 0 & \frac{1}{V_T}\cos\alpha_T \sec\beta_T \end{bmatrix}, \; V_{w_T} = \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} \tag{2.6}$$

where (V_T, β_T, α_T) are airspeed, side slip angle, and angle of attack, respectively. There are also external forces acting on the tanker, the gravitational force (M_T), the aerodynamic force (A_T), and the propulsive force (P_T). It should be noted that the gravitational force is expressed in the inertial frame, the aerodynamic force is expressed in the wind frame, and the propulsive force is expressed in the tanker's body frame. These forces are represented as

$$M_T = \begin{bmatrix} 0 \\ 0 \\ m_T g \end{bmatrix} A_T = \begin{bmatrix} -D_T \\ -S_T \\ -L_T \end{bmatrix} P_T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} T_T \cos\delta_T \\ 0 \\ -T_T \sin\delta_T \end{bmatrix} \tag{2.7}$$

where m_T is the mass of the tanker aircraft, g is the gravitational acceleration, (D_T, S_T, L_T) are the drag, side force, and lift on the tanker, T_T is the thrust magnitude, and δ_T is the thrust inclination angle. In Eq. (2.5), $\mathbf{S}(\omega_{B_T})$ is the skew-symmetric matrix, and is defined as

$$\mathbf{S}(\omega_{B_T}) = \begin{bmatrix} 0 & r_T & -q_T \\ -r_T & 0 & p_T \\ q_T & -p_T & 0 \end{bmatrix} \tag{2.8}$$

The aerodynamic forces are expressed in the standard form as

$$D_T = \frac{1}{2}\rho V_T^2 S_T C_{D_T} \tag{2.9}$$

$$S_T = \frac{1}{2}\rho V_T^2 S_T C_{S_T} \tag{2.10}$$

$$L_T = \frac{1}{2}\rho V_T^2 S_T C_{L_T} \tag{2.11}$$

where S_T is the reference area of the aircraft and ρ is the air density. The aerodynamic coefficients are

$$C_{D_T} = C_{D_0} + C_{D_{\alpha^2}}\alpha_T^2 \tag{2.12a}$$

$$C_{S_T} = C_{S_0} + C_{S_\beta}\beta_T + C_{S_{\delta_r}}\delta_{r_T} \tag{2.12b}$$

$$C_{L_T} = C_{L_0} + C_{L_\alpha}\alpha_T + C_{L_{\alpha^2}}(\alpha_T - \alpha_{ref})^2 + C_{L_q}\frac{c_T}{2V_T}q_{rel} + C_{L_{\delta_e}}\delta_{eT} \tag{2.12c}$$

where δ_{r_T} and δ_{eT} are the control surface deflections, rudder and elevator, respectively. c_T is chord length for the tanker, and q_{rel} is y-component of the angular velocity of the aircraft relative to the surrounding air, which is different from the angular velocity relative to the inertial frame in the presence of angular wind. In the case of nonuniform wind exposure, aircraft effectively experiences rotational wind along with the translational wind. Considering the vector relation among the angular velocity of aircraft, angular velocity of air and relative angular velocity of aircraft with respect to the air, the components are similarly related as

$$\begin{aligned} p_{rel} &= p_T - p_{eff} \\ q_{rel} &= q_T - q_{eff} \\ r_{rel} &= r_T - r_{eff} \end{aligned} \tag{2.13}$$

where (p_{rel}, q_{rel}, r_{rel}) are the angular velocity components relative to the surrounding air, ($p_{_T}$, $q_{_T}$, $r_{_T}$) are the angular velocity components relative to the inertial frame, ($p_{_{eff}}, q_{_{eff}}, r_{_{eff}}$) are the rotational wind components relative to the inertial frame. The rotational dynamic equation of the tanker aircraft is

$$\dot{\omega}_{_{B_T}} = \underline{\underline{\mathrm{I}}}_{\mathbf{T}}^{-\mathbf{1}} \left[M_{B_T} + \mathbf{S}(\omega_{_{B_T}}) \underline{\underline{\mathrm{I}}}_{T} \omega_{_{B_T}} \right] \tag{2.14}$$

where $\underline{\underline{\mathrm{I}}}_{\mathbf{T}}$ is the moment of inertia matrix of the tanker aircraft, and M_{B_T} is the applied moment on the tanker aircraft expressed in its body frame. The rolling, pitching and yawing moments ($\mathcal{L}_T$, $\mathcal{M}_T$, $\mathcal{N}_T$) are given as

$$\begin{aligned} \mathcal{L}_T &= \frac{1}{2}\rho V_{_T}^2 S_{_a} b_{_T} C_{\mathcal{L}_T} \\ \mathcal{M}_T &= \frac{1}{2}\rho V_{_T}^2 S_{_a} c_{_T} C_{\mathcal{M}_T} + \triangle_{z_T} T_{_T} \\ \mathcal{N}_T &= \frac{1}{2}\rho V_{_T}^2 S_{_a} b_{_T} C_{\mathcal{N}_T} \end{aligned} \tag{2.15}$$

where $b_{_T}$ is the wingspan of the tanker aircraft, $\triangle_{z_T}$ is the moment arms of the thrust in the tanker's body frame. The moment coefficients are expressed as

$$\begin{aligned} C_{\mathcal{L}_T} &= C_{\mathcal{L}_0} + C_{\mathcal{L}_{\delta_a}} \delta_{a_T} + C_{\mathcal{L}_{\delta_r}} \delta_{r_T} + C_{\mathcal{L}_\beta} \beta_{_T} + C_{\mathcal{L}_p} \frac{b_{_T}}{2V_{_T}} p_{rel} + C_{\mathcal{L}_r} \frac{b_{_T}}{2V_{_T}} r_{rel} \\ C_{\mathcal{M}_T} &= C_{\mathcal{M}_\alpha} \alpha_{_T} + C_{\mathcal{M}_{\delta_e}} \delta_{e_T} + C_{\mathcal{M}_q} \frac{c_{_T}}{2V_{_T}} q_{rel} \\ C_{\mathcal{N}_T} &= C_{\mathcal{N}_0} + C_{\mathcal{N}_{\delta_a}} \delta_{a_T} + C_{\mathcal{N}_{\delta_r}} \delta_{r_T} + C_{\mathcal{N}_\beta} \beta_{_T} + C_{\mathcal{N}_p} \frac{b_{_T}}{2V_{_T}} p_{rel} + C_{\mathcal{N}_r} \frac{b_{_T}}{2V_{_T}} r_{rel} \end{aligned} \tag{2.16}$$

where δ_{a_T} is the aileron deflection.

2.2 Equations of Motion of the Receiver Aircraft

In this dissertation, the equations of motion of the receiver aircraft are used to model a Learjet 25 aircraft. In an aerial refueling operation, the receiver needs to be controlled with respect to the tanker's position and orientation. Thus, the equations of motion are written in terms of the position and orientation of the receiver relative to the body frame of the tanker. The position vector of the receiver aircraft relative to the inertial frame is the sum of the position vector of the tanker relative to the inertial frame and the relative position vector of the receiver with respect to the tanker. This vector relation is written in terms of the vector representations as

$$[\hat{\underline{\mathrm{I}}}]^T r_R = [\hat{\underline{\mathrm{I}}}]^T r_T + [\hat{\underline{\mathrm{I}}}]^T \mathbf{R}^{\mathbf{T}}_{\mathbf{B_T I}} \xi \tag{2.17}$$

where r_R is inertial-frame representation of the position of the receiver relative to the inertial frame, r_T is the inertial-frame representation of the position of the tanker relative to the inertial frame, and ξ is tanker body frame representation of the receiver position relative to the tanker. Considering this relation, the relative translational kinematics is given in matrix form as

$$\dot{\xi} = \mathbf{R}^{\mathbf{T}}_{\mathbf{B_R B_T}} \mathbf{R}_{\mathbf{B_R W_R}} V_w + \mathbf{R}^{\mathbf{T}}_{\mathbf{B_R B_T}} W_{B_R} - \mathbf{R}_{\mathbf{B_T I}} \dot{r}_T + \mathbf{S}(\omega_{\mathbf{B_T}}) \xi \tag{2.18}$$

where ξ is

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} \tag{2.19}$$

Rotational kinematics of the receiver aircraft is given as

$$\begin{aligned}\dot{\psi} &= (q\sin\phi + r\cos\phi)\sec\theta \\ \dot{\theta} &= (q\cos\phi - r\sin\phi) \\ \dot{\phi} &= p + (q\sin\phi + r\cos\phi)\tan\theta\end{aligned} \tag{2.20}$$

where (ψ, θ, ϕ) are the orientation, and *(p,q,r)* are the angular velocities. They both are represented in the receiver body frame relative to the tanker.

Translational dynamics in matrix form are given as

$$\begin{aligned}\begin{bmatrix}\dot{V} \\ \dot{\beta} \\ \dot{\alpha}\end{bmatrix} &= \varepsilon_{\mathbf{R}}^{-\mathbf{1}}\left[\mathbf{S}(\omega_{B_R B_T}) + \mathbf{R}_{\mathbf{B_R B_T}}\mathbf{S}(\omega_{B_T})\mathbf{R}_{\mathbf{B_R B_T}}^{\mathbf{T}}\right](\mathbf{R}_{\mathbf{B_R W_R}}V_w + W_{B_R}) \\ &- \varepsilon_{\mathbf{R}}^{-\mathbf{1}}\dot{W}_{B_R} + \frac{1}{m_R}\varepsilon_{\mathbf{R}}^{-\mathbf{1}}(\mathbf{R}_{\mathbf{B_R B_T}}\mathbf{R}_{\mathbf{B_T I}}M_R + \mathbf{R}_{\mathbf{B_R W_R}}A_R + P_R)\end{aligned} \tag{2.21}$$

where

$$\varepsilon_{\mathbf{R}}^{-\mathbf{1}} = \begin{bmatrix}\cos\alpha\cos\beta & \sin\beta & \cos\beta\sin\alpha \\ -\frac{1}{V}\cos\alpha\sin\beta & \frac{1}{V}\cos\beta & -\frac{1}{V}\sin\alpha\sin\beta \\ -\frac{1}{V}\sec\beta\sin\alpha & 0 & \frac{1}{V}\cos\alpha\sec\beta\end{bmatrix} \tag{2.22}$$

where (V, β α) are airspeed, side slip angle, and angle of attack, respectively. Also, V_w is the velocity relative to the wind frame, $\omega_{B_R B_T}$ is the relative angular velocity, m_R is the mass of the receiver aircraft, W_{B_R} represents the wind expressed in the receiver body frame, and $\dot{W}_{B_R}$ represents the wind derivative relative to the receiver body frame expressed in the receiver body frame, $\mathbf{R}_{\mathbf{B_R B_T}}$ is the rotation matrix from the tanker's body frame to the receiver's body frame, $\mathbf{R}_{\mathbf{B_R W_R}}$ is the rotation matrix from the receiver's wind frame to the receiver's body frame.

There are external forces acting on the receiver, gravitational force (M_R), aerodynamic force (A_R), and propulsive force (P_R). It should be noted that gravitational force is expressed in inertial frame, aerodynamic force is expressed in wind frame, and propulsive force is expressed in receiver aircraft's body frame. These forces are represented as

$$M_R = \begin{bmatrix} 0 \\ 0 \\ m_{_R} g \end{bmatrix} A_R = \begin{bmatrix} -D \\ -S \\ -L \end{bmatrix} P_R = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{2.23}$$

where $m_{_R}$ is the mass of the receiver, g is the gravitational acceleration, (D, S, L) are drag, side force, and lift on the receiver, (T_x, T_y, T_z) are the thrust in x-y-z directions of the receiver body frame.

The aerodynamic forces are expressed in standard form as

$$D = \frac{1}{2}\rho V^2 S_{_R} C_D \tag{2.24a}$$

$$S = \frac{1}{2}\rho V^2 S_{_R} C_S \tag{2.24b}$$

$$L = \frac{1}{2}\rho V^2 S_{_R} C_L \tag{2.24c}$$

where $S_{_R}$ is the reference area of the aircraft and ρ is the air density. The coefficients of aerodynamic force are given as

$$C_D = C_{D_0} + C_{D_\alpha}\alpha + C_{D_{\alpha 2}}\alpha^2 + C_{D_{\delta_e}}\delta_e + C_{D_{\delta_e 2}}\delta_e^2 + C_{D_{\delta_s}}\delta_s + C_{D_{\delta_s 2}}\delta_s^2 \tag{2.25a}$$

$$C_S = C_{S_0} + C_{S_\beta}\beta + C_{S_{\delta_a}}\delta_a + C_{S_{\delta_r}}\delta_r \tag{2.25b}$$

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_\alpha 2}(\alpha - \alpha_{ref})^2 + C_{L_q}\frac{c}{2V_{_R}}q_{rel} + C_{L_{\delta_e}}\delta_e + C_{L_{\delta_s}}\delta_s \tag{2.25c}$$

where δ_r, δ_e and δ_a, and δ_s are the deflections for rudder, elevator, aileron, and stabilizer respectively. q_{rel} in C_L expression is y-component of the angular velocity of the

aircraft relative to the surrounding air in the body frame. With all the components, the representation of the angular velocity of the receiver relative to the surrounding air in the body frame is

$$\omega_{rel} = \begin{bmatrix} p_{rel} \\ q_{rel} \\ r_{rel} \end{bmatrix} \tag{2.26}$$

The angular velocity of the receiver aircraft relative to the air expressed in the body frame can be represented using angular velocity of the receiver relative to the tanker, angular velocity of the receiver relative to the inertial frame and rotational wind, as

$$\omega_{rel} = \omega_{_{B_R B_T}} + \mathbf{R_{B_R B_T}} \omega_{_{B_T}} - \begin{bmatrix} p_{_{eff}} \\ q_{_{eff}} \\ r_{_{eff}} \end{bmatrix} \tag{2.27}$$

where rotation matrix $\mathbf{R_{B_R B_T}}$ is used due to the fact that $\omega_{_{B_T}}$ is the tankers angular velocity in the tankers body frame, and $[p_{_{eff}}\ q_{_{eff}}\ r_{_{eff}}]^T$ are the effective rotational wind components relative to the inertial frame expressed in the body frame of the receiver.

The matrix form of the rotational dynamics in terms of the angular velocity and orientation of the receiver relative to the tankers body frame is given as

$$\begin{aligned} \dot{\omega}_{_{B_R B_T}} &= \underline{\underline{\mathrm{I}}}_R^{-1} M_{B_R} + \underline{\underline{\mathrm{I}}}_R^{-1} \mathbf{S}(\omega_{_{B_R B_T}} + \mathbf{R_{B_R B_T}} \omega_{_{B_T}}) \underline{\underline{\mathrm{I}}}_R (\omega_{_{B_R B_T}} + \mathbf{R_{B_R B_T}} \omega_{_{B_T}}) \\ &- \mathbf{S}(\omega_{_{B_R B_T}}) \mathbf{R_{B_R B_T}} \omega_{_{B_T}} - \mathbf{R_{B_R B_T}} \dot{\omega}_{_{B_T}} \end{aligned} \tag{2.28}$$

where $\underline{\underline{I}}_R$ is the inertia matrix of the receiver aircraft, and M_{B_R} is applied moment on the receiver aircraft in its body frame. The moment of the receiver aircraft at the receiver body frame can be written as

$$M_{B_R} = \begin{bmatrix} \mathcal{L} \\ \mathcal{M} \\ \mathcal{N} \end{bmatrix} \tag{2.29}$$

where ($\mathcal{L}$, $\mathcal{M}$, $\mathcal{N}$) are the rolling, pitching and yawing moments, respectively. The expression of the moments are expressed as

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}\rho V^2 S_{_R} b C_{\mathcal{L}} - \triangle_z T_y + \triangle_y T_z \\ \mathcal{M} &= \frac{1}{2}\rho V^2 S_{_R} c C_{\mathcal{M}} + \triangle_z T_x + \triangle_x T_z \\ \mathcal{N} &= \frac{1}{2}\rho V^2 S_{_R} b C_{\mathcal{N}} - \triangle_y T_x - \triangle_x T_y \end{aligned} \tag{2.30}$$

where b is the wingspan, c is the chord length of the receiver aircraft, ($\triangle_x$, $\triangle_y$, $\triangle_z$) are the moment arms of the thrust in the body frame of the receiver. The moment coefficients are expressed as

$$\begin{aligned} C_{\mathcal{L}} &= C_{L_0} + C_{L_{\delta_a}}\delta_a + C_{L_{\delta_r}}\delta_r + C_{L_\beta}\beta + C_{L_p}\frac{b}{2V}p_{rel} + C_{L_{r_{rel}}}\frac{b}{2V}r_{rel} \\ C_{\mathcal{M}} &= C_{M_0} + C_{M_\alpha}\alpha + C_{M_{\delta_e}}\delta_e + C_{M_q}\frac{c}{2V}q_{rel} + C_{M_{\delta_s}}\delta_s \\ C_{\mathcal{N}} &= C_{N_0} + C_{N_{\delta_a}}\delta_a + C_{N_{\delta_r}}\delta_r + C_{N_\beta}\beta + C_{N_p}\frac{b}{2V}p_{rel} + C_{N_{r_{rel}}}\frac{b}{2V}r_{rel} \end{aligned} \tag{2.31}$$

2.3 Modeling of Wind Sources

The tanker and receiver aircraft are both subject to prevailing wind and atmospheric turbulence during the aerial refueling operations. The receiver aircraft is also subject to nonuniform wind field induced by the wake of the tanker aircraft.

The magnitude and direction of the nonuniform wind field over the receiver aircraft vary with respect to its position relative to tanker aircraft. The total wind that receiver aircraft is exposed to is the superposition of the prevailing wind, turbulence and vortex induced wind. Similarly, the total wind that is experienced by the tanker aircraft is superposition of prevailing wind and turbulence. In this section, modeling techniques for each wind components are presented.

2.3.1 Modeling of Prevailing Wind

The prevailing wind is represented relative to the inertial frame. One of the methods for modeling the prevailing wind is to use actual flight data. Lewis [67] presents a method based on flight data for obtaining prevailing wind. The total wind is calculated by using velocity triangle vector and flight data of both aircraft previously recorded from an aerial refueling test flight of KC-135R and a Learjet 25. The velocity equation is formulated as

$$W_I = \dot{r}_{B_T} - \mathbf{R}^{\mathbf{T}}_{\mathbf{B_T I}} \mathbf{R}_{\mathbf{B_T W_T}} [V_T \;\; 0 \;\; 0]^T \tag{2.32}$$

where W_I represents the total wind vector in the inertial frame, V_T is the tanker aircraft's airspeed obtained from an airdata sensor, $\dot{r}_{B_T}$ is the tanker aircraft's velocity in the inertial frame obtained from a GPS unit, $\mathbf{R_{B_T I}}$ is the rotational matrix from the inertial frame to the tanker's body frame obtained from a IMU, and $\mathbf{R_{B_T W_T}}$ is the rotational matrix from the tanker's wind frame to the body frame and constructed from the side slip angle and angle of attack, which are measured by airdata sensor. Prevailing wind is considered to be the mean value after the turbulence effect is removed from the total wind calculated by using Eq. (2.32).

2.3.2 Modeling of Turbulence

The Dryden model PSD functions are used to model turbulence as a random process [67, 68]. Zero mean Gaussian white noise with unity power spectrum is taken as input. The differential equations of the Dryden model are

$$\begin{bmatrix} \dot{x}_{w1} \\ \dot{x}_{w2} \\ \dot{x}_{w3} \\ \dot{x}_{w4} \\ \dot{x}_{w5} \end{bmatrix} = \begin{bmatrix} -\frac{V}{L_u} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{V}{L_v} & -2\frac{V}{L_v} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{V}{L_w} & -2\frac{V}{L_w} \end{bmatrix} \begin{bmatrix} x_{w1} \\ x_{w2} \\ x_{w3} \\ x_{w4} \\ x_{w5} \end{bmatrix} + \begin{bmatrix} \sqrt{\frac{2V\sigma_u^2}{\pi L_u}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \tag{2.33}$$

where (x_{w1}, x_{w2}, x_{w3}, x_{w4}, x_{w5}) are states of the Dryden model, V is the airspeed of the aircraft, and (η_1, η_2, η_3) are the zero mean Gaussian white noise with unity power spectrum.

The wind components due to turbulence in the body frame are given as

$$W_{tur} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{\sigma_v}{\sqrt{\pi}}\left(\frac{V}{L_v}\right)^{1.5} & \sigma_v\sqrt{\frac{3V}{\pi L_v}} & 0 & 0 \\ 0 & 0 & 0 & \frac{\sigma_w}{\sqrt{\pi}}\left(\frac{V}{L_w}\right)^{1.5} & \sigma_w\sqrt{\frac{3V}{\pi L_w}} \end{bmatrix} \begin{bmatrix} x_{w1} \\ x_{w2} \\ x_{w3} \\ x_{w4} \\ x_{w5} \end{bmatrix} \tag{2.34}$$

where σ_u, σ_v, and σ_w are the RMS turbulence intensities, L_u, L_v, and L_w represent the scale lengths. From Ref. [67], the intensity and scale length values are taken as $\sigma_u = \sigma_v = \sigma_w$ and $L_u = L_v = L_w$ due to isotropy for turbulence above 20,000 ft. Scale lengths are set to 533.4 m, and turbulence intensities are set to 0.39 m/s (1.3

ft/s) for the altitude of 22,998 ft. (7010 m). These values represent a light turbulence condition [68].

2.3.3 Modeling of Wake Vortex Induced Wind

The wing and horizontal tail of the tanker aircraft are modeled as horseshoe vortices (Fig. 1.1). Tail vortices rotate outward generating a negative lift and wing vortices rotate inward generating a positive lift. The magnitude of the wind induced by each filament is

$$W_r = \frac{\Gamma\, r}{2\pi(r^2 + r_c^2)} \left[1 - \exp(-\frac{r^2}{4\upsilon\tau})\right] \tag{2.35}$$

where Γ is the vortex strength, r is the radial distance from the point of interest to the filament, r_c is the radius of the vortex core, υ is the viscosity parameter specified, and τ is the distance to the filament over the speed of the tanker. The strength of a vortex filament is given as

$$\Gamma = \frac{L}{\rho\, V\, (\pi/4)\, b} \frac{\cos\gamma_1 + \cos\gamma_2}{2} \tag{2.36}$$

where L is the lift generated by the wing or tail, ρ is the air density, V is the airspeed of the tanker, b is the span of the wing or tail, and γ_i are the angles of the vectors from the filament to the point of interest.

During aerial refueling operations, the tanker's wake vortex creates a nonuniform wind field over the receiver aircraft. In Ref. [64], a method is introduced to model the effect of the nonuniform wind over the dynamics of the receiver aircraft. This method approximates the nonuniform wind distribution by using uniform wind component ($(W_i)_{eff}$) and uniform wind gradient ($(\partial W_i/\partial i)_{eff}$). All three components of wind distribution is approximated using the same approach, and three components of effective translational wind velocity and three components of effective rotational

wind velocity can be computed. The effective rotational wind components, which are used in Eq. (2.13), are formulated as

$$p_{eff} = \left(\frac{\partial W_z}{\partial y}\right)_{eff} - \left(\frac{\partial W_y}{\partial z}\right)_{eff} \tag{2.37}$$

$$q_{eff} = \left(\frac{\partial W_x}{\partial z}\right)_{eff} - \left(\frac{\partial W_z}{\partial x}\right)_{eff} \tag{2.38}$$

$$r_{eff} = \left(\frac{\partial W_y}{\partial x}\right)_{eff} - \left(\frac{\partial W_x}{\partial y}\right)_{eff} \tag{2.39}$$

2.4 Controller Design for the Tanker Aircraft

The controller for the tanker aircraft is designed in Ref. [66] and the equations are summarized in this section. The controller flies the aircraft in a pre-specified trajectory including straight level and turn segments, at a constant altitude and with a constant airspeed. The controller of the tanker aircraft is based on LQR-based approach using gain scheduling technique with the altitude and speed hold, and yaw rate tracking. Commanded airspeed and commanded yaw rate are used in gain scheduling, and there are six nominal conditions associated with the gain scheduling. Equations of motion of the tanker aircraft are linearized and LQR-controller based on the linear model is used. The linear model for each nominal condition is expressed as

$$\Delta \dot{\mathrm{x}}_T = \mathbf{A}\Delta \mathrm{x}_T + \mathbf{B}\Delta u \tag{2.40}$$

where Δ terms indicates deviation from the nominal condition, $\Delta \mathrm{x}_T$ is the perturbations of the system state vector of the tanker aircraft from their nominal values, and Δu is the perturbations of the control input. The linearized state vector vector is

$$\Delta \mathrm{x}_{_T} = [\Delta V_{_T}\ \Delta\beta_{_T}\ \Delta\alpha_{_T}\ \Delta p_{_T}\ \Delta q_{_T}\ \Delta r_{_T}\ \Delta\psi_{_T}\ \Delta\theta_{_T}\ \Delta\phi_{_T}\ \Delta x_{_T}\ \Delta y_{_T}\ \Delta z_{_T}]^T \tag{2.41}$$

and the linearized control input vector is

$$\Delta u = [\Delta\delta_{a_T}\ \Delta\delta_{e_T}\ \Delta\delta_{r_T}\ \Delta_{T_T}]^T \tag{2.42}$$

Integral control terms are added to LQR-controllers in order to improve steady-state tracking performance. For the integral control terms, the error signals are defined as

$$\dot{e} = \Delta \mathrm{y}_T - \Delta \mathrm{y}_c \tag{2.43}$$

where error is defined as

$$e = \begin{bmatrix} \Delta V - \Delta V_c \\ \Delta z - \Delta z_c \\ \Delta\dot{\psi} - \Delta\dot{\psi}_c \end{bmatrix} \tag{2.44}$$

the subscript c in Eq. (2.44) indicates the commanded signal. By combining the Eq. (2.40) and Eq. (2.43), the augmented system equation can be obtained as

$$\begin{bmatrix} \Delta\dot{\mathrm{x}}_T \\ \dot{e} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathrm{x}_T \\ e \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \Delta u + \begin{bmatrix} 0 \\ -\Delta \mathrm{y}_c \end{bmatrix} \tag{2.45}$$

With the feedback gain matrices obtained through the LQR method, the control law is formulated as

$$\Delta u_i = -\mathrm{K}_{\mathbf{i}} \cdot \Delta \mathrm{x}_{T_i} - \mathrm{K}_{\mathbf{e,i}} \cdot e \tag{2.46}$$

where

$$u_i = u_{0,i} + \Delta u_i \tag{2.47}$$

$$\Delta \mathrm{x}_{T_i} = \mathrm{x}_T - \mathrm{x}_{T_{0,i}} \tag{2.48}$$

where i represents corresponding nominal conditions, and $[\mathrm{K_i}\ \mathrm{K_{e,i}}]$ is the augmented state feedback gain matrix obtained by minimizing the cost function in LQR. A Lagrange interpolation is used between linear controllers associated with the nominal conditions in implementing the gain scheduling. The gain scheduling controller equation is given as

$$\begin{aligned} u = & \frac{(\dot{\psi}_c - \dot{\psi}_2)(\dot{\psi}_c - \dot{\psi}_3)(V_c - V_2)}{(\dot{\psi}_1 - \dot{\psi}_2)(\dot{\psi}_1 - \dot{\psi}_3)(V_1 - V_2)}\ u_1 + \frac{(\dot{\psi}_c - \dot{\psi}_1)(\dot{\psi}_c - \dot{\psi}_3)(V_c - V_2)}{(\dot{\psi}_2 - \dot{\psi}_1)(\dot{\psi}_2 - \dot{\psi}_3)(V_1 - V_2)}\ u_3 \\ & + \frac{(\dot{\psi}_c - \dot{\psi}_1)(\dot{\psi}_c - \dot{\psi}_2)(V_c - V_2)}{(\dot{\psi}_3 - \dot{\psi}_1)(\dot{\psi}_3 - \dot{\psi}_2)(V_1 - V_2)}\ u_5 + \frac{(\dot{\psi}_c - \dot{\psi}_2)(\dot{\psi}_c - \dot{\psi}_3)(V_c - V_1)}{(\dot{\psi}_1 - \dot{\psi}_2)(\dot{\psi}_1 - \dot{\psi}_3)(V_2 - V_1)}\ u_2 \\ & + \frac{(\dot{\psi}_c - \dot{\psi}_1)(\dot{\psi}_c - \dot{\psi}_3)(V_c - V_1)}{(\dot{\psi}_2 - \dot{\psi}_1)(\dot{\psi}_2 - \dot{\psi}_3)(V_2 - V_1)}\ u_4 + \frac{(\dot{\psi}_c - \dot{\psi}_1)(\dot{\psi}_c - \dot{\psi}_2)(V_c - V_1)}{(\dot{\psi}_3 - \dot{\psi}_1)(\dot{\psi}_3 - \dot{\psi}_2)(V_2 - V_1)}\ u_6 \end{aligned} \tag{2.49}$$

2.5 Controller Design for the Receiver Aircraft

While the tanker aircraft flies through its pre-specified trajectory, the receiver aircraft's position should be controlled relative to the tanker during aerial refueling operation. For this purpose, a relative position tracking controller is designed for the receiver aircraft. Similar to the tanker aircraft's controller, a state-feedback and integral control method with gain scheduling approach is used. The gains are calculated by using LQR technique at the pre-defined six nominal conditions. The tanker's commanded airspeed and yaw rate are used as scheduling variables. Equations of motion of the receiver aircraft are linearized, and the linearized model for each nominal condition is expressed as

$$\Delta \dot{\mathrm{x}}_R = \mathbf{A}\Delta \mathrm{x}_R + \mathbf{B}\Delta u \tag{2.50}$$

where Δ terms indicates deviation from the nominal condition, $\Delta \mathrm{x}_R$ is the perturbation of the system state vector of the receiver aircraft from their nominal values, and Δu is the perturbation of the control input. The linearized state vector vector is

$$\Delta \mathrm{x}_{_R} = [\Delta V_{_R}\ \Delta \beta_{_R}\ \Delta \alpha_{_R}\ \Delta p_{_R}\ \Delta q_{_R}\ \Delta r_{_R}\ \Delta \psi_{_R}\ \Delta \theta_{_R}\ \Delta \phi_{_R}\ \Delta \xi_1\ \Delta \xi_2\ \Delta \xi_3]^T \tag{2.51}$$

and linearized control input vector is given as

$$\Delta u = [\Delta \delta_a\ \Delta \delta_e\ \Delta \delta_r\ \Delta T\ \Delta \delta_y\ \Delta \delta_z]^T \tag{2.52}$$

where T is the thrust, (δ_y, δ_z) are engine nozzle deflections for an aircraft capable of thrust vectoring control and (δ_a, δ_e, δ_r) are the aileron, elevator and rudder deflections, respectively. In this dissertation, the engine nozzle deflections are set to zero as thrust vectoring capability is not used. For the integral control of the relative position, the error signals are defined as

$$\dot{e} = \Delta \mathrm{y}_{_R} - \Delta \mathrm{y}_c \tag{2.53}$$

where error is defined as

$$e = \begin{bmatrix} \Delta \xi_1 - \Delta \xi_{1,c} \\ \Delta \xi_2 - \Delta \xi_{2,c} \\ \Delta \xi_3 - \Delta \xi_{3,c} \end{bmatrix} \tag{2.54}$$

the subscript c indicates the commanded signal. By combining the Eq. (2.50) and Eq. (2.53), the augmented system equation can be obtained as

$$\begin{bmatrix} \Delta \dot{\mathrm{x}}_R \\ \dot{e} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathrm{x}_R \\ e \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \Delta u + \begin{bmatrix} 0 \\ -\Delta \mathrm{y}_c \end{bmatrix} \tag{2.55}$$

With the feedback gain matrices obtained through the LQR method, the control law is formulated as

$$\Delta u_i = -\mathrm{K}_{\mathbf{i}} \cdot \Delta \mathrm{x}_{R_i} - \mathrm{K}_{\mathbf{e,i}} \cdot e \tag{2.56}$$

where

$$u_i = u_{0,i} + \Delta u_i \tag{2.57}$$

$$\Delta \mathrm{x}_{R_i} = \mathrm{x}_R - \mathrm{x}_{R_{0,i}} \tag{2.58}$$

where i represents corresponding nominal conditions, and $[\mathrm{K}_{\mathbf{i}}\ \mathrm{K}_{\mathbf{e,i}}]$ is the augmented state feedback gain matrix obtained by minimizing the cost function in LQR. A Lagrange interpolation is used between linear controllers associated with the nominal conditions in implementing the gain scheduling. The gain scheduling controller is given as

$$\begin{aligned} u = {} & \frac{(\dot{\psi}_c - \dot{\psi}_2)(\dot{\psi}_c - \dot{\psi}_3)(V_c - V_2)}{(\dot{\psi}_1 - \dot{\psi}_2)(\dot{\psi}_1 - \dot{\psi}_3)(V_1 - V_2)}\, u_1 + \frac{(\dot{\psi}_c - \dot{\psi}_1)(\dot{\psi}_c - \dot{\psi}_3)(V_c - V_2)}{(\dot{\psi}_2 - \dot{\psi}_1)(\dot{\psi}_2 - \dot{\psi}_3)(V_1 - V_2)}\, u_3 \\ & + \frac{(\dot{\psi}_c - \dot{\psi}_1)(\dot{\psi}_c - \dot{\psi}_2)(V_c - V_2)}{(\dot{\psi}_3 - \dot{\psi}_1)(\dot{\psi}_3 - \dot{\psi}_2)(V_1 - V_2)}\, u_5 + \frac{(\dot{\psi}_c - \dot{\psi}_2)(\dot{\psi}_c - \dot{\psi}_3)(V_c - V_1)}{(\dot{\psi}_1 - \dot{\psi}_2)(\dot{\psi}_1 - \dot{\psi}_3)(V_2 - V_1)}\, u_2 \\ & + \frac{(\dot{\psi}_c - \dot{\psi}_1)(\dot{\psi}_c - \dot{\psi}_3)(V_c - V_1)}{(\dot{\psi}_2 - \dot{\psi}_1)(\dot{\psi}_2 - \dot{\psi}_3)(V_2 - V_1)}\, u_4 + \frac{(\dot{\psi}_c - \dot{\psi}_1)(\dot{\psi}_c - \dot{\psi}_2)(V_c - V_1)}{(\dot{\psi}_3 - \dot{\psi}_1)(\dot{\psi}_3 - \dot{\psi}_2)(V_2 - V_1)}\, u_6 \end{aligned} \tag{2.59}$$

CHAPTER 3

AIRDATA SENSOR MEASUREMENTS IN NONUNIFORM WIND FIELD

This chapter provides information about the effect of nonuniform wind field on the airdata sensor measurements. Considering that the receiver aircraft is exposed to nonuniform wind field induced by tanker aircraft in aerial refueling operation, the influence of the wind field on the airdata sensors is presented, as it causes difference in sensor readings even the sensors are mounted on the same aircraft. First, information about the airdata sensors and their working principles for measuring airdata variables is presented to understand better why their readings are affected by the air surrounds them. Also, information about the positions of the receiver aircraft inside the nonuniform wind field during the aerial refueling operation is provided. Lastly, the procedure for creating maps of wind field in terms of magnitude and direction with respect to the receiver's position relative to the tanker, is explained.

3.1 General Introduction to Airdata Sensors

Basic airdata sensor is used to retrieve information about physical characteristics of air that surrounds the aircraft. Modern airdata sensors usually provide the air pressure, airflow and temperature measurements. Airdata sensors have the advantage that multiple sensors can be mounted for having redundant measurements which are safety monitored [69]. Today's aircraft use Air Data Computers (ADC) which process the airdata sensor measurements and compute the following variables of the aircraft

- Airspeed
- Side Slip Angle

- Angle of Attack
- Static/Total Pressure
- Outside/Total Temperature

The airdata sensors are mounted on the aircraft in a location where measured air flow is in its natural form, generally side of the nose (Fig. 3.1). Typical airdata sensor (probe) (Fig. 3.2) has the following components; pressure head, temperature head, mast, and heaters [70].

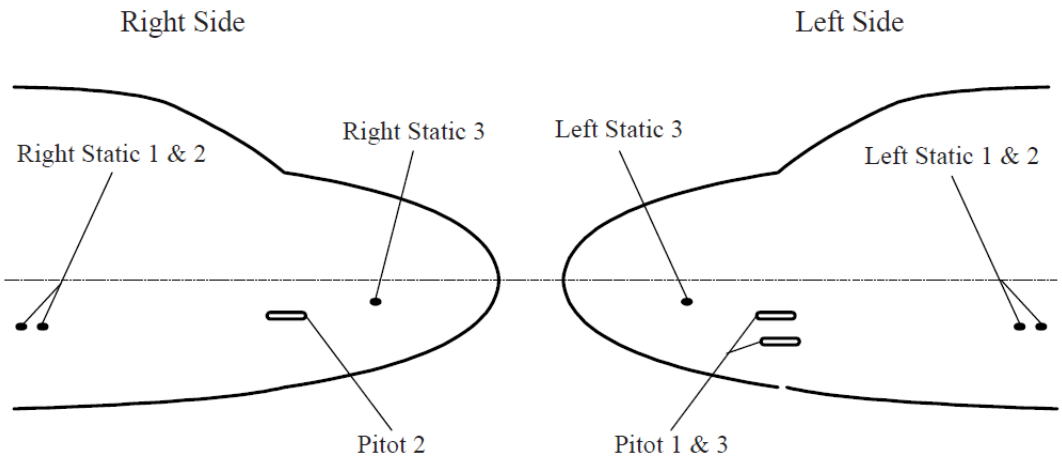


Figure 3.1. Typical Location for Airdata Sensors [69].

Airdata sensor is one of the core elements in avionic systems, and they are required in all today's aircraft, civilian or military [71]. In order to continue flying safely, the reliable provision of air data of the aircraft should be provided, and that information provided by airdata systems is very essential for some key avionic sub-systems which enable the (auto)pilot to carry out his mission [69, 71]. For instance, onboard Guidance, Navigation, and Control (GNC) sub-system uses the information about the movement of the aircraft in the atmosphere, which is provided by airdata

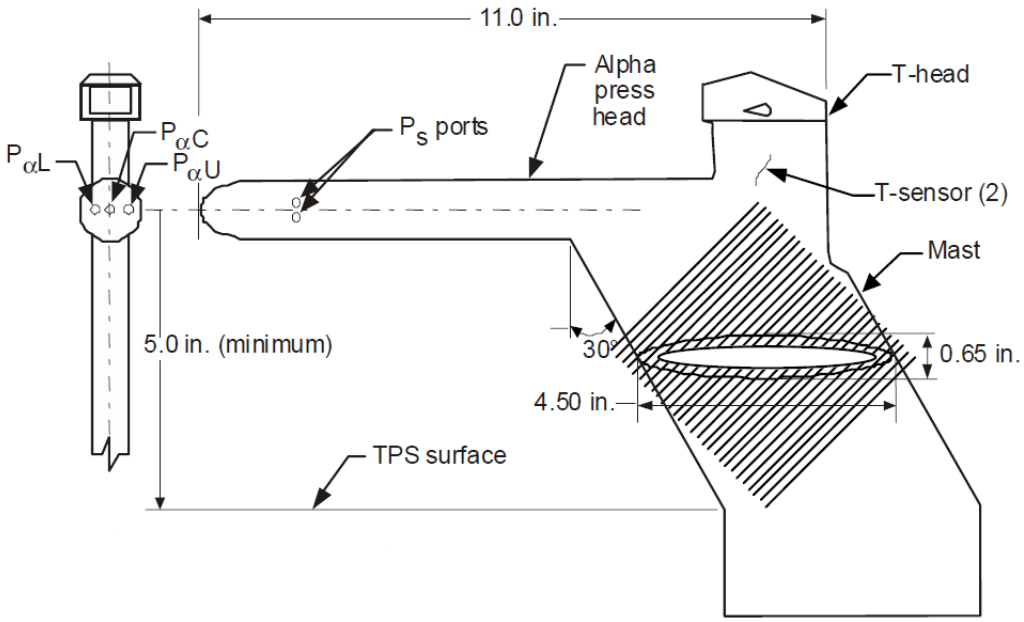


Figure 3.2. Typical Airdata Sensor (Probe) [70].

sensors [70]. Thus, airdata systems should provide information about the ambient atmosphere and the state of the aircraft flight relative to ambient air by combining information with other onboard or remote measurements [72]. Airdata sensors provide pressure values as electrical signals on the altimeter, vertical speed indicator, airspeed indicator, and Machmeter to be used to obtain airspeed and altitude [69, 73]. Static pressure can be described as the local pressure surrounding the aircraft and varies with altitude [69], and total pressure can be described as the sum of static pressure and the pressure caused by the forward motion of the aircraft [69]. Side slip angle and angle of attack values are obtained from measured local flow angles. Also, true airspeed and other parameters can be obtained using algorithms in air data computers given the pressure and flow information.

Airspeed sensors need to quantify two main information. The first one is the information of air that surrounds the aircraft [72]. For this information, the pressure

measurement (or density value) and air temperature is needed. The second one is information about aerodynamic state of the aircraft, which is determined by the velocity vector relative to surrounding air in a body fixed coordinate system. Three scalar quantities are required in order to determine velocity vector, (i) magnitude of velocity V (ii) angle of attack α and (iii) side slip angle β [72].

Airdata sensors consist of two main components, i.e. pressure sensor and temperature sensor [71]. The temperature sensor is relatively simple and straightforward device that is composed of a simple resistance bridge where one resistant arm of that bridge is exposed to airstream. Pressure sensors, on the other hand, require an extremely high accuracy and they are affected by acceleration, humidity, shock, vibration and temperature changes. Thus it takes long and expensive process to develop pressure sensors [71]. Different companies and organizations have developed wide variety of air data pressure sensors. Generally pitot and static probes are used to measure static and pitot pressure. Pitot probe (tube) is named after his inventor French mathematician and scientist Henri Pitot.

Pitot probes with separate static pressure orifices are used in some civil aircraft, and they are placed between the nose and the wing. Generally, L type pitot-static probes (Fig. 3.2) are used in civil transport aircraft (e.g., Boeing 747) [71]. In Fig. 3.3, a typical flight test noseboom is shown that measures local flow angles and pitot-static pressures [73].

Another example of pitot-static probe is shown in Fig. 3.4. The face of this probe has pitot and static orifices, as well as the orifices for side slip angle measurements [69]. The principle is to use the differential output ($P\beta 1 - P\beta 2$) for corresponding measurement. Similarly, face of a probe with angle of attack orifices is given in Fig. 3.5, which gives differential output ($P\alpha 1 - P\alpha 2$) for angle of attack measurements.

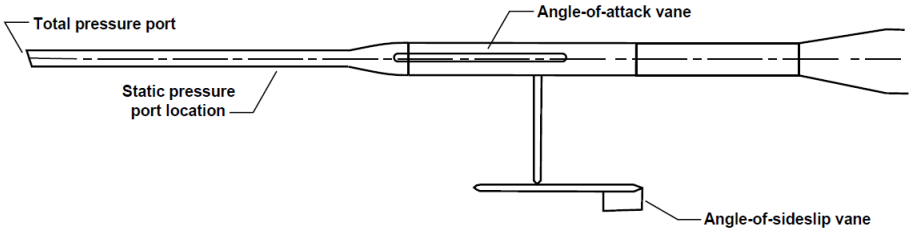


Figure 3.3. Typical Airdata Noseboom [73].

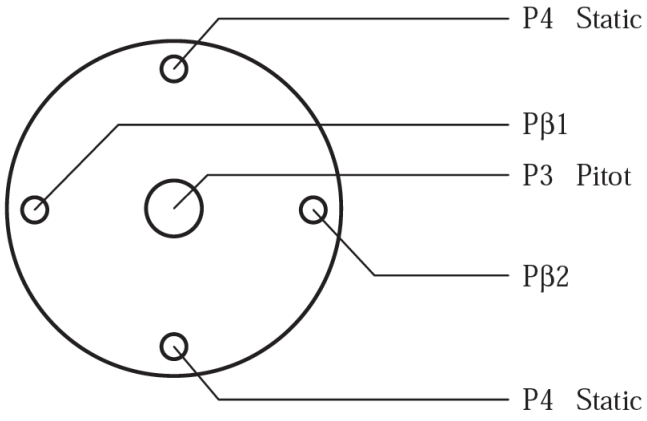


Figure 3.4. Side Slip Angle Measurement [69].

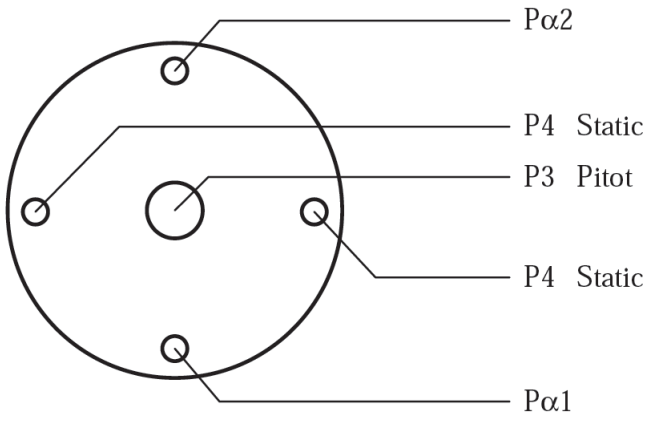
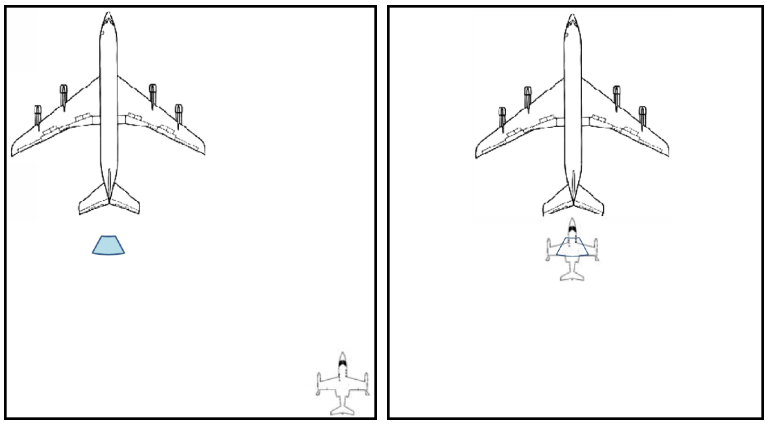


Figure 3.5. Angle of Attack Measurement [69].

3.2 Positions of Receiver Aircraft during Aerial Refueling Operation

In aerial refueling operation, the receiver aircraft is required to fly relative to the tanker aircraft. This flight includes two positions, i.e. observation and contact

position, as well as transition between these positions (Fig. 3.6). The contact position is where the actual fuel transfer takes place. It is right behind the tanker, slightly below and behind the tail of the tanker. At this position, the receiver is in the downwash region of the vortex induced nonuniform wind. Besides the contact position, the observation position and the area between these two are also important. During aerial refueling operations, the receiver aircraft maneuvers from the observation position to the contact position. The observation position is at the same altitude as the tanker's on the right side of the tanker, outside the wake of the tanker. Thus, the effect of the wake vortices in the wind field the receiver flies in is minimal.



(a) (b)

Figure 3.6. Aerial Refueling Positions (a) Observation Position (b) Contact Position [74].

3.3 Maps of Wind Field in Terms of Magnitude and Direction

When receiver aircraft flies in aerial refueling operation, it is exposed to nonuniform wind field. Wind vector has varying magnitude and direction with the relative position with respect to the tanker. To construct maps illustrating the variation of

airdata variables (V, β, α), a set of discrete points are defined around the contact position. This set of points is referred to as "the grid".

The grid is defined around the contact position with 2 meters margins in y and z directions (Fig. 3.7). It should be noted that coordinates of the contact position is $y = 0$ and $z = 8.5$ (meters) relative to the tanker aircraft. Thus, the grid is defined between -2 and 2 meters in y direction, and between 6.5 and 10.5 meters in z direction. In both directions, 0.2 meters step size is used. This adds up to equally spaced 441 points within a range of 4 meters, where the center of the grid is the contact position. The points in the grid varies in y and z directions only. In this study, variation of the grid in x direction is not taken into consideration due the fact that the wind vortex model used has negligible variation in x direction of the specified contact position of the flight.

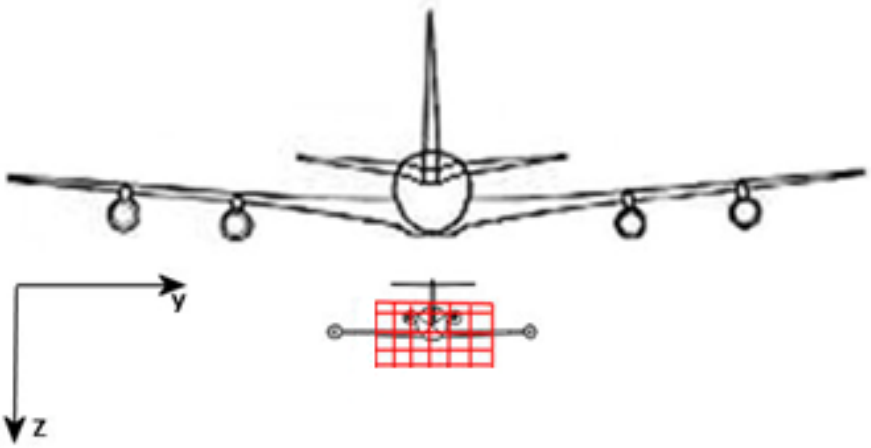


Figure 3.7. The Representation of the Grid for Receiver Aircraft behind the Tanker.

In order to create maps of wind field in terms of magnitude and direction, static simulations are run by placing the receiver (the origin of the receiver's body frame)

at each point of the grid and airdata variables (airspeed V, angle of attack α, sideslip angle β) are recorded from each sensor. Figure 3.8 shows the placements of three identical airdata sensors on the Learjet-25. Since the nonuniform wind has significant variation our the receiver aircraft, the airdata sensors placed at different locations on the receiver record different variables at each point of the grid.

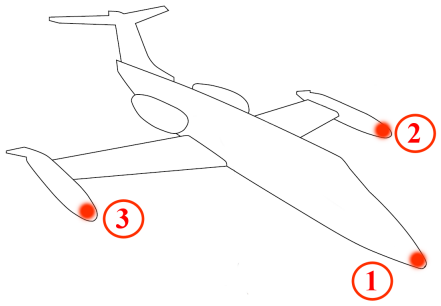


Figure 3.8. Placement of the Sensors.

In static simulations, the tanker and receiver aircraft are placed statically and the position and orientation are kept fixed. In other words, static simulations are similar to wind tunnel tests, where two aircraft models are placed in wind tunnel with fixed position and orientation relative to each other. Also, the effects of the turbulence and prevailing winds are turned off, and the measurement noise is not added to the sensor readings in order to isolate the nonuniform wind effect. In this step, the receiver aircraft flies at a point in the grid, and airdata measurements are recorded and saved with the associated coordinate (y, z) information. Then, same procedure is repeated for other points in the grid until all 441 locations are covered.

Figures 3.9-3.11 show the maps of the airdata variables. As seen from the figures, each variable from a given sensor varies as the position of the aircraft on the

grid changes. Further, each sensor gives different reading at a given point as well as different variation over the grid.

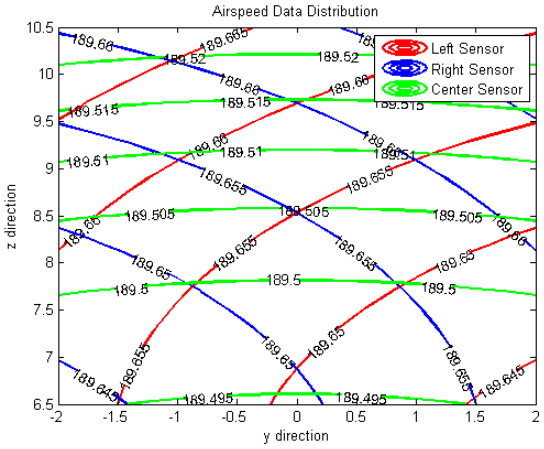


Figure 3.9. Distribution of Airdata Measurements - Airspeed.

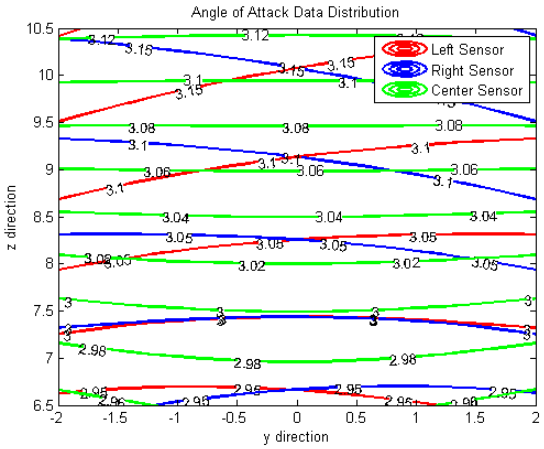


Figure 3.10. Distribution of Airdata Measurements - Angle of attack.

For fuel transfer, the receiver aircraft needs to stay at the contact position. However, an entire refueling operation requires the receiver to fly in formation with the tanker at the observation position and transition to the contact position. Thus,

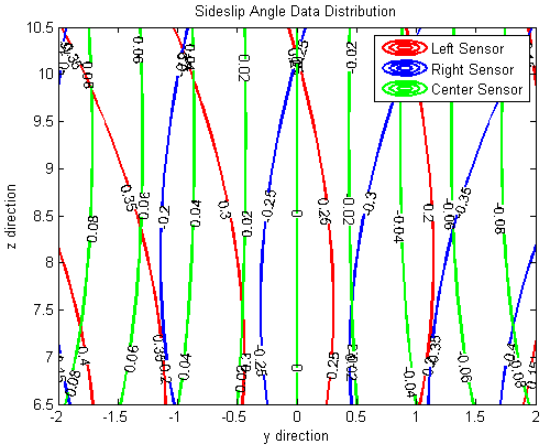


Figure 3.11. Distribution of Airdata Measurements - Side slip angle.

a larger set of points is defined to cover the whole y and z ranges. This set of points is referred to as "the extended grid" (Fig. 3.12). The extended grid has a 20 meters length in z direction, and 70 meters length in y direction. The airdata measurements are saved between -4 to 68 meters in y direction, and between -6 and 14 meters in z direction. The points are placed on the extended grid with 1 meter spacing in both y- and z- directions. This adds up to 1533 equally spaced points.

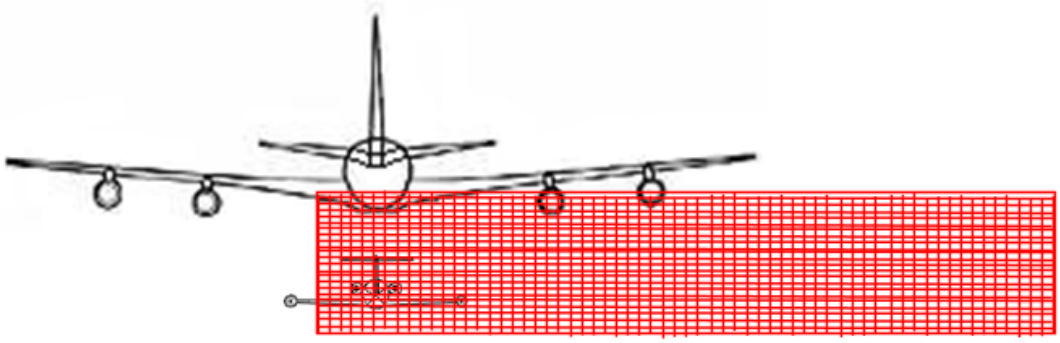


Figure 3.12. The Representation of the Extended Grid for Receiver Aircraft behind the Tanker.

Same as the first grid, simulations are performed at every point on the extended grid, and the airspeed (V), angle of attack (α), sideslip angle (β) measurements are recorded from each sensor. Same as in simulations for the first grid, the effects of the turbulence and prevailing winds are turned off, and the measurement noise is not added to the sensor readings in order to isolate the nonuniform wind effect.

After the simulations are run for all points in the extended grid, maps are created. Distributions of all airdata measurements are presented in Figs. 3.13 to 3.15. In each plot, three sensors' measurement for one variable is shown as contour plots.

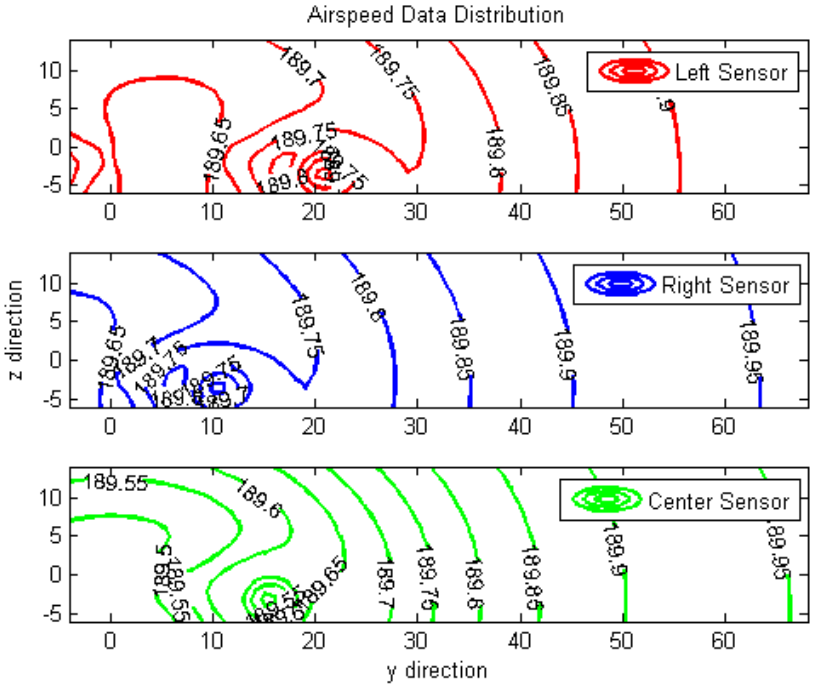


Figure 3.13. Distribution of Airdata Measurements - Airspeed in Extended Grid.

The maps discussed in this chapter are used in both estimation of relative position and the improvement of FDI algorithm. In relative position estimation part of

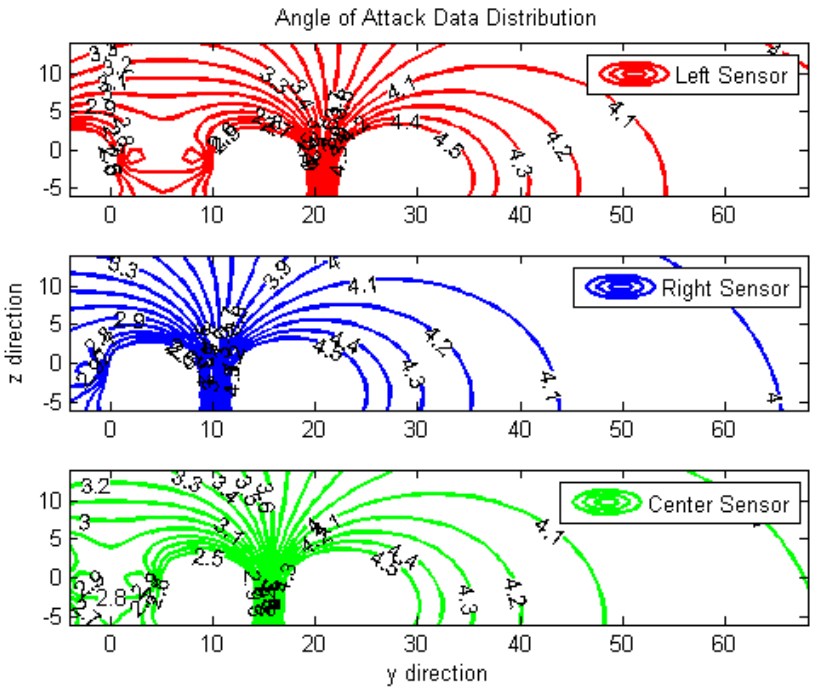


Figure 3.14. Distribution of Airdata Measurements - Angle of attack in Extended Grid.

this research, the instant sensor readings are compared with the corresponding map and a set of possible lateral (y) and vertical (z) position information is determined. The algorithms developed to process the maps to estimate relative position are discussed in Chapter 4. For the fault diagnosis part, the aim is to find expected values of the measurements at a given position by using these maps. The current relative position is given as an input, and then the expected values of airdata measurements are found by reading the maps, and this process is repeated for other sensors and for other variables. The details of this procedure are given in Chapter 5.

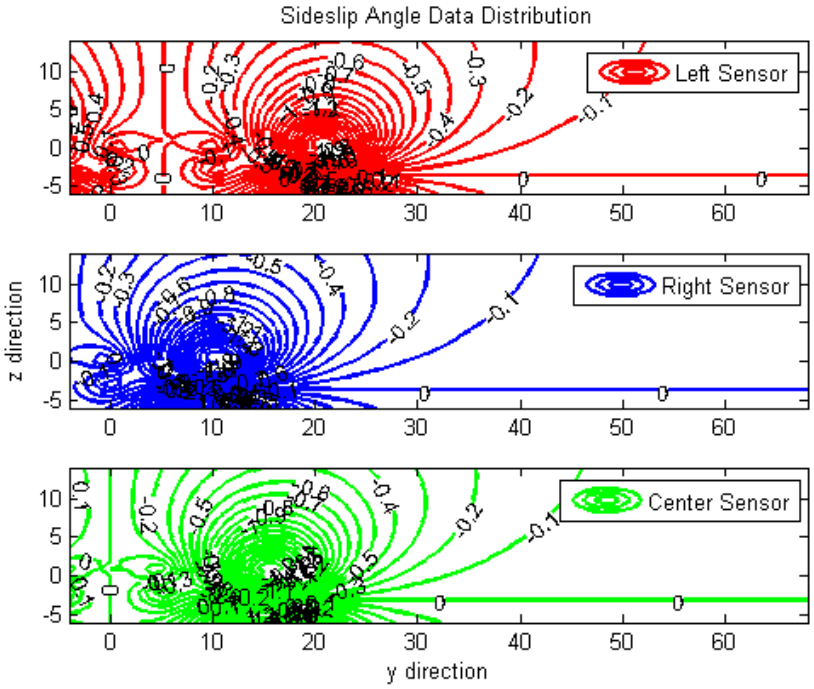


Figure 3.15. Distribution of Airdata Measurements - Side slip angle in Extended Grid.

CHAPTER 4

RELATIVE POSITION ESTIMATION USING AIRDATA SENSORS

This research uses airdata measurements to determine where the airdata sensor is relative to the tanker aircraft. If a map of the wind field is available in terms of the magnitude and direction as a function of the relative position, the inverse of this map may be used to estimate the relative position given the wind velocity vector measurement. For this idea to work, there should be a one-to-one mapping from relative position to airdata sensor measurements. This is to be able to generate the inverse map that reads airdata measurements and gives the relative position without any ambiguity. This research assumes the availability of the maps of the wind field. The maps can be built using different methods such as analytic models, CFD simulations and wind tunnel tests. Section 3.3 presents the maps constructed from a model of vortex-induced wind field and used in this research. Before developing any algorithm implementing this idea, the following questions should be answered:

i. Whether there is a one-to-one mapping between position and any one of the three airdata sensor measurements
ii. If there is, which variable can give the most accurate and reliable inverse map
iii. If there is not, is it possible to use the maps of the wind field in terms of two or three of airdata variables measured at the same position on the receiver aircraft to determine the relative position
iv. If that is not possible, what is the minimum number of sensors placed at different locations to determine relative position without ambiguity

v. Is there any benefit of using more than the minimum required number of sensors in terms of accuracy and reliability

Based on the answers to these questions, several algorithms are developed that compute the relative position based on airdata sensor measurements. The estimation procedure is as follows. The instant airdata measurements are given as an input to the associated wind field map, and the output will be isolines representing possible $y - z$ positions. Considering different measurements from different sensors, there are different contour lines for each input. To estimate the relative position, intersection between individual contour lines are used. Various algorithms developed herein differ from each other based on the method of finding the intersections, number of sensor and variables used, and the decision process for estimation. Details of each algorithm are given in next subsection.

4.1 Position Estimation Algorithms

The proposed position estimation procedure consists of taking instant airdata measurements and comparing them with the corresponding map of the wind field. For each input, different contour level curves can be obtained, and the intersection of these contour lines gives the possible positions of the aircraft at that instant. Mainly, the contour lines are formed by set of points. In order to find the intersection of different contour lines, the points that form a given contour line are processed. The following paragraphs describe the two methods developed to determine if there is an intersection between two given contour lines and, if so, where the intersection point is.

The first method for finding the intersection takes a pair of consecutive points from each contour line. Each pair of points are connected to each other with a straight line. Two lines from two pairs are then checked if there is an intersection. Afterwards,

one pair of points is kept, and another closest pair is taken from other contour line until all points are checked. This process is repeated for all points of the contour lines. The accuracy of finding the intersection point by using this method depends on the number of points that represent the contour line, yet this method is fast and gives accurate results. This method is referred to as "intersection method-1 (IM-1)" in this research. Figure 4.1 shows an illustration of the technique used in intersection method-1.

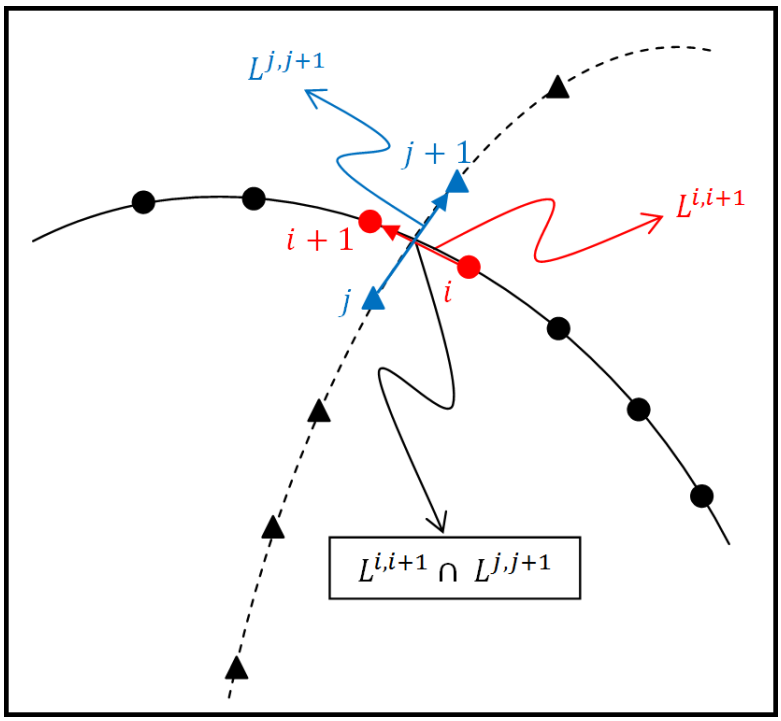


Figure 4.1. The Illustration of Intersection Method-1.

The second method also relies on the comparison of points from the contour lines. In this method, one point from each contour line is taken and the distance between those is calculated. Similar to the first method, one point from first contour line is kept and the next point from the second contour line is taken until all points

are checked, and this process repeated for all points of the first contour line. Then, the minimum of the calculated distances is selected and the middle point between the points with minimum distance is obtained as the intersection point. The drawback of this method is that it gives a result even though two contour lines do not intersect. Basically, this method performs successfully under the assumption that there exists an intersection point between two contour lines. Also, unless the distances between the points are the same, this method cannot provide more than one intersection point, despite of existence of multiple intersection points. This method is referred to as "intersection method-2 (IM-2)" in this dissertation. An illustration of intersection method-2 is given in Fig. 4.2. The IM-2 is relatively faster than IM-2 in sense of computational speed due to the fact IM-1 needs not to only to compare all points to each other, also needs to find intersection point of two vectors. Although this difference is not significant, the computational time difference will grow exponentially as the number of points increases. Also, the main reason for introducing IM-2 is to use it to find common intersection points from different variables.

In addition to the method of finding intersections, the position estimation algorithms differ from each other by the number of sensors used, the number of measurements processed, and the criterion used for choosing the estimated position from set of intersection points. 16 algorithms using various combinations of these factors are described below. It should be noted that there are 3 different airdata measurements from three sensors which gives the total number of the combinations as 511 without taking account of choosing criterion and method of intersection.

Algorithm-1) Measurements of sideslip angle (β) from two sensors (located at left wing tip and right wing tip) and respective contour lines are used. The intersection of these two contour lines is obtained by using intersection method-1 as the position

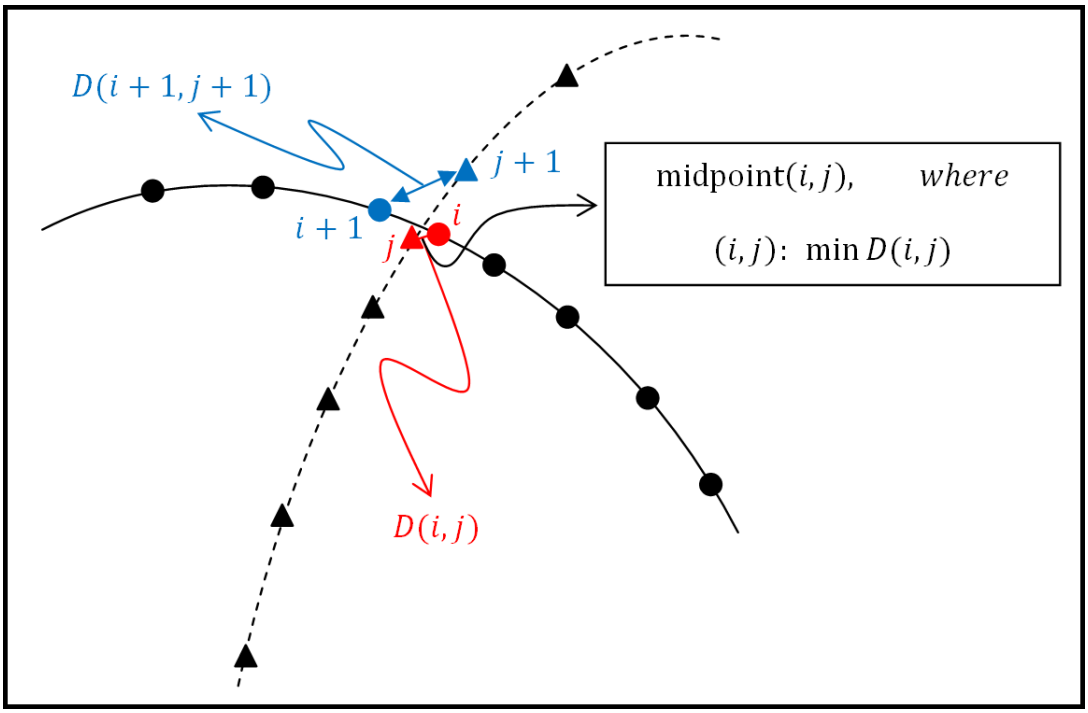


Figure 4.2. The Illustration of Intersection Method-2.

estimation. In the case of multiple intersections of two contour lines, the first element of the set of intersection points is chosen.

Algorithm-2) Measurements of sideslip angle (β) from two sensors (located at left wing tip and right wing tip) and respective contour lines are used. The intersection of these two contour lines is obtained by using intersection method-1 as the position estimation. In the case of multiple intersections of two contour lines, the last element of the set of intersection points is chosen.

Algorithm-3) Measurements of angle of attack (α) from two sensors (located at left wing tip and right wing tip) and respective contour lines are used. The intersection of these two contour lines is obtained by using intersection method-1 as the position estimation. In the case of multiple intersections of two contour lines, the first element of the set of intersection points is chosen.

Algorithm-4) Measurements of angle of attack (α) from two sensors (located at left wing tip and right wing tip) and respective contour lines are used. The intersection of these two contour lines is obtained by using intersection method-1 as the position estimation. In the case of multiple intersections of two contour lines, the last element of the set of intersection points is chosen.

Algorithm-5) Measurements of sideslip angle (β) from two sensors (located at left wing tip and right wing tip) and respective contour lines are used. The intersection of these two contour lines is obtained by using intersection method-1 as the position estimation. In the case of multiple intersections, the default value is assigned for estimated value. The default value indicates that no absolute decision for estimation is made.

Algorithm-6) Measurements of angle of attack (α) from two sensors (located at left wing tip and right wing tip) and respective contour lines are used. The intersection of these two contour lines is obtained by using intersection method-1 as the position estimation. In the case of multiple intersections, the default value is assigned for estimated value.

Algorithm-7) Measurements of angle of attack (α) and sideslip angle (β) from two sensors (located at left wing tip and right wing tip) and respective contour lines are used. The intersection of two contour lines obtained from same airdata measurement is found by using intersection method-1. Position estimation is performed only when intersection points found from angle of attack (α) readings and intersection points found from sideslip angle (β) readings are the same. In the case of having multiple common points, the first element of the set of intersection points is chosen.

Algorithm-8) Measurements of angle of attack (α) and sideslip angle (β) from two sensors (located at left wing tip and right wing tip) and respective contour lines are used. The intersection of two contour lines obtained from same airdata mea-

surement is found by using intersection method-1. The set of distances between the intersection points found by using angle of attack (α) and sideslip angle (β) readings is calculated. Then the pair of closest points is obtained and the center of these points is taken as the estimated position.

Algorithm-9) Measurements of angle of attack (α) and sideslip angle (β) from three sensors (located at nose, left wing tip and right wing tip) and respective contour lines are used. The intersection of two contour lines obtained from same airdata measurement is found by using intersection method-1. The set of distances between the intersection points found from angle of attack (α) and sideslip angle (β) are calculated. Then the pair of closest points is obtained and the center of these points is taken as the estimated position.

Algorithm-10) Measurements of angle of attack (α) and sideslip angle (β) from three sensors (located at nose, left wing tip and right wing tip) and respective contour lines are used. The intersection of two contour lines obtained from same airdata measurement is found by using intersection method-1. The set of distances between the intersection points found from angle of attack (α) and sideslip angle (β) are calculated. Then the pair of closest points is obtained and the center of these points is taken as the estimated position. In the case of only from one variable there is only one intersection point, that point is taken as estimated position. If there are more than one intersection point from one variable, the first element of the set of intersection points is chosen.

Algorithm-11) Measurements of angle of attack (α) and sideslip angle (β) from three sensors (located at nose, left wing tip and right wing tip) and respective contour lines are used. The intersection of two contour lines obtained from same airdata measurement is found by using intersection method-1. The set of distances between the intersection points found from angle of attack (α) and sideslip angle (β) are

calculated. Then the pair of closest points is obtained and the center of these points is taken as the estimated position. In the case of only from one variable there is only one intersection point, that point is taken as as estimated position. If there are more than one intersection point from one variable, the distances between those points are calculated. The distances first are compared with a threshold, epsilon (ϵ). The ones which are smaller than epsilon are compared with each other. The closest pair of points is obtained and the center point of those two is taken as estimated position. If there is no pair of points that are closer than epsilon, the default value is assigned for estimated value.

Algorithm-12) Measurements of angle of attack (α) and sideslip angle (β) from one sensor (located at nose of the aircraft) and respective contour lines are used. Instead of determining intersection points from same variables taken from different sensor, in this algorithm intersection of different variables from the same sensor is used. In the case of multiple intersections, the first element of the set of intersection points is chosen.

Algorithm-13) Measurements of angle of attack (α) and sideslip angle (β) from one sensor (located at nose of the aircraft) and respective contour lines are used. Instead of determining intersection points from same variables taken from different sensor, in this algorithm intersection of different variables from the same sensor is used. In the case of multiple intersections, the last element of the set of intersection points is chosen.

Algorithm-14) Measurements of angle of attack (α) and sideslip angle (β) from three sensors (located at nose, left wing tip and right wing tip) and respective contour lines are used. The intersection of two contour lines obtained from same airdata measurement is found by using intersection method-2. The set of distances between the intersection points found from angle of attack (α) and sideslip angle (β) are

calculated. Then the pair of closest points is obtained and the center of these points is taken as the estimated position.

Algorithm-15) Measurements of angle of attack (α) and sideslip angle (β) from three sensors (located at nose, left wing tip and right wing tip) and respective contour lines are used. The intersection of two contour lines obtained from same airdata measurement is found by using intersection method-1. The set of distances between the intersection points found from angle of attack (α) and sideslip angle (β) are calculated. Then the pair of closest points is obtained and the center of these points is taken as the estimated position. In the case of only from one variable there is only one intersection point, the default value is assigned for estimated value. If there are more than one intersection point from one variable, the distances between those points are calculated. The distances first are compared with a threshold, epsilon (ϵ). The ones which are smaller than epsilon are compared with each other. The closest pair of points is obtained and the center point of those two is taken as estimated position. If there is no pair of points that are closer than epsilon, the default value is assigned for estimated value.

Algorithm-16) Measurements of angle of attack (α) and sideslip angle (β) from three sensors (located at nose, left wing tip and right wing tip) and respective contour lines are used. The intersection of two contour lines obtained from same airdata measurement is found by using intersection method-1. The set of distances between the intersection points found from angle of attack (α) and sideslip angle (β) are calculated. Then the pair of closest points is obtained and the center of these points is taken as the estimated position. In the case of only from one variable there is only one intersection point, the default value is assigned for estimated value. If there are more than one intersection point from one variable, the first element of the set of intersection points is chosen.

Table 4.1. Definitions of Variables for the Algorithm-15 Pseudocode

$\{(y_i, z_i, \beta_i), \ldots\}$	set of points from sensor q
U_q^β	set of all points (y_i, z_i, β_i) from sensor q
U_q^α	set of all points (y_i, z_i, α_i) from sensor q
A_q^β	set of points (y_n, z_n) for given α_n from sensor q
A_q^α	set of points (y_n, z_n) for given β_n from sensor q
a_n^β	elements of set A^β
a_n^α	elements of set A^α
$L_q^{a_j^\beta, a_{j+1}^\beta}$	straight line between elements a_j^β and a_{j+1}^β from sensor q
$L_q^{a_j^\alpha, a_{j+1}^\alpha}$	straight line between elements a_j^α and a_{j+1}^α from sensor q
$L_q^{a_j^\beta, a_{j+1}^\beta} \cap L_r^{a_j^\beta, a_{j+1}^\beta}$	intersection of two lines from sensor q and r
$C_{i,j}^\alpha \cup C_{j,k}^\alpha$	Union of sets $C_{i,j}^\alpha$ and $C_{j,k}^\alpha$
C^β	set of points that are intersection of two lines
C^α	set of points that are intersection of two lines
c_n^β	elements of set C^β
c_n^α	elements of set C^α
$D(c_j, c_k)$	straight line distance between two elements c_j and c_k
$\{c : \phi\}$	The set of all c for which ϕ is true
$\exists c : \phi$	There exists c such that ϕ is true
$\varnothing$	Empty set
$\lvert A_m^\alpha \rvert$	The number of elements in set A_m^α

As seen from the descriptions of the algorithms, the complexity of the algorithms increases by the algorithm number. The differences between the algorithms can be seen in Table 4.2. The increase in complexity is due to the number of sensors and number of variables used, i.e., the number and variety of contour lines used, as well as

the decision criterion for choosing from multiple intersection points. As an example, the pseudocode of Algorithm-15 is provided in Table 4.1 and Algorithm 15.

Table 4.2. Comparison of the Algorithms

	Airdata Variable Used	**Intersection Method**	**Method to find common Intersection**	**In the case of multiple results**	**In the case of single result from one variable only**
Algorithm-1	$A^{\beta}_{2,3}$	IM-1	N/A	First Found	Keep
Algorithm-2	$A^{\beta}_{2,3}$	IM-1	N/A	Last Found	Keep
Algorithm-3	$A^{\alpha}_{2,3}$	IM-1	N/A	First Found	Keep
Algorithm-4	$A^{\alpha}_{2,3}$	IM-1	N/A	Last Found	Keep
Algorithm-5	$A^{\beta}_{2,3}$	IM-1	N/A	Discard	Keep
Algorithm-6	$A^{\alpha}_{2,3}$	IM-1	N/A	Discard	Keep
Algorithm-7	$A^{\beta}_{2,3}$,$A^{\alpha}_{2,3}$	IM-1	IM-2	First Found	Discard
Algorithm-8	$A^{\beta}_{2,3}$,$A^{\alpha}_{2,3}$	IM-1	IM-2	First Found	Keep
Algorithm-9	$A^{\beta}_{1,2,3}$,$A^{\alpha}_{1,2,3}$	IM-1	IM-2	Last Found	Keep
Algorithm-10	$A^{\beta}_{1,2,3}$,$A^{\alpha}_{1,2,3}$	IM-1	IM-2	First Found	Keep
Algorithm-11	$A^{\beta}_{1,2,3}$,$A^{\alpha}_{1,2,3}$	IM-1	IM-2	Compare with ϵ	Keep
Algorithm-12	A^{β}_{1},A^{α}_{1}	IM-1	N/A	First Found	N/A
Algorithm-13	A^{β}_{1},A^{α}_{1}	IM-1	N/A	Last Found	N/A
Algorithm-14	$A^{\beta}_{1,2,3}$,$A^{\alpha}_{1,2,3}$	IM-2	IM-2	First Found	Keep
Algorithm-15	$A^{\beta}_{1,2,3}$,$A^{\alpha}_{1,2,3}$	IM-1	IM-2	Compare with ϵ	Discard
Algorithm-16	$A^{\beta}_{1,2,3}$,$A^{\alpha}_{1,2,3}$	IM-1	IM-2	First Found	Discard

Algorithm 15 The Pseudocode Representation.

Input: $\beta_1(k)$, $\alpha_1(k)$, $\beta_2(k)$, $\alpha_2(k)$, $\beta_3(k)$, $\alpha_3(k)$
Parameters: ϵ, γ (γ is the default value)
Output: $e(k)$ (estimated position $e(k) = (\hat{y}(k), \hat{z}(k))$

$A_1^\beta \equiv \{(y_i, z_i) : U_1^\beta(\beta_1(k))\}$ $A_2^\beta \equiv \{(y_i, z_i) : U_2^\beta(\beta_2(k))\}$ $A_3^\beta \equiv \{(y_i, z_i) : U_3^\beta(\beta_3(k))\}$
$A_1^\alpha \equiv \{(y_i, z_i) : U_1^\alpha(\alpha_1(k))\}$ $A_2^\alpha \equiv \{(y_i, z_i) : U_2^\alpha(\alpha_2(k))\}$ $A_3^\alpha \equiv \{(y_i, z_i) : U_3^\alpha(\alpha_3(k))\}$
for $j = 1$ to $|A_1^\beta| - 1$ **do**
  for $l = 1$ to $|A_2^\beta| - 1$ **do**
    for $p = 1$ to $|A_3^\beta| - 1$ **do**
      $C_{1,2}^\beta \equiv \{c_i^\beta : L_1^{a_j^\beta, a_{j+1}^\beta} \cap L_2^{a_l^\beta, a_{l+1}^\beta}\}$
      $C_{1,3}^\beta \equiv \{c_i^\beta : L_1^{a_j^\beta, a_{j+1}^\beta} \cap L_3^{a_p^\beta, a_{p+1}^\beta}\}$
      $C_{2,3}^\beta \equiv \{c_i^\beta : L_2^{a_l^\beta, a_{l+1}^\beta} \cap L_3^{a_p^\beta, a_{p+1}^\beta}\}$
    end for
  end for
end for
$C^\beta \equiv C_{1,2}^\beta \cup C_{1,3}^\beta \cup C_{2,3}^\beta$
for $j = 1$ to $|A_1^\alpha| - 1$ **do**
  for $l = 1$ to $|A_2^\alpha| - 1$ **do**
    for $p = 1$ to $|A_3^\alpha| - 1$ **do**
      $C_{1,2}^\alpha \equiv \{c_i^\alpha : L_1^{a_j^\alpha, a_{j+1}^\alpha} \cap L_2^{a_l^\alpha, a_{l+1}^\alpha}\}$
      $C_{1,3}^\alpha \equiv \{c_i^\alpha : L_1^{a_j^\alpha, a_{j+1}^\alpha} \cap L_3^{a_p^\alpha, a_{p+1}^\alpha}\}$
      $C_{2,3}^\alpha \equiv \{c_i^\alpha : L_2^{a_l^\alpha, a_{l+1}^\alpha} \cap L_3^{a_p^\alpha, a_{p+1}^\alpha}\}$
    end for
  end for
end for
$C^\alpha \equiv C_{1,2}^\alpha \cup C_{1,3}^\alpha \cup C_{2,3}^\alpha$
if $C^\beta \not\equiv \varnothing$ & $C^\alpha \not\equiv \varnothing$ **then**
  $\exists \min D(c_n^\beta, c_m^\alpha)$
  $e(k) = \{(y, z) : (c_n^\beta + c_m^\alpha)/2\}$
else if $C^\beta \not\equiv \varnothing$ & $C^\alpha \equiv \varnothing$ **then**
  if $|C^\beta| = 1$ **then**
    $e(k) = \gamma$
  else
    $\exists \min D(c_n^\beta, c_m^\beta) \ : \ D(c_n^\beta, c_m^\beta) < \epsilon$
    $e(k) = \{(y, z) : (c_n^\beta + c_m^\beta)/2\}$
  end if
else if $C^\beta \equiv \varnothing$ & $C^\alpha \not\equiv \varnothing$ **then**
  if $|C^\alpha| = 1$ **then**
    $e(k) = \gamma$
  else
    $\exists \min D(c_n^\alpha, c_m^\alpha) \ : \ D(c_n^\alpha, c_m^\alpha) < \epsilon$
    $e(k) = \{(y, z) : (c_n^\alpha + c_m^\alpha)/2\}$
  end if
else if $C^\beta \equiv \varnothing$ & $C^\alpha \equiv \varnothing$ **then**
  $e(k) = \gamma$
end if

4.2 Performance Metrics for Evaluating Position Estimation Algorithms

This section presents two metrics defined to quantify the accuracy and reliability of each of the 16 algorithms. The metrics are used to evaluate the performance of each algorithm and to make comparison among the various algorithms. The metrics are also utilized to quantify the sensitivity of the performance of the algorithms against the strength of the atmospheric turbulence and the variances of the sensor measurement noises. The metric quantifying the accuracy of the estimated position is referred to as "Cumulative Error (CE)" and that for the reliability as "Estimation Ratio (ER)". Cumulative error (CE) is the sum of the estimated position error for all locations in the grid, and formulated as

$$CE = \sum_{i=1}^{M} e_i \tag{4.1}$$

where e_i is position error at i^{th} point of the grid, and M is the number of points at which the estimation algorithm can generate an estimate. That is, the grid points at which the algorithm fails to estimate a position are excluded. The estimation ratio (ER) is the ratio of the number of grid points (M) at which the algorithm generates an estimate to the total number of points in the grid (N).

$$ER = \frac{M}{N} \tag{4.2}$$

In the above definitions of the metrics, all the points in the grid are considered equally important. In the case of aerial refueling, the most important region of the grid is at the contact position. As the position is further away from the contact position, more error in position estimation could be tolerated. Based on the concept, "weighted" versions of the metrics are defined. The weighting function is defined, as depicted

in Fig. 4.3, as to reduce the importance of the points as they are located at longer distance from the contact position. The maximum value of the weighting function is 1, at the point corresponding to the contact position, and decreases linearly to 0.1 at the edge of the grid.

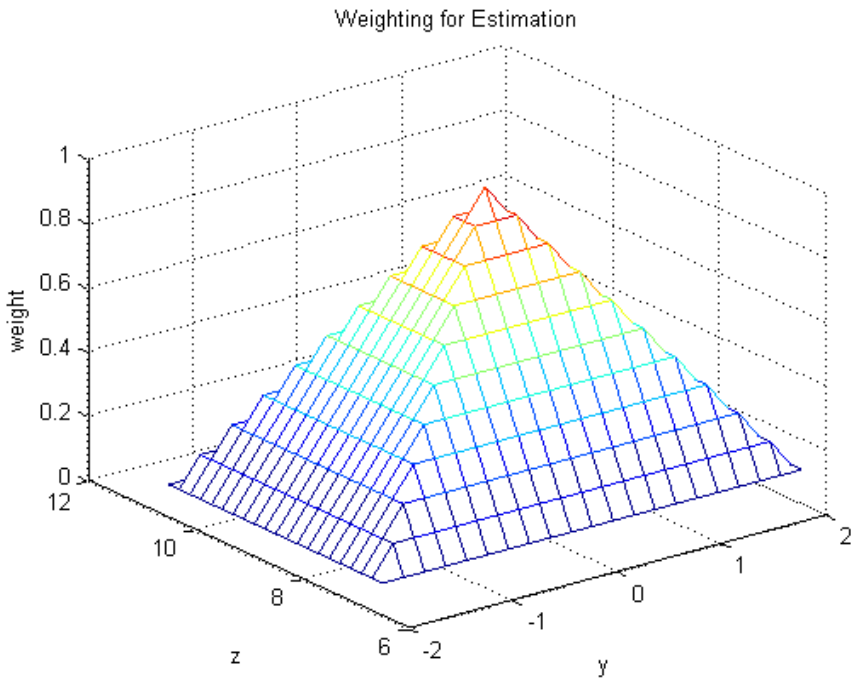


Figure 4.3. Performance Index Weighting.

Using the weighting function, the weighted Cumulative Error (WCE) is defined as

$$WCE = \sum_{i=1}^{M} w_i \, e_i \tag{4.3}$$

where w_i is the value of the weighting function at the i^{th} point of the grid. Recall that M points used in WCE calculations are the ones at which the algorithm can produce an estimated position. Similarly, weighted Estimation Ratio (WER) is formulated as

$$WER = \frac{\sum_{i=1}^{M} w_i}{N} \tag{4.4}$$

where the total number of points in the grid (N) in the denominator while the numerator sums up the weighting function only over the points at which the algorithm can produce a solution. Note that, when weighting function is 1 everywhere, Eq. (4.2) is recovered from Eq. (4.4).

CHAPTER 5

AIRDATA SENSOR FAULT DETECTION AND ISOLATION

This chapter presents the application of a Fault Detection and Isolation (FDI) algorithm to the airdata sensors of a receiver aircraft in aerial refueling flight. FDI algorithms are used to identify any occurrence of a fault in the system (detection) and to pinpoint the type and location of the fault (isolation). While there are various types of FDI development methods, the FDI presented here is based on sensor redundancy. Such an FDI system processes the measurements from a group of the same type of sensors in order to detect and isolate any fault in any one of the sensors in the group. The receiver aircraft is assumed to be equipped with same type of airdata sensors, placed at different locations. The airdata sensors are to measure the airspeed, angle-of-attack and side-slip angle of aircraft.

5.1 Sensor Fault Detection and Isolation using Statistical Approach

In order to detect faults in the redundant airdata sensor system, a parity space approach with chi-square test hypothesis is used [75, 76]. In a parity space based fault detection, the idea is to create residuals that represent the difference between actual system and the model outputs [50]. The procedure for developing the residual vector starts from the measurement equation for three sensors

$$\begin{bmatrix} y_{m_1}(k) \\ y_{m_2}(k) \\ y_{m_3}(k) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} y(k) + \begin{bmatrix} \xi_1(k) \\ \xi_2(k) \\ \xi_3(k) \end{bmatrix} + \begin{bmatrix} f_1(k) \\ f_2(k) \\ f_3(k) \end{bmatrix} \tag{5.1}$$

where $y_{m_i}(k)$ is the measured value by the i^{th} sensor of state variable $y(k)$, $\xi_i(k)$ is the measurement noise and $f_i(k)$ is the fault value for the i^{th} sensor. The measurement noises $\xi_i(k)$, $i = \{1, 2, 3\}$, are assumed to be normally distributed random sequences with zero mean and standard devaiation of σ and independent from each other. The measurement equation in Eq. (5.1) can be written in matrix form as

$$\mathrm{Y}(k) = \mathrm{H}y(k) + \Xi(k) + \mathrm{F}(k) \tag{5.2}$$

where $\mathrm{Y} = [y_{m_1}\ y_{m_2}\ y_{m_3}]^T$, $\mathrm{H} = [1\ 1\ 1]^T$, $\Xi = [\xi_1\ \xi_2\ \xi_3]^T$ with covariance matrix $\mathbf{R} = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2) \in \mathbb{R}^{3\times3}$ and $\mathrm{F} = [f_1\ f_2\ f_3]^T$.

To eliminate the state variable from Eq. (5.2), the left null space of the vector H is introduced to the equation. A left null vector of H, $v \in \mathbb{R}^{1\times3}$, is a row vector such that $v\mathrm{H}^T = 0$. The matrix $\mathbf{V} \in \mathbb{R}^{2\times3}$ is defined to represent the left null space of H and can be constructed by the two independent null vectors of H such that

$$\mathbf{V} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{5.3}$$

Premultiplying the measurement equation in Eq. (5.2) by $\mathbf{V}$ eliminates the state variable $y(k)$ and leads to a two-dimensional vector as

$$\mathrm{T}(k) = \mathbf{V}\mathrm{Y}(k) = \mathbf{V}\Xi(k) + \mathbf{V}\mathrm{F}(k) \tag{5.4}$$

where $\mathrm{T}(k) = [\tau_1(k)\ \tau_2(k)]^T \in \mathbb{R}^{2\times1}$ is defined as the residuals, which are to indicate any inconsistency in the measurement signals, $y_{m_i}(k)$, $i = \{1, 2, 3\}$.

In the case of no faulty sensor, $\mathrm{F}(k) = 0$, Eq. (5.4) implies that the residuals are simply the measurement noises multiplied by the null space matrix as

$$\mathrm{T}(k) = \mathbf{V}\Xi(k) \tag{5.5}$$

where note that the expected value of the residuals are zero because $\mathbf{V}$ is deterministic constant and the measurement noise vector $\Xi(k)$ has zero means. In this case, the covariance of the residual vector is calculated as

$$E[\mathrm{T}(k)\mathrm{T}^T(k)] = E[\mathbf{V}\Xi(k)\Xi^T(k)\mathbf{V}^T] = \mathbf{\Gamma}^T\mathbf{\Gamma} \tag{5.6}$$

where a new matrix $\mathbf{\Gamma} \in \mathbb{R}^{2\times 2}$ is introduced such that

$$\mathbf{V}\mathbf{R}\mathbf{V}^T = \mathbf{\Gamma}^T\mathbf{\Gamma} \tag{5.7}$$

This new matrix $\mathbf{\Gamma}$ is used to normalize the residuals as

$$\bar{\mathrm{T}}(k) = \mathbf{\Gamma}^{-T}\mathrm{T}(k) \tag{5.8}$$

such that the covariance of the normalized residuals are

$$E[\bar{\mathrm{T}}(k)\bar{\mathrm{T}}^T(k)] = \mathbf{\Gamma}^{-T}E[\mathrm{T}(k)\mathrm{T}^T(k)]\mathbf{\Gamma}^{-1} \tag{5.9}$$

which, using Eq. (5.6), leads to

$$E[\bar{\mathrm{T}}(k)\bar{\mathrm{T}}^T(k)] = \mathbf{\Gamma}^{-T}\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{\Gamma}^{-1} = \mathbf{I}_{2\times 2} \tag{5.10}$$

which is a 2×2 identity matrix. That is, the normalized residuals $\bar{\tau}_1(k)$ and $\bar{\tau}_2(k)$ have unity variance. Note also that the expected values of $\bar{\mathrm{T}}(k)$ are zero since $E[\mathrm{T}(k)] = 0$.

In the case of a faulty sensor, by Eq. (5.4), the expected value of the residuals are

$$E[\mathrm{T}(k)] = \mathbf{V}\mathrm{F}(k) \neq 0 \tag{5.11}$$

which implies

$$E[\bar{\mathrm{T}}(k)] = \mathbf{\Gamma}^{-T}\mathbf{V}\mathrm{F}(k) \neq 0 \tag{5.12}$$

This observation is used to set the hypothesis test for detecting sensor fault as

$$\begin{aligned} H_0 &: E[\bar{\mathrm{T}}(k)] = 0 \\ H_1 &: E[\bar{\mathrm{T}}(k)] \neq 0 \end{aligned} \tag{5.13}$$

The acceptance of the hypothesis is carried out by the second degree chi-square test as

$$\chi^2(2, k) = \gamma(k) = \bar{\tau}_1^2(k) + \bar{\tau}_2^2(k) \tag{5.14}$$

The chi-square test is based on a look-up table that gives the threshold value h for a given false alarm probability α

$$Pr(\gamma(k) \leq h) = 1 - \alpha \tag{5.15}$$

In order to detect faults, the χ^2 random variable $\gamma(k)$ is compared to h. If $\gamma(k)$ is equal or smaller than the given value, the test results in the acceptance of H_0. On the other hand, if $\gamma(k)$ is greater than h, the test results in the acceptance of H_1. It should be noted that this evaluation includes a false alarm probability of α.

Once a fault is detected, the isolation is accomplished by the angle between the residual vector and the columns of the null space. It should be noted that, in this study, a fault is assumed to occur in one sensor only at a time. In parity space approach, inconsistency in one of the sensors results in orthogonal projection of the fault vector on the null space [77]. In the case of a fault in one of the sensors, $\mathrm{F}(k)$ is non-zero and the residual equation in Eq.(5.4) becomes

$$\mathrm{T}(k) = \mathbf{V}\Xi(k) + \mathbf{V}\mathrm{F}(k) \tag{5.16}$$

which can be rewritten in terms of matrix $\mathbf{V}$, $\mathrm{V}_i \in \mathbb{R}^{2\times 1}$, $i = \{1, 2, 3\}$ as

$$\mathrm{T}(k) = \mathbf{V}\Xi(k) + \mathrm{V}_1 f_1(k) + \mathrm{V}_2 f_2(k) + \mathrm{V}_3 f_3(k) \tag{5.17}$$

which indicates that the expected value of the residual vector $\mathrm{T}(k)$ is shifted in the direction of V_i when there is a fault in the i^{th} sensor, i.e., $f_i(k) \neq 0$. Thus, simply by checking the angle between normalized residual vector and the column vectors of matrix $\mathbf{V}$, the sensor with the fault can be determined. For this purpose, the angle difference index ϑ is introduced as

$$\vartheta_i(k) = \left\| \frac{\bar{\mathrm{T}}(k)}{\|\bar{\mathrm{T}}(k)\|} - \frac{\mu_i}{\|\mu_i\|} \right\|, \; i = \{1, 2, 3\} \tag{5.18}$$

where $\mu_i = \mathbf{\Gamma}^{-T}\mathrm{V}_i$ is the normalized i^{th} column vector of the null space matrix $\mathbf{V}$. Note in Eq. (5.18) that $\bar{\mathrm{T}}(k)/\|\bar{\mathrm{T}}(k)\|$ is the unit vector along the direction of $\mathrm{T}(k)$ and $\mu_i/\|\mu_i\|$ is the unit vector along the i^{th} null space vector. The angles of these

unit vectors are denoted as $\theta(k)$ and ϕ_i, respectively. Thus, the unit vector can be written in terms of their angles as

$$\frac{\bar{\mathrm{T}}(k)}{\|\bar{\mathrm{T}}(k)\|} = \begin{bmatrix} \cos\theta(k) \\ \sin\theta(k) \end{bmatrix} \text{ and } \frac{\mu_i}{\|\mu_i\|} = \begin{bmatrix} \cos\phi_i \\ \sin\phi_i \end{bmatrix} \tag{5.19}$$

The angle difference index can be rewritten in terms of these angles as

$$\vartheta_i(k) = \sqrt{(\cos\theta(k) - \cos\phi_i)^2 + (\sin\theta(k) - \sin\phi_i)^2} \tag{5.20}$$

which, using the trigonometric relations after carrying out the squares, becomes

$$\vartheta_i(k) = \sqrt{2 - 2\cos(\theta(k) - \phi_i)}\ ,\ i = \{1, 2, 3\} \tag{5.21}$$

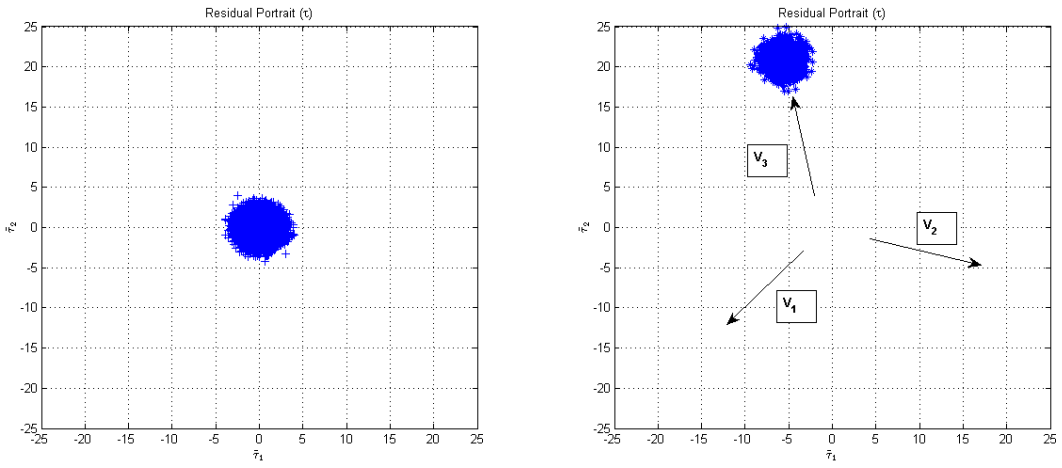
Note that maximum angle difference, $\theta(k) - \phi_i$, can be π, in which case $\vartheta_i(k) = 2$, and the minimum angle difference is 0 when $\theta(k) = \phi_i$, in which case $\vartheta_i(k) = 0$. Thus, $\vartheta_i(k) \in (0, 2)$. Among $\vartheta_1(k)$, $\vartheta_2(k)$ and $\vartheta_3(k)$ calculated by Eq. (5.21), the smallest $\vartheta_i(k)$ is

$$\vartheta_k(k) = \min\{\vartheta_i(k), i = 1, 2, 3\} \tag{5.22}$$

which will indicate the k^{th} sensor as the faulty one because the residuals will be shifted in the direction of the k^{th} column of the null space matrix.

To visualize the detection and isolation processes described above, the residuals can be plotted with respect to $\bar{\tau}_1(k)$ and $\bar{\tau}_2(k)$ as shown in Fig. 5.1. In the case of no fault, normalized residuals appear at the origin in the residual portrait plot (Fig. 5.1(a)). From statistical point of view, if there is no fault in the system, normalized residuals should be inside a disc centered at the origin. The size of this disc is defined by the threshold h, and residuals are expected to fall outside only by the false alarm

probability α. In the presence of a fault in one of the sensors, the normalized residuals are not only shifted from the origin and located outside the disc much more than false alarm probability, they are shifted in the direction of the corresponding null space column vector (Fig. 5.1(b)). In Fig. 5.1(b), the vector directions of columns of the null space V_i are shown. For example, when there exists a fault in sensor 3, the normalized residuals will move to the new location as shown in Fig. 5.1(b). By checking the minimum angle difference between normalized residuals vector and the column vectors leads to the isolation of the faulty sensor. Further details of this approach can be found in Ref. [78].



(a) (b)

Figure 5.1. Residual Portrait of Side Slip Angle Data (a) Typical Example of the Residuals Plot in the Case of No Fault (b) In the Case of Faulty Sensor 3, Residuals are Shifted in the Vector Direction of the Corresponding Column of the Null Space.

5.2 Robust Fault Detection and Isolation using Relative Position Information

The proposed measurement equation for Fault Detection and Isolation algorithm in Section 5.1 is based on the assumption that the sensor readings are identical.

This assumption is presented as the vector H being equal to $[1\ 1\ 1]^T$. Consequently, the left null space $\mathbf{V}$ of the vector H is calculated once and it is constant. In aerial refueling operations, the receiver aircraft is exposed to nonuniform wind field induced by tanker aircraft, and this nonuniform wind field leads to differences in sensor readings. In other words, due to the nonuniform wind field, different sensors placed at different location on the aircraft give different airdata measurements. Thus, the identical readings assumption for FDI algorithm is not valid when aircraft flies in a nonuniform wind field. To overcome this problem, the expected values of each sensor reading are used instead of the vector H, and the null space is calculated accordingly based on the expected values.

Recall that Section 3.3 gives the maps of the nonuniform wind field in terms of airspeed, side slip angle and angle of attack as functions of the relative position with respect to the tanker aircraft. These maps are used to compute the expected values of the airdata variables at each sensor given the relative position of the receiver aircraft. This approach is formulates as follows. At time k, given the position of the aircraft, especially $y(k)$ and $z(k)$, the expected values of airspeed (V), sideslip angle (β) and angle-of-attack (α) are obtained from the maps of the nonuniform wind field for the i^{th} sensor as

$$\begin{aligned}
\hat{V}_i(k) &= U_i^V(y(k), z(k)) \\
\hat{\beta}_i(k) &= U_i^\beta(y(k), z(k)) \\
\hat{\alpha}_i(k) &= U_i^\alpha(y(k), z(k))
\end{aligned} \tag{5.23}$$

where U_i^V, U_i^β, U_i^α are the the maps (used as lookup tables) and V, β and α are expected airdata measurements for i^{th} sensor, and (y, z) are the relative position. Using expected values, the measurement equation (Eq. (5.1)) becomes

$$\begin{bmatrix} y_{m_1}(k) \\ y_{m_2}(k) \\ y_{m_3}(k) \end{bmatrix} = \begin{bmatrix} a(r) \\ b(r) \\ c(r) \end{bmatrix} y(k) + \begin{bmatrix} \xi_1(k) \\ \xi_2(k) \\ \xi_3(k) \end{bmatrix} + \begin{bmatrix} f_1(k) \\ f_2(k) \\ f_3(k) \end{bmatrix} \tag{5.24}$$

where $a(r), b(r), c(r)$ are expected values obtained from nonuniform wind map, $y_{m_i}(k)$ is the measured value by the i^{th} sensor of state variable $y(k)$. Similar to in the Eq. (5.1), $\xi_i(k)$ and $f_i(k)$ are the normally distributed measurement noise with zero mean and covariance $\mathbf{R}$ and the corresponding fault values, respectively.

To eliminate the first vector from Eq. (5.24), the left null space is employed. For Eq. (5.1), the null space matrix $\mathbf{V}$ was constant and thus calculated once. Considering that the expected values of airdata variables vary with relative position in Eq. (5.24), the left null space for this equation should be recalculated for varying relative position. After pre multiplying both sides of Eq. (5.24) with the computed null space $\mathbf{V}_r(k)$, the measurement equation becomes

$$\mathrm{T}_r(k) = \mathbf{V}_r(k)\mathrm{Y}(k) = \mathbf{V}_r(k)\Xi(k) + \mathbf{V}_r(k)\mathrm{F}(k) \tag{5.25}$$

where $\mathrm{T}_r(k)$ is two dimensional residual vector with elements $[\tau_{r_1}(k)\ \tau_{r_2}(k)]^T$. The subscript r means robust which indicates that null space values ($\mathbf{V}_r(k)$) are varying in each sample step (k). In the no-fault case, the measured values are simply equal

to measurement noise multiplied by the null space matrix. Thus, the covariance of the robust residuals in no fault case can be found as

$$E[\mathrm{T}_r(k)\mathrm{T}_r^T(k)] = \mathbf{\Gamma}_r^T\mathbf{\Gamma}_r \tag{5.26}$$

The normalized residuals can be found in terms of variable $\mathbf{\Gamma}_r$ as

$$\bar{\mathrm{T}}_r(k) = \mathbf{\Gamma}^{-T}\mathrm{T}_r(k) \tag{5.27}$$

The covariance of the robust normalized residuals can be found by combining the Eq. (5.26) and (5.27) in no fault case as

$$E[\bar{\mathrm{T}}_r(k)\bar{\mathrm{T}}_r^T(k)] = \mathbf{\Gamma}_r^{-T}E[\mathrm{T}_r(k)\mathrm{T}_r^T(k)]\mathbf{\Gamma}_r^{-1} = \mathbf{I}_{2\times 2} \tag{5.28}$$

which shows, as in Eq. (5.10), that the normalized residuals have unity variance.

Once the residuals are calculated, the remaining steps of the detection and isolation procedure in this case are the same as the ones in Section 5.1.

5.3 Performance Metrics for Evaluating FDI Algorithms

In order to quantify the statistical test of the FDI, the "Alarm Index" ($AI(k)$) is introduced as

$$AI(k) = \begin{cases} 1 & \text{if } \gamma_r(k) \leq h\text{; acceptance of } H_0 \\ 0 & \text{if } \gamma_r(k) > h\text{; acceptance of } H_1 \end{cases} \tag{5.29}$$

where $\gamma_r(k)$ is χ^2 random variable and h is the threshold corresponding to the false alarm probability α. $AI(k)$ in Eq. (5.29) simply gives 1 when the H_0 is accepted

which indicates of fault detection, and gives 0 when there is no fault detected. For the isolation, "Sensor Isolation Index" ($SII(k)$) is introduces as

$$SII(k) = \begin{cases} 0 & \text{if } AI(k) = 0 \\ 1 & \text{if } AI(k) = 1 \ \& \ \min\{\vartheta_{r_i}(k)\} = \vartheta_{r_1}(k) \\ 2 & \text{if } AI(k) = 1 \ \& \ \min\{\vartheta_{r_i}(k)\} = \vartheta_{r_2}(k) \\ 3 & \text{if } AI(k) = 1 \ \& \ \min\{\vartheta_{r_i}(k)\} = \vartheta_{r_3}(k) \end{cases} \tag{5.30}$$

which is 0 when there is no fault detected, and gives the number corresponding to the sensor with the minimum angle difference when the algorithm leads to an alarm. The performance of both FDI and robust FDI algorithms are quantified by two metrics called "True Alarm Percentage" (TAP) and "False Alarm Percentage"(FAP). FAP is the ratio of the number of data points with alarm cases to all data points while there is no fault injected in the system and is formulated as

$$FAP = 100 \cdot \frac{1}{N} \sum_{k=1}^{N} AI(k) \tag{5.31}$$

where N is the total number of data points. Similarly, TAP is expressed as

$$TAP = 100 \cdot \frac{1}{(M-P)+1} \sum_{k=P}^{M} AI(k) \tag{5.32}$$

where the fault is injected between the interval P and M. Also, if any fault is injected to the system, the Eq. (5.31) becomes

$$FAP = 100 \cdot \frac{1}{N} \frac{1}{(M-P)+1} \left\{ \sum_{k=1}^{N} AI(k) \ - \ \sum_{k=P}^{M} AI(k) \right\} \tag{5.33}$$

It should be noted that, if there is no fault injected to the system, Eq. (5.33) is equivalent to Eq. (5.31). The *AI*, *SII*, *TAP* and *FAP* can also be calculated for the FDI algorithm presented in Section 5.1, only the variables without subscript r should be used in the equations.

The robust detection and isolation processes stated above can be summarized as follows. In the case of no fault, the normalized residuals should appear inside a disc centered at the origin, and the AI and SII will be zero. In the presence of a fault at one of the sensors, the normalized residuals are shifted in the direction of the corresponding null space column vector. For example, when there exists a fault in sensor 3, the normalized residuals will be shifted in the direction of the corresponding null space column vector and this results in $AI = 1$. By checking the minimum angle leads to an isolation of the faulty sensor, and it will indicate that the angle between the residual vector and $\mathbf{V}_{r_3}$ is the minimum, and consequently SII will be equal to 3.

5.4 Sensor Measurement Correction

Airdata sensor measurements give the information of the aircraft velocity vector with respect to the local air. If the translational velocities of the sensors relative to the air are different, the sensors will give different measurements of airspeed, sideslip angle and angle-of-attack even though they are on the same aircraft. There are mainly two mechanisms that can cause different translational velocity relative to the air. (1) The air has different translational velocities relative to the inertial frame. (2) The positions of the sensor have different translational velocities relative to an inertial frame. An example of the former is the nonuniform wind distribution previously mentioned in the case of aerial refueling. The latter occurs when a rigid aircraft has rotational motion in addition to the translational motion. Given the angular

velocity measurement of the aircraft and the position of the sensors, the contribution of the rotational motion on the airdata measurements can be isolated. With this correction, all the airdata sensors regardless of their positions in the aircraft give the translational velocity of the aircraft relative to the surrounding air. Then, any occurrence of difference in the airdata measurements can be attributed to either (1) difference in the wind velocity at the sensor positions or (2) fault in a sensor. Airdata sensor measurements, airspeed, sideslip angle and angle-of-attack, are to parameterize the translational velocity of the aircraft relative to the air. This translational velocity vector can also be written in terms of its components in the body frame of the aircraft. Then, the components, *u, v, w* and the sensor readings are related as

$$\begin{aligned} u_m &= V_m \cos\beta_m \cos\alpha_m \\ v_m &= V_m \sin\beta_m \\ w_m &= V_m \cos\beta_m \sin\alpha_m \end{aligned} \tag{5.34}$$

where the subscript m indicates that values are measured values. The airdata sensors are to measure the airspeed, angle-of-attack and sideslip angle of aircraft, thus from the sensors, V_m, β_m and α_m values are read. By Eq. (5.34), (u_m, v_m, w_m) are the components of the translational velocity vector of an airdata sensor with respect to air, which includes the effect of the rotational motion of the aircraft. The components of the translational velocity due to only the rotational motion of the aircraft are

$$\begin{bmatrix} u_s \\ v_s \\ \omega_s \end{bmatrix} = -\mathbf{S}(\omega_{\mathbf{B_R}})\rho_{pt} \tag{5.35}$$

where $\mathbf{S}(\omega_{\mathbf{B_R}})$ is skew-symmetric matrix of angular velocity of the receiver relative to inertial frame, and ρ_{pt} is the representation of pitot-tube location in receiver's body frame. The skew-symmetric matrix is constructed from the angular velocity components of the receiver with respect to the inertial frame ($p_{\mathbf{R}}, q_{\mathbf{R}}$ and $r_{\mathbf{R}}$). The components of the translational velocity of an airdata sensor relative to the local air without the effect of the rotational motion of the aircraft can be determined as

$$\begin{aligned} u_c &= u_m - u_s \\ v_c &= v_m - v_s \\ w_c &= w_m - w_s \end{aligned} \tag{5.36}$$

which are used to calculate the corrected airspeed, sideslip angle, angle of attack (V_c, β_c, α_c) that will be fed into the FDI system as

$$\begin{aligned} V_c &= \sqrt{u_c^2 + v_c^2 + w_c^2} \\ \beta_c &= \arcsin \frac{v_c}{V_c} \\ \alpha_c &= \arctan \frac{w_c}{u_c} \end{aligned} \tag{5.37}$$

CHAPTER 6

SIMULATION RESULTS AND DISCUSSIONS

The simulations are performed in an integrated aerial refueling simulation environment. The simulation consists of both tanker and receiver aircraft models, their respective controllers, vortex-induced wind model and turbulence model. The simulation also models the effect of the vortex induced wind on the dynamics of the receiver aircraft. The aircraft models are based on the full 6-DOF nonlinear equations of motions including the wind effects, as presented in Chapter 2. The tanker aircraft under the influence of turbulence is flown by its controller with constant airspeed and at constant altitude in a straight level flight. The trajectory tracking controller of the receiver flies the aircraft relative to the tanker aircraft. In the wake of the tanker aircraft, the receiver flies in a nonuniform wind field in addition to turbulence and prevailing wind. The magnitude and direction of the vortex induced wind field change with the position relative to the tanker, as detailed in Section 2.3.3. The tanker aircraft model represents a KC-135R and the receiver model is a representation of a Learjet 25 in an aerial refueling test flight condition [1].

6.1 Relative Position Estimation Simulation Results

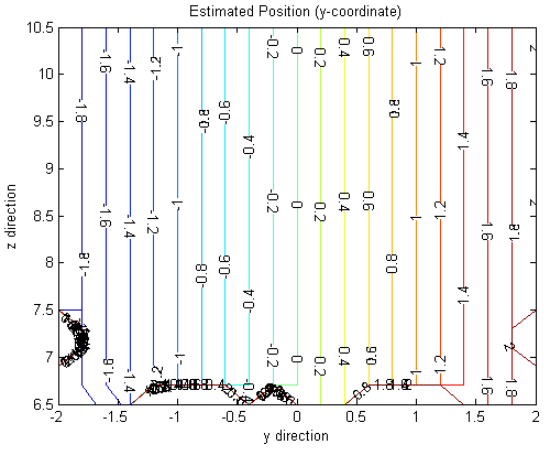
This section presents the simulation results of relative position estimation algorithms and the results of a parameter analysis. The position estimation algorithms are performed in both static and dynamic cases. The static simulation term indicates that the receiver and tanker aircraft are placed statically and their positions and orientations are fixed. In dynamic simulation, the full dynamics of both aircraft

are simulated. The tanker aircraft is flown by its controller based on the commanded altitude and airspeed. The controller of the receiver flies the aircraft through the commanded trajectory or position relative to the tanker. For parameter analysis, the effects of turbulence intensity, the prevailing wind and measurement noise variance are taken into consideration and the influence of these effects on the position estimation algorithms are presented.

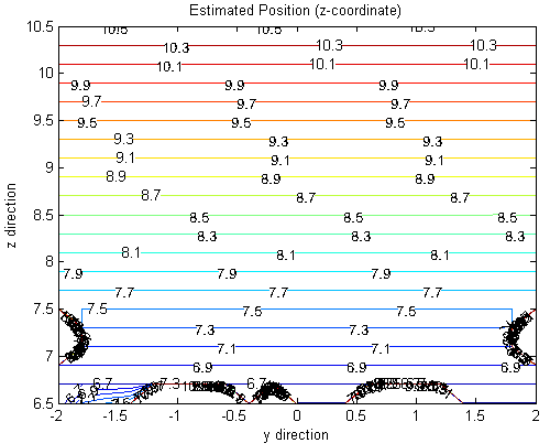
6.1.1 Static Simulation Results and Parameter Study

The simulations are first run with no turbulence or measurement error at each point of the grid and the instant airdata measurements are provided as inputs for the position estimation algorithm. In all simulation runs, the parameter epsilon (ϵ) is set to be 0.1 m. As an example, the estimated positions computed by Algorithm-11 for all locations are presented as contour plots on the grid in Fig. 6.1. In Fig. 6.1(a), the contour of the y-component of the estimated position, which is plotted with respect to actual position of the receiver aircraft, are shown. In Fig. 6.1(b), the result for z-component is given. As can be seen from the figures, the algorithm successfully estimates the position of the receiver aircraft at most of the points in the grid. At the bottom of the plots, the algorithm gives estimations with some error or no estimation due to the fact that the isolines in those areas do not have enough number of points for intersection calculation.

A comparison of the all algorithms in terms of the performance metrics is given in Fig. 6.2. The performance indices, ER and CE, are computed for each algorithm after the respective simulation. The results are presented in the ER-CE plane, as shown in Fig. 6.2. Each marker on the figure represents the performance of an algorithm, indicated by the number next to the marker. In the best case, ER and CE should both be zero, implying that the algorithm can estimate exact position



(a)



(b)

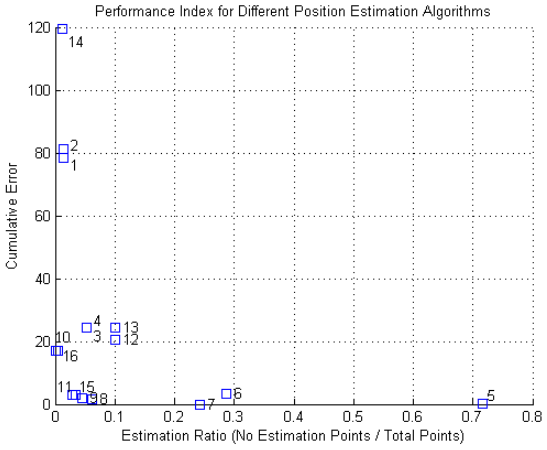
Figure 6.1. Estimated Position Results for all Locations in the Grid.

without any error at all the points of the grid. Such an algorithm would be placed at the origin of the ER-CE plane. Any algorithm placed at any other position indicates the degradation in its performance. The distance from the origin in ER direction means that the algorithm cannot produce any position estimation at some points of the grid; number of such points is proportional to the coordinate in the ER direction. Similarly, the coordinate in the CE direction indicates the cumulative error in position estimation at all the points at which the algorithm can generate some estimation.

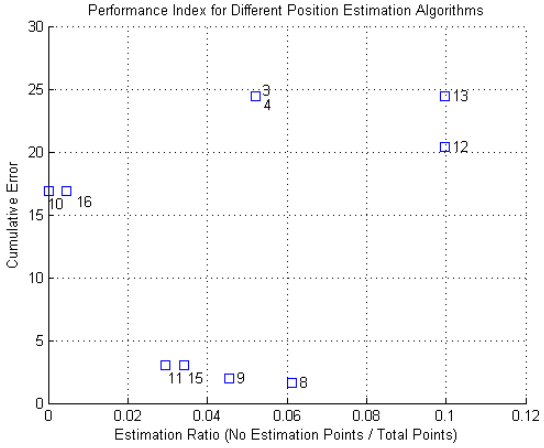
For example, an algorithm placed somewhere away from the origin on the ER axis means that there are points of the grid at which it cannot compute any estimation, but at the other points, it can generate perfect estimation. Similarly, an algorithm placed on the CE axis indicates that the algorithm generate some estimates at all points of the grid, but with some error in the estimated positions. As clearly seen in Fig. 6.2, the best two algorithms are 11 and 15 as they are the closest to the origin. Note that algorithm-9 and -8 results in slightly better performance in the CE direction however they are placed further in the ER direction (i.e., fail to compute some estimation at more points than algorithm-11 and -15 do) and, thus, perform worse overall. Therefore, for the further investigation, only algorithm-11 and -15 are considered.

Recall from Section 4.2 that weighted ER and CE are defined to put more emphasis at and around the contact position than those away from the contact position. Fig. 6.3 shows the performance comparison of the algorithms in terms of WER and WCE.

As the next step, the turbulence effect on the performance of the position estimation is investigated. Different turbulence intensities are set in the simulation. The simulations are run for all points and for each point the airdata measurements are given as input to the position estimation system. Due to the fact that turbulence causes fluctuations in the airdata measurements, instead of instant airdata measurements, the average values of last 1 second of readings are given as an input to the algorithm. In these simulations, the parameter epsilon (ϵ) is set as 0.1. The resulting plots are given in Fig. 6.4. In the figure, the square marker indicates the results from Algorithm-11, the star marker is used for the results of Algorithm-15, and the corresponding turbulence level is indicated with a text next to each marker. The results show that the increase of the turbulence intensity degrades the performance



(a)



(b)

Figure 6.2. (a) Performance Index for Different Position Estimation Algorithms (b) Detail View of Part a.

of the position estimation. Both estimation ratio and cumulative error are increased as the turbulence intensity increases, and thus, the sensitivity of the algorithms to turbulence is quantified. Figure 6.4 also indicates the difference in sensitivity between the two algorithms. Recall that in Algorithm-11, in the case of one intersection point occurrence from one airdata variable, that point is taken as the estimated position; however Algorithm-15 in the same case discards that value. As a result, the per-

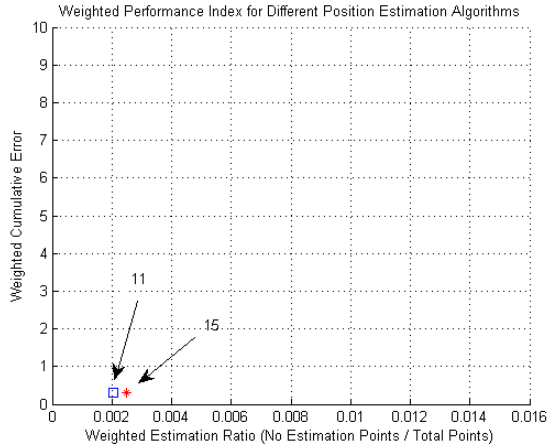


Figure 6.3. Weighted Performance Index for Different Position Estimation Algorithms.

formance index results of Algorithm-11 are higher in cumulative error and lower in estimation ratio compared to those of Algorithm-15 for the same turbulence intensity.

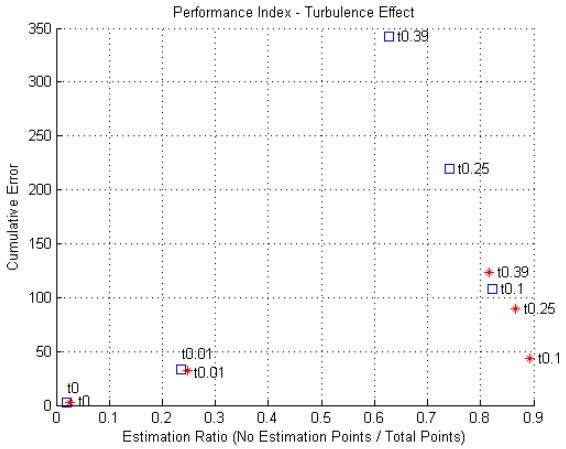


Figure 6.4. Performance Index for Turbulence Effect. The Results from Algorithm-11 are Shown with Square and the Results from Algorithm-15 are Shown with Star. Turbulence Levels are Shown with a Text Next to the Markers.

The sensitivity to turbulence effect can be reduced if the turbulence effect in the measurements is filtered. This is attempted in this research by increasing the length

of the Moving Average Window (MAW) although a larger window size is expected to introduce larger time lags in the mean value calculation relative to the actual signal. Fig. 6.5 shows the performances of the algorithms-11 and 15 with different MAW sizes, 1 sec to 25 sec, when the turbulence intensity is 0.39 m/s. Both algorithms initially show the same level of sensitivity when MAW is increased to 10. As the length of mean value calculation increases, the error in estimation decreases but the estimation ratio increases. Even larger MAW size then leads to decrease of estimation ratio, thus results of the algorithms approach to the origin in ER-CE plane.

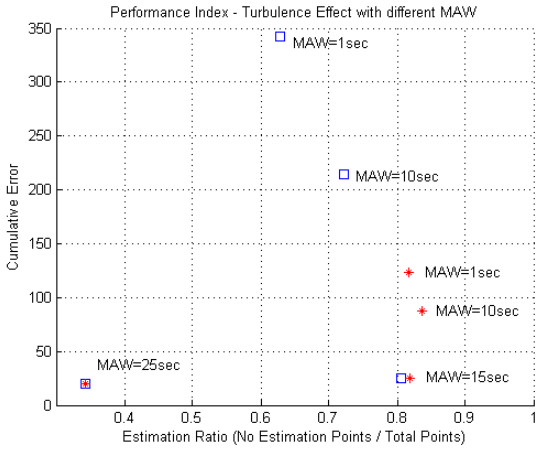


Figure 6.5. Performance Index for Turbulence Effect with Different MAW. The Results from Algorithm-11 are Shown with Square and the Results from Algorithm-15 are Shown with Star. MAW Values are Shown with a Text Next to the Markers.

The sensitivity of the algorithms to the measurement noise is also investigated. Fig. 6.6 shows the performances of the algorithms with 1 second MAW size and four different values of noise variance. In the figure, the square markers refer to the results from Algorithm-11, the star markers to Algorithm-15. The noise variance for each case is indicated with a text next to each marker. Both algorithms initially show the same level of sensitivity when variance is increased to 0.0002 as the corresponding

markers move to almost same position in ER-CE plane. However, any further increase in the noise variance affects Algorithm-11 more in CE direction.

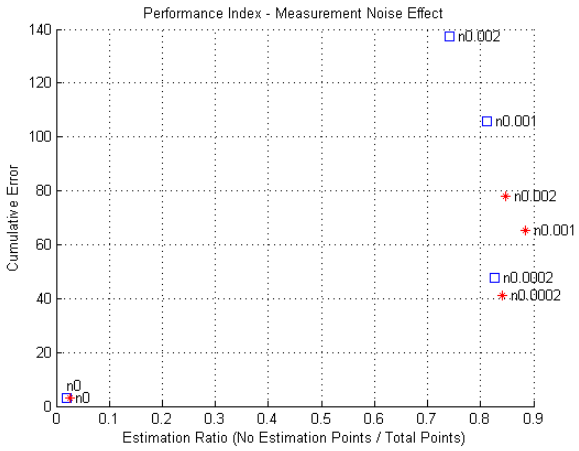


Figure 6.6. Performance Index for Measurement Noise Effect. The Results from Algorithm-11 are Shown with Square and the Results from Algorithm-15 are Shown with Star. Noise Variances are Shown with a Text Next to the Markers.

Similar to the turbulence effect, the sensitivity to measurement noise effect can also be reduced if the it is filtered. Fig. 6.7 shows the performances of the algorithms-11 and 15 with two different MAW sizes, 1 sec to 25 sec when the measurement noise variance is 0.002 $(deg)^2$. Same as in the turbulence effect analysis, both algorithms produce less error when MAW is increased to 10 but the estimation ratio increases. As the MAW size increases further, estimation ratio is also decreased, thus results of the algorithms approach to the origin in ER-CE plane.

Similarly, prevailing wind effect on position estimation performances is analyzed. The simulations are run for all points in the grid and 1 second average values of the airdata measurements are given as an input to the position estimation system. The resulting plots for prevailing effect is presented in Fig. 6.8. In the figure, the case whether prevailing wind on or off is indicated with a text next to the marker. Similar

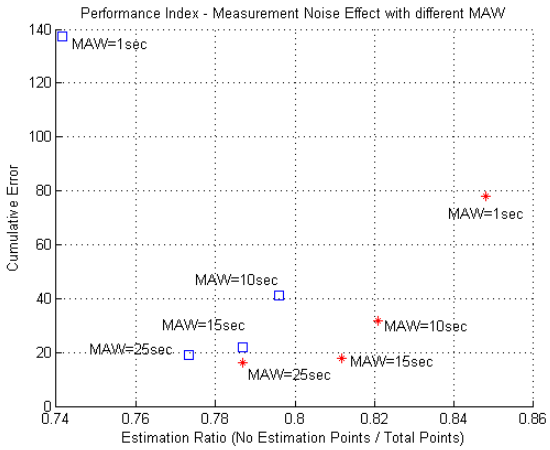


Figure 6.7. Performance Index for Measurement Noise Effect with Different MAW. The Results from Algorithm-11 are Shown with Square and the Results from Algorithm-15 are Shown with Star. MAW Values are Shown with a Text Next to the Markers.

to the results from turbulence and noise effect analysis, both estimation ratio and cumulative error increases with addition of prevailing wind. This is because prevailing wind causes a shift on wind field. As the values are shifted and become different that the values in the nonuniform wind map that the algorithms use, performance of the algorithms is degraded. The effect of the prevailing wind is different than those of turbulence and measurement noise. Thus, the larger MAW size will not make any difference on the performance of the algorithms.

In static simulations, further analysis are performed using the extended grid (Fig. 3.12). The resulting estimated positions for all locations are presented in Fig. 6.9. The plots show the results of the position estimation performed by using Algorithm-11.

In Fig. 6.9(a), the contour of the y-component of the estimated position are shown. In Fig. 6.9(b), the result for z-component is given. As depicted in the figure, the algorithm successfully estimates the position of the receiver aircraft for almost all

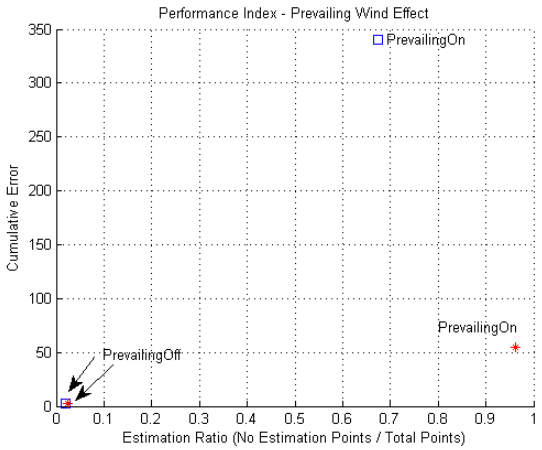
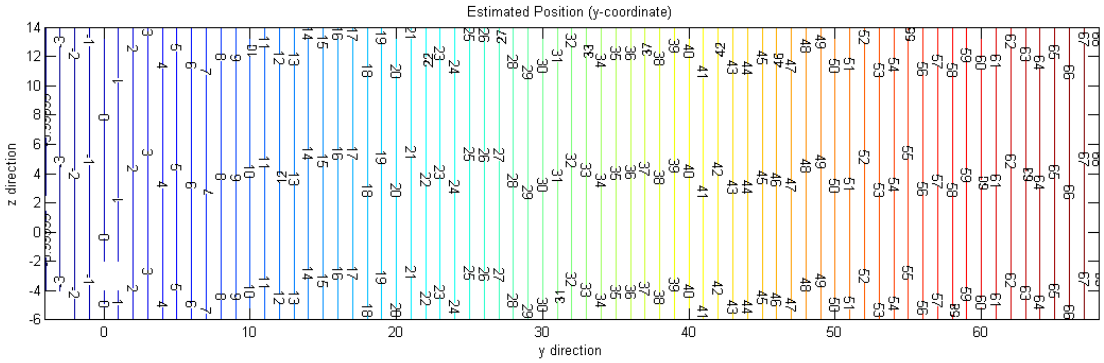


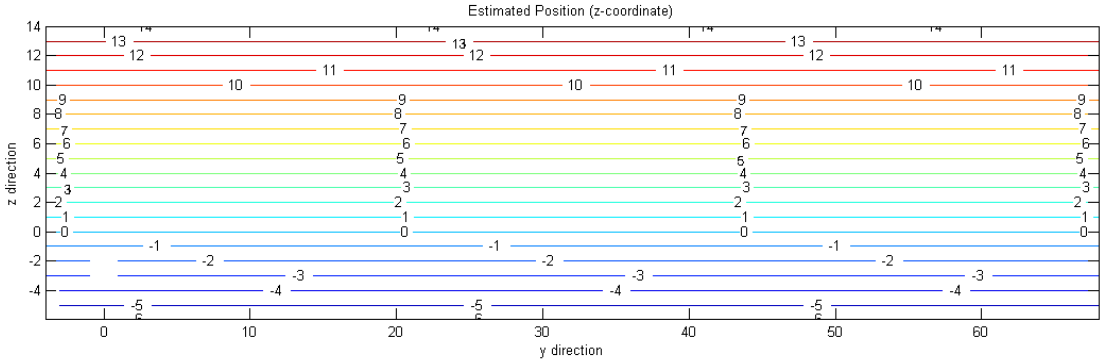
Figure 6.8. Performance Index for Prevailing Wind Effect. The Results from Algorithm-11 are Shown with Square and the Results from Algorithm-15 are Shown with Star. The Cases of Prevailing Wind are Shown with a Text Next to the Markers.

of the points in the grid. The algorithm generates zero error in position estimation for the points at which it gives an output, and at only 3 out of 1533 points the algorithm cannot produce an estimation.

The main reason to construct the extended grid is to use it in aerial refueling flight cases, in which the receiver maneuvers from the observation position to the contact position (Fig. 3.6). Recall that the observation position is at the right side of the tanker, and extended grid includes the observation position. In the next section, dynamic simulation results are presented. And in the dynamic cases, the relative position estimation performance of the algorithm is demonstrated for all points in the grid. Also, the time histories of the estimation are presented when the receiver is commanded to fly at certain points around the contact position, as well as when the receiver maneuvers from observation position to contact position.



(a)



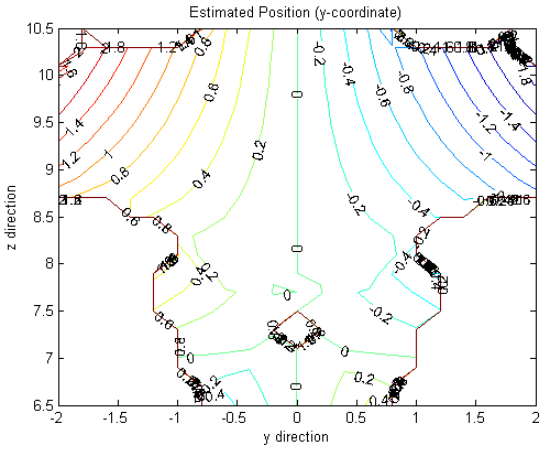
(b)

Figure 6.9. Estimated Position Results for all Locations in the Extended Grid.

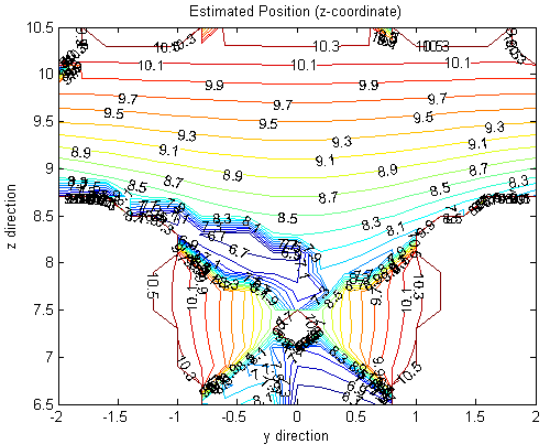
6.1.2 Dynamic Simulation Results

The dynamic simulations are run for all points in the grid defined around the contact position, and the resulting estimated positions for all locations are presented in Fig. 6.10. The plots show the results of the position estimation performed by using Algorithm-11. In these simulation runs, the parameter epsilon (ϵ) is taken as 0.1, prevailing wind and turbulence effects are turned off, and no measurement noise is added. In Fig. 6.10(a), the contour of the y-component of the estimated position, which is plotted with respect to actual position of the receiver aircraft, are shown. In Fig. 6.10(b), the result for z-component is given. Unlike the static case, the algorithm cannot estimate the position of the receiver aircraft for most of the points in the grid,

or it gives estimations with large errors. The main factor for that is the orientation difference of the receiver aircraft. In dynamic case, the controller of the receiver trims the orientations and if the trim values are different from that in static simulations used to generate the nonuniform wind maps, this results in the degradation of the relative position estimation performance. However, considering that the point of interest here is the contact position ($y = 0$ and $z = 8.5$), the algorithm still successfully estimates the position of the receiver aircraft.



(a)

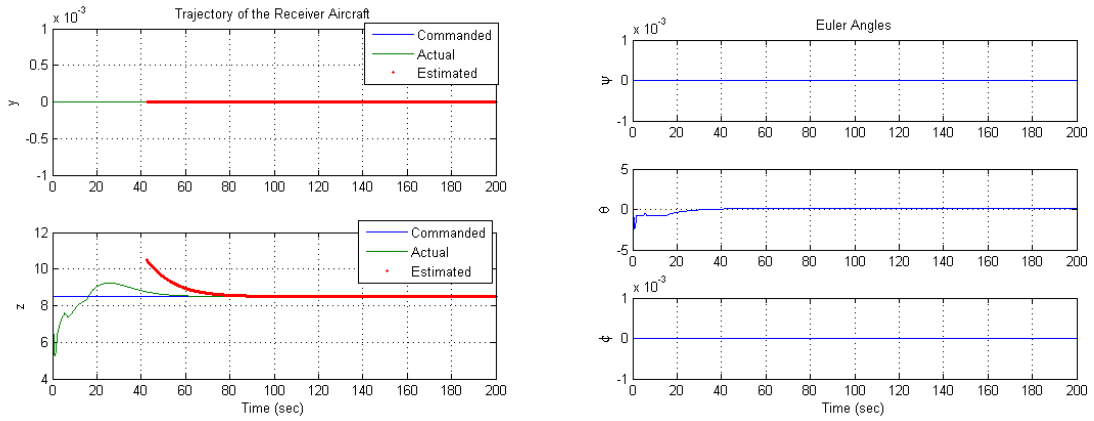


(b)

Figure 6.10. Estimated Position Results for all Locations in the Grid.

As seen in Fig. 6.10, the performance of the algorithm is satisfactory for the contact position in dynamic simulations. To quantify this and examine the time history of the estimations, Fig. 6.11 is presented. These results are obtained from dynamic simulations where the receiver aircraft is commanded to fly at contact position. In Fig. 6.11, top plot shows commanded and actual positions of the receiver aircraft in y-direction, as well as the position estimations provided by the Algorithm-11. The bottom plot in the same figure shows the z-components of the estimations. On the right side of the figure (Fig. 6.11(b)), the Euler angles of the receiver aircraft relative to the tanker, are shown for the same period of time of the simulations. The reason for providing Euler angle information side by side to estimation plots is the fact that changes in Euler angles directly affect the position estimation procedure. This is because in the static simulations used to generate wind field data to construct the nonuniform wind maps, the orientation of the receiver relative to the tanker is set to be fixed. However, in dynamic simulations representing actual flight, aircraft trim orientation varies with relative position because the receiver is to exposed to different induced aerodynamic moments. Any change in the orientation of the receiver moves the locations of the sensors in the wind field and different airdata variable readings are obtained at the same relative position. Also, this can lead to no estimation case where the algorithms cannot find any intersection point, thus cannot generate an output. At the contact position, however, the algorithms perform satisfactorily, and this is because nonuniform wind has a symmetry axis through the contact position. In the beginning of the simulations (up to 42 seconds), the algorithm cannot provide any estimation. This is due to fact that at the beginning of the simulation, the wake induced wind is turned off. The wind is turned on after 15 sec, then the system gives the transient response. Once it reaches the steady state, the algorithm provides

estimations. Also by looking to the Euler angles, especially θ in the same period of time, this situation can be identified.

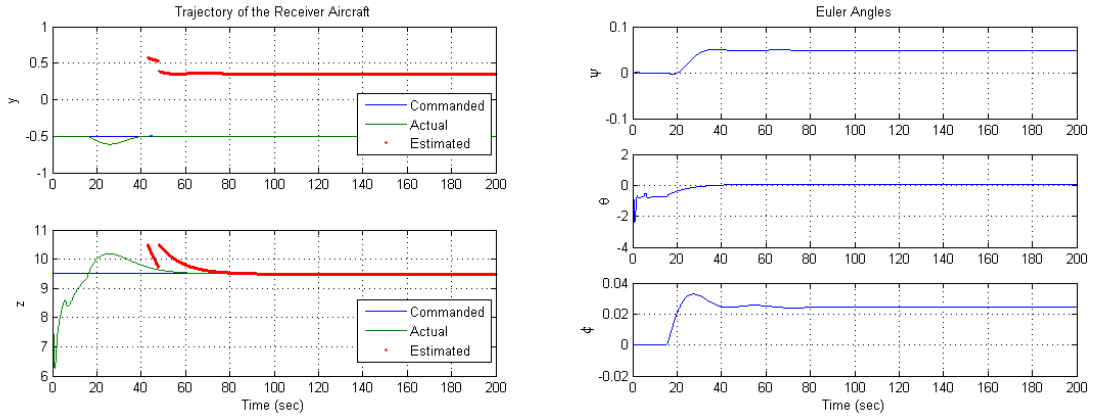


(a) (b)

Figure 6.11. Estimated Position Results at Contact Position (a) Estimation in y and z-direction (b) Euler angles.

In order to demonstrate the effect of the trim orientation over relative estimation performance, the time histories of the simulation results are presented for two more points other than contact position are selected, $(y = -0.5, z = 9.5)$ and $(y = 0.5, z = 6.5)$. In Fig. 6.12, y and z components of the results for the first point are given. Unlike the results from the contact position, the error in y direction during these simulations does not approach to zero, instead it goes to a constant value. The estimation in the z-direction, however, still goes to zero after the initial absence of the output of estimation. Recalling the reason of this caused by the the trim orientations in the dynamic case.

At the point $(y = 0.5, z = 6.5)$, the similar results are obtained with the previous point, only the direction of the error in y-direction is changed. This can be explained by the symmetry feature of the nonuniform wind field through the contact

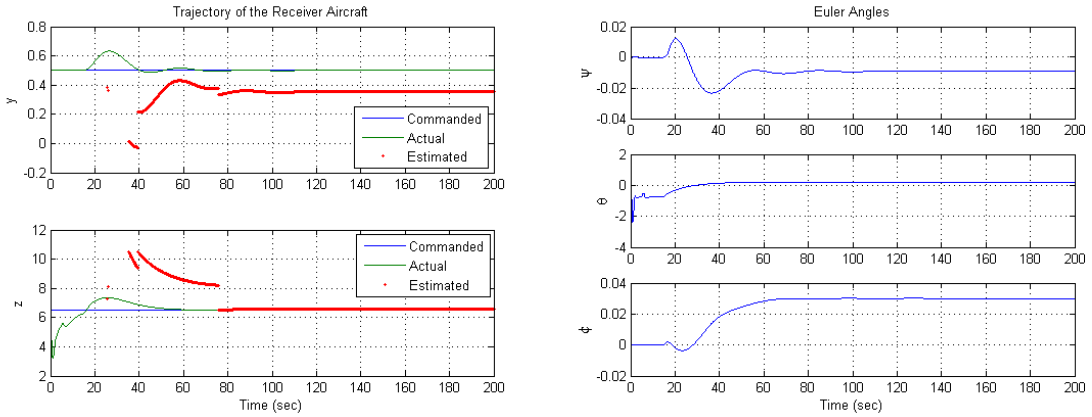


(a) (b)

Figure 6.12. Estimated Position Results at $(y = -0.5, z = 9.5)$ (a) Estimation in y and z-direction (b) Euler angles.

position. Also, by checking the Euler angle values, it can be concluded that the deviation in position estimation outputs are because of the trim orientation of the aircraft in dynamic case is different than the aircraft's orientation used in construction of the maps wind field. Since the receiver is exposed to different aerodynamic force and moments in the nonuniform wind field, in dynamic simulations, the controller trims the aircraft with different orientations at different relative positions. The difference in orientation between the static setup that generates the maps and the dynamic simulation when the aircraft is trimmed, results in different relative positions of the individual airdata sensors relative to the tanker. Hence, this causes a variation in sensor readings from their static cases, which leads to deviation in estimation performance or the algorithms cannot generate any outputs.

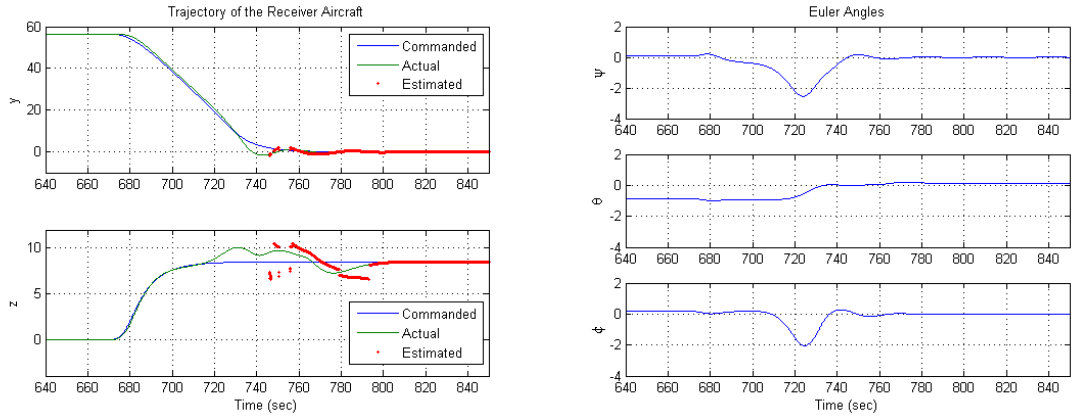
The performance of the estimation algorithm is also evaluated during the maneuver from the observation to contact position. In this simulation case, the nonuniform wind maps on the extended grid are used. In Fig. 6.14, the y and z-component of the estimations and the Euler angles during the same period are provided. The



(a) (b)

Figure 6.13. Estimated Position Results at $(y = 0.5, z = 6.5)$ (a) Estimation in y and z-direction (b) Euler angles.

algorithm cannot produce any estimate when the receiver is at the observation position and during the transition maneuver to contact position. Three possible reasons are considered contributing to the failure of the algorithm. The main factor is the difference in orientation of the receiver from that in the static simulations used to generate the nonuniform wind maps. The orientation difference is prominent in the observation position and during the maneuver. The second factor, particularly at the observation position, is the fact the induced wind is very small as the receiver is away from the wake of the tanker. This results in minimal variation in the wind maps and thus large deviation in the inverse map even with small orientation differences. The third factor, effective during the maneuver, is the variation of the sensor readings due to the motion of the receiver. To improve the position estimation algorithms even further, the trim orientation on the sensor readings should be incorporated into either the construction of the maps or into the processing of the inverse maps. This effort will result in dynamic simulation to have closer performance to the static simulations.



(a) (b)

Figure 6.14. Estimated Position Results During Maneuver (a) Estimation in y and z-direction (b) Euler angles.

6.2 Airdata Sensor Fault Detection and Isolation Simulation Results

In this study, the FDI system is not integrated into the feedback control system; the FDI runs as a separate unit. The tanker aircraft is flown by its controller with constant airspeed and at constant altitude in a straight level flight. In the simulations, the variance of sensor noise for airspeed, sideslip angle and angle of attack measurements are set as 0.2 $(m/s)^2$, 0.002 $(deg)^2$ and 0.002 $(deg)^2$, respectively. The data sets are collected with a sampling rate of 20 Hz. The tanker is flying at the speed of 200 m/sec, with the nominal altitude of 7010 meters.

6.2.1 Solo Flight Simulation Results and Parameter Study

In solo flight simulations, first FDI algorithm is analyzed without any fault injection. While the tanker aircraft flies straight level, the receiver aircraft flies at a constant position relative to the tanker. In the simulation, the wake vortex effect is turned off to represent the receiver aircraft flying solo. The commanded position and actual position of the receiver aircraft relative to the tanker is shown in Fig.

6.15(a) in terms of the coordinates in the body frame of the tanker. The receiver aircraft flies at the observation position without the wake effect of tanker aircraft. In Fig. 6.15(c), the relative angle difference index value varies due to the fact that the normalized residuals scatter within the disc centered at the origin (Fig. 6.15(b)) and they do not favor a vector direction. Thus, there is no distinct difference throughout the simulation. Additionally, two more index plots, Alarm Index (AI) and Sensor Isolation Index (SII), are presented. In Fig. 6.15(d), the top plot shows the alarm index. Whenever the chi-square test results fail to reject, in other words, whenever the normalized residuals fall out of the disc prescribed by the threshold h, the alarm index gives 1. Otherwise, in the case of acceptance of H_0, the alarm index gives 0, that can be interpreted as there is no fault detected. It should be noted that, alarm index also gives value 1 with the false alarm probability α. The bottom plot in Fig. 6.15(d) represents the sensor isolation index plot. Fault index points out the sensor number which has the fault by checking minimum angle difference between the normalized residuals and the column vectors of matrix $\mathbf{V}$. The fault index value is depended on the alarm index. Only when the alarm index gives 1, which means detection of a fault, the fault index checks for the minimum angle difference index and gives the corresponding sensor number which has the fault. In the case of the alarm index being 0, the fault index shows 0 value. Similarly, because of the false alarm probability, the fault index gives 1, 2 or 3 in a random fashion even though no fault exists. The False Alarm Percentage (FAP) values for solo flight without fault injection are given in Table 6.1. As it is expected, the FAP values turn out to be around 1%. It should be noted that, for all cases, the false alarm probability, α, is set as 1%

The detection and isolation performance of the algorithms is also tested in the simulations. The fixed type faults (bias-type) are introduced into measurement model

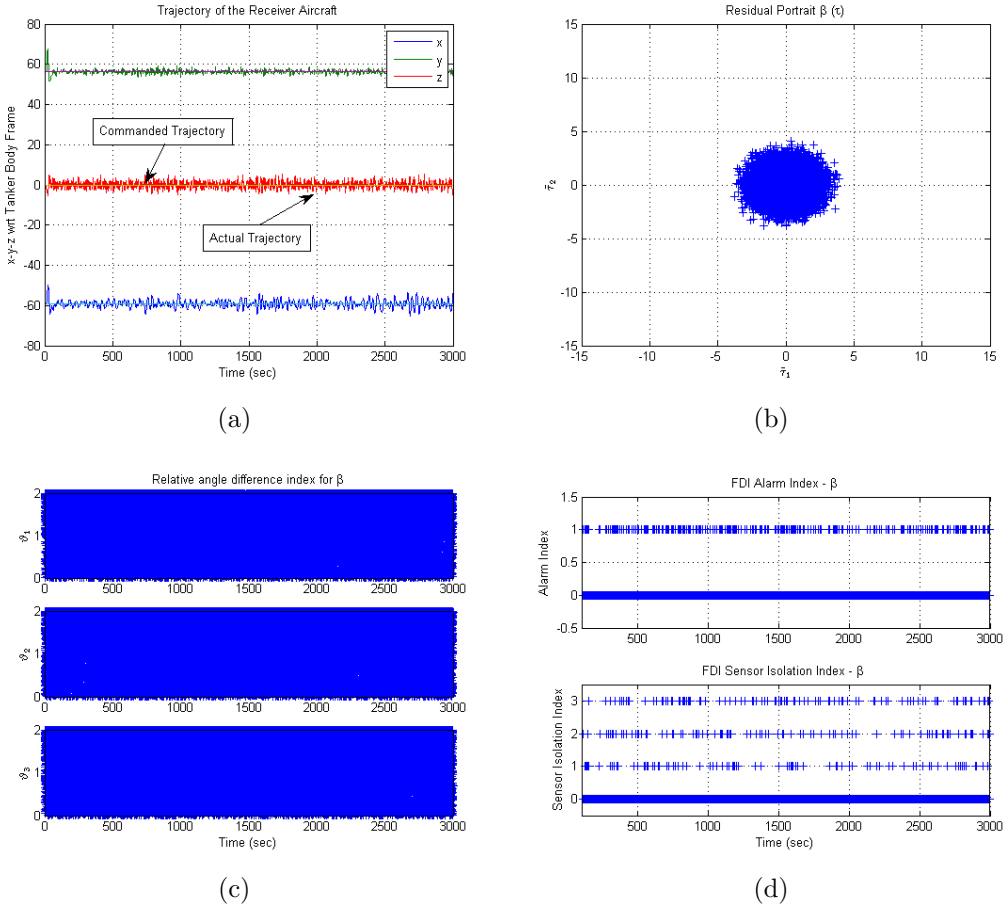


Figure 6.15. (a)Commanded Trajectory and the x-y-z Positions of the Receiver Aircraft (b) Residual Portrait of Side slip Angle Data of the Receiver Aircraft during the Flight without the Wake Vortex Effect of Tanker Aircraft (c) Relative Angle Difference Index between Residual Vector and the Column Vectors of the Null Space (d) FDI Alarm Index and Sensor Isolation Index.

Table 6.1. Alarm Percentages of FDI in Straight Level Solo Flight

		Alarm Percentages		
		Airspeed	**Sideslip Angle**	**Angle of Attack**
No Fault	***FAP***	0.99%	0.99%	0.95%
Fault	***TAP***	99.40%	100%	100%
Injected	***FAP***	1.01%	1.03%	0.91%

by changing the fault vector, F(k). The faults are introduced on only one sensor at a time with the values of 3 m/s for airspeed and 1 deg for sideslip angle and angle of attack. Different variations of fault injection are applied in the simulations. In one case, bias-type fault is created on sensor 2 between 500 and 1500 seconds, in other case fault is added to sensor 1 between 1125 and 2125 seconds, and in another simulation, fault is introduced on sensor 3 between 1750 and 2750 seconds. In all these cases, FDI algorithm successfully detects the fault and pinpoints the sensor which has the fault (Fig. 6.16). In Fig. 6.16(a), AI and SII values are shown for airspeed measurements where the fault is injected during the simulation to the sensor 2 between 500 and 1500 seconds. During the fault occurrence from 500 to 1500 seconds of the simulation, the AI notifies the fault detection by giving value 1 consistently. Moreover, fault index marks the sensor number 2 as the faulty sensor during that period, again consistently. Similarly, Fig. 6.16(b) and 6.16(c) show the AI and SII for sideslip angle measurements where the fault is injected to the sensor 1 between 1125 and 2125 seconds, and for angle of attack measurements where the fault is injected to the sensor 3 between 1750 and 2750 seconds, respectively. The True Alarm Percentage (TAP) and False Alarm Percentage (FAP) values for solo flight with fault injection are given in Table 6.1. It should be noted that the FAP in this case is calculated for the period except the fault injected interval. The FDI algorithm detects faults for sideslip angle and angle-of-attack measurement without missing any alarm case (TAP=100%), and for airspeed measurements it detects faults 99.40% of the time. And all FAP values which cover the remaining simulation period are around 1%.

In addition to the above simulation experiments, the performance of the FDI algorithm is investigated with varying sensor noise levels as well, and it is observed that the noise level affects the fault detection performance (Fig. 6.17). As the noise

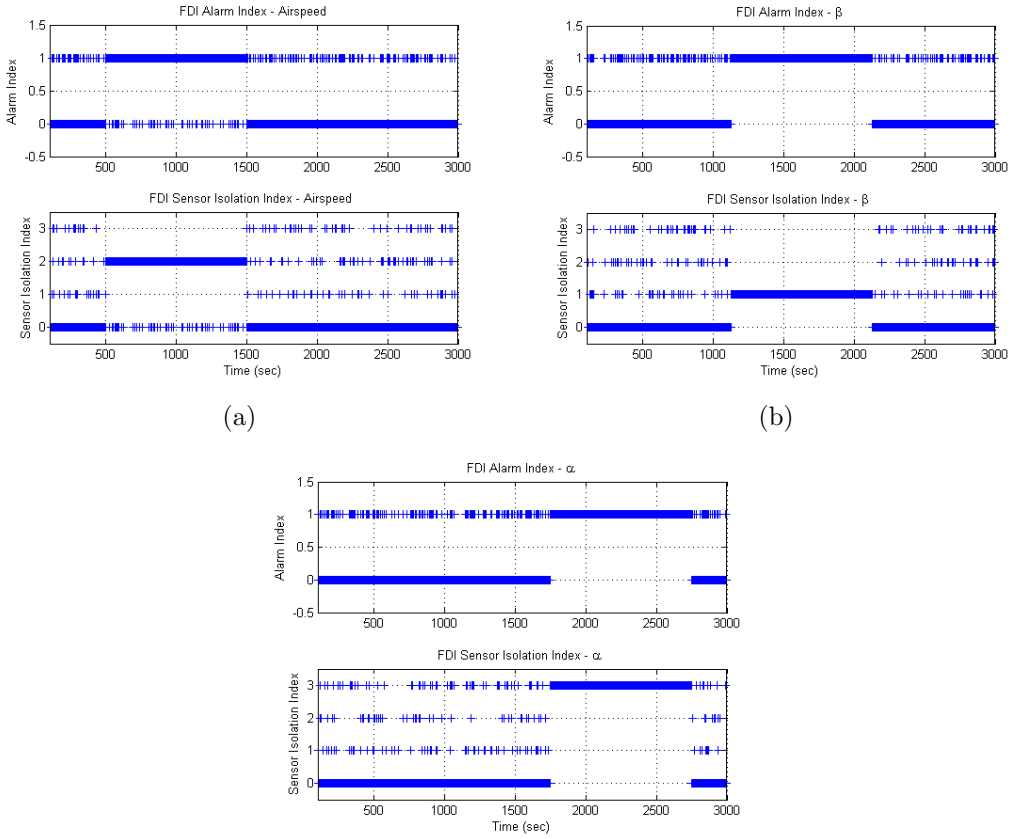


Figure 6.16. FDI Alarm Index and Sensor Isolation Index (a) For Airspeed Measurements where Fault is Injected to Sensor 2 between 500 and 1500 sec (b) For Side slip Angle Measurements where Fault is Injected to Sensor 1 between 1125 and 2125 sec (c) For Angle of Attack Measurements where Fault is Injected to Sensor 3 between 1750 and 2750 sec.

variance increase, the chance of detecting small differences between sensors, which could be a indication of a fault, decreases. This observation can also be made by a closer look at the normalization process of the residuals in Eq. (5.27). The residuals are pre-multiplied by a term which includes covariance matrix. In a faulty-case, when the fault term in the residual equation in Eq. (5.25) is not zero, if the sensor noise variances increase, this results in decrease of contribution of sensor fault values in the sensor readings. Further, similar analysis is done for magnitude of the additive

fault (Fig. 6.18). If the value of fault is bigger, in other words if the difference of one sensor readings from the other sensors is larger, the FDI algorithm detects that fault more easily.

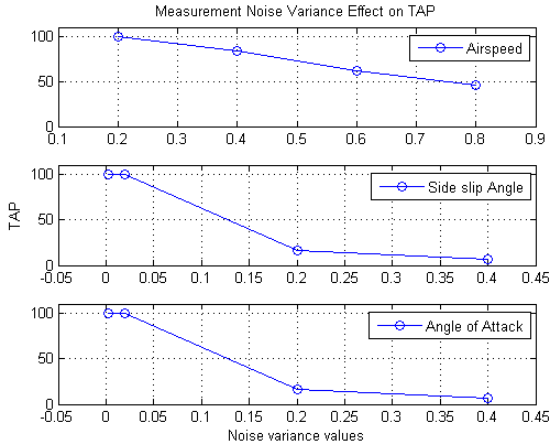


Figure 6.17. Measurement Noise Variance Effect on the Performance of FDI in terms of *TAP*.

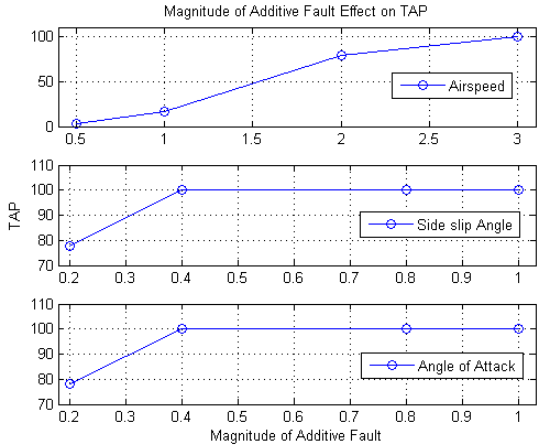


Figure 6.18. Magnitude of Additive Fault Effect on the Performance of FDI in terms of *TAP*.

6.2.2 Aerial Refueling Flight Simulation Results

In aerial refueling flight simulations, the wake vortex effect is turned on and no fault is injected into the sensor readings throughout the simulation. While the tanker aircraft flies straight level, the receiver aircraft maneuvers from the observation position to the contact position. The observation position is at the same altitude as the tanker's on the right side of the tanker, outside the wake of the tanker. Thus, the effect of the wake vortices in the wind field the receiver flies in is minimal.

At the contact position, the receiver is in the downwash region of the vortex induced nonuniform wind. The receiver transitions from the observation to the contact position by (i) lowering its altitude from that of the tanker to that of the contact position, (ii) maneuvers laterally to the centerline of the tanker, and (iii) moves forward towards the tanker until it reaches the contact position.

The tanker is flying at the speed of 200 m/sec, with the nominal altitude of 7010 meters. The simulation starts when the tanker is in a straight-level flight and stays in this condition and the receiver aircraft follows the tanker. For the first 670 seconds of 3000 seconds of the total simulation time, the receiver aircraft flies at the observation position, and then it moves from the observation position (coordinates (-59.13, 56.33, 0) in meters with respect to tanker's body frame) to the refueling contact position (coordinates (-35.5, 0, 8.5) in meters with respect to tanker's body frame)(Fig. 6.19(a)).

As stated previously, when the receiver moves behind the tanker, it enters into the nonuniform wind field induced by the vortices of the tanker. Thus, the three airdata sensors are exposed to different wind vectors, especially when receiver is at the contact position. If FDI system with the assumption of three identical measurement is used (Eq. (5.1)), the algorithm leads to false detection and isolation although there is no fault introduced to the system. The corresponding plots are given in Fig. 6.19

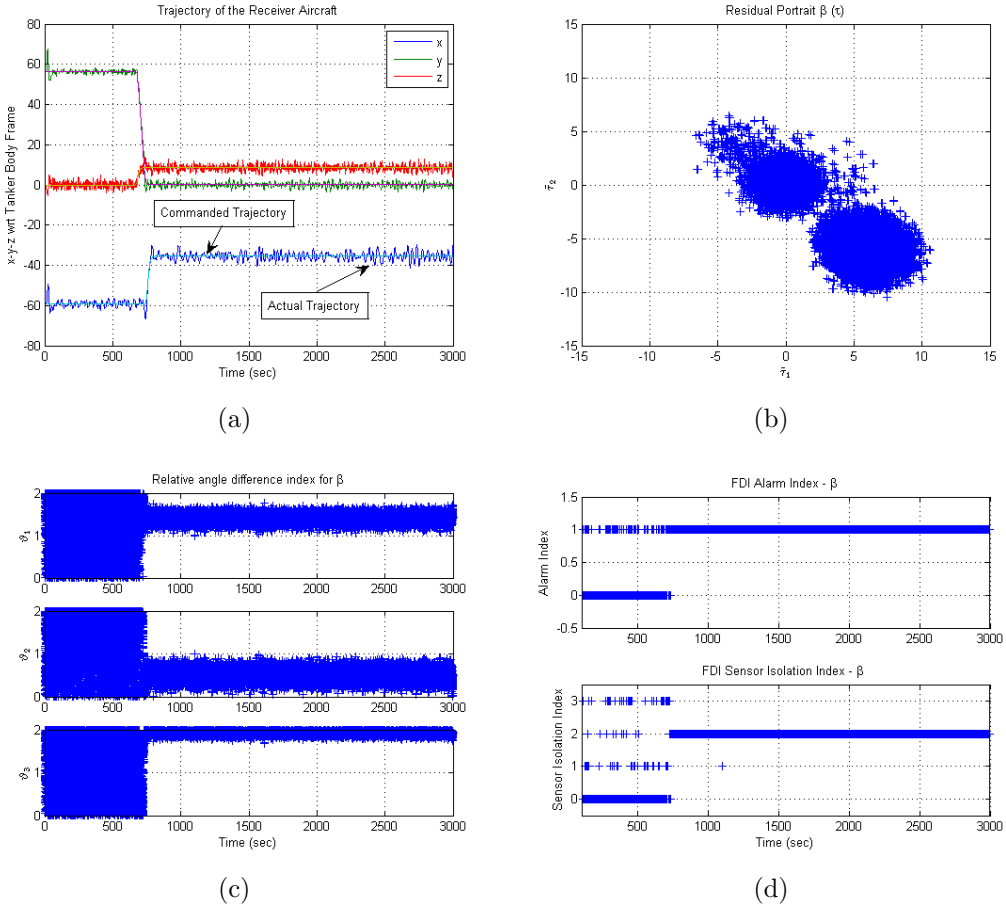


Figure 6.19. (a)Commanded Trajectory and the x-y-z Positions of the Receiver Aircraft (b) Residual Portrait of Side Slip Angle Data during the Flight with Receiver Aircraft is Exposed to Wake Vortex of Tanker Aircraft (c) Relative Angle Difference Index between Residual Vector and the Column Vectors of the Null Space (d) FDI Alarm Index and Sensor Isolation Index.

for sideslip angle data. The x-y-z coordinates of the receiver aircraft with respect to tanker aircraft's body frame is given in Fig. 6.19(a). After 670 seconds, receiver aircraft starts to maneuver from observation position to contact position, where it is exposed to wake vortex of the tanker aircraft. Thus, the normalized residuals move away from the origin (Fig. 6.19(b)), which leads to the acceptance of H_1. As a result, the algorithm gives alarm consistently and indicates a faulty sensor after the receiver aircraft starts its maneuver from observation position to contact position

(Fig. 6.19(d)), although there is no fault in the system at all. In Fig. 6.19(b), the relative angle difference index $\vartheta_2(k)$ conclusively appear to be minimum, which indicates that the normalized residuals are located in the vector direction of second column vector of the matrix $\mathbf{V}(k)$ at the contact position. As a result, the fault index consistently marks the sensor 2 (the one on the left wing) as it has fault. In airspeed and angle of attack data, this false detection and isolation is not clear as it is in sideslip angle, however wake vortex causes an increase in the false alarm probability in all measurements. The resulting false alarm percentages are given in Table 6.2. As listed in the table, an increase in false alarm percentage is observed during the maneuver from the observation position to the contact position due to the nonuniform wind distribution. This results in a false fault detection. False alarm percentages are calculated in three phases of the simulation. In the first phase, the receiver aircraft flies at the observation position. It is indicated that even in the observation position, the receiver aircraft is affected by the wake vortices of the tanker aircraft, and the false alarm percentages appear to be higher than the expected value 1%. During the maneuver, the false alarm percentages are significantly increased, yielding false fault detection. But most important result appears at the contact position stage. Sideslip angle measurements consistently give alarm although no fault exists in the system. In other words, at the contact position due to nonuniform wind field, FDI algorithm is led to a false fault detection decisively with sideslip angle readings.

6.2.3 Flight around Contact Position Simulation Results using Robust FDI

In the third group of simulations, robust FDI algorithm is used instead of FDI which leads to false fault detection in nonuniform wind field. Contrary to previous two simulation group, in this group of simulations prevailing wind and turbulence effects are turned off, and only the wake vortex effect is kept on in order to observe the

Table 6.2. False Alarm Percentages of FDI during Aerial Refueling Maneuver

	False Alarm Percentages (*FAP*)		
	Airspeed	**Sideslip Angle**	**Angle of Attack**
Before Maneuver	1.10%	1.11%	2.16%
During Maneuver	0.87%	77.03%	15.27%
After Maneuver	1.15%	100%	2.42%

performance of robust FDI. Recall that the robust FDI uses expected airdata values which are obtained from the nonuniform wind maps based on the relative position. It should be noted that in the simulation, relative position values, $(y(k), z(k))$, are assumed to be available from a relative position sensor as discussed in Section 1.2.3. Noted that the measurement noise of this position information is not taken into consideration in this study, and is left as a future work. Simulations are run while the receiver aircraft flies at the contact position. In Fig. 6.20(a) the trajectory of the aircraft relative to tanker is given. In these simulations, robust FDI algorithm is analyzed first without any fault injection. In Fig. 6.20, resulting plots are given for sideslip angle measurements. The relative angle difference index value that is given in Fig. 6.20(c), varies due to the fact that the normalized residuals scatter within the disc centered at the origin (Fig. 6.20(b)) and they do not favor a vector direction. Thus, there is no distinct difference throughout the simulation which indicates that no fault is detected and isolated. In Fig. 6.20(d), the top plot shows the AI, and the bottom plot in Fig. 6.20(d) represents the SII. The False Alarm Percentage (FAP) values for simulations at contact position without fault injection are given in Table 6.3. By using robust FDI, the FAP values at contact position are found around 1%, which indicates that the false fault detection problem is successfully avoided.

Also in this set of simulations, the detection and isolation performance of the algorithms is evaluated. As done in the first group of simulations, the fixed type faults

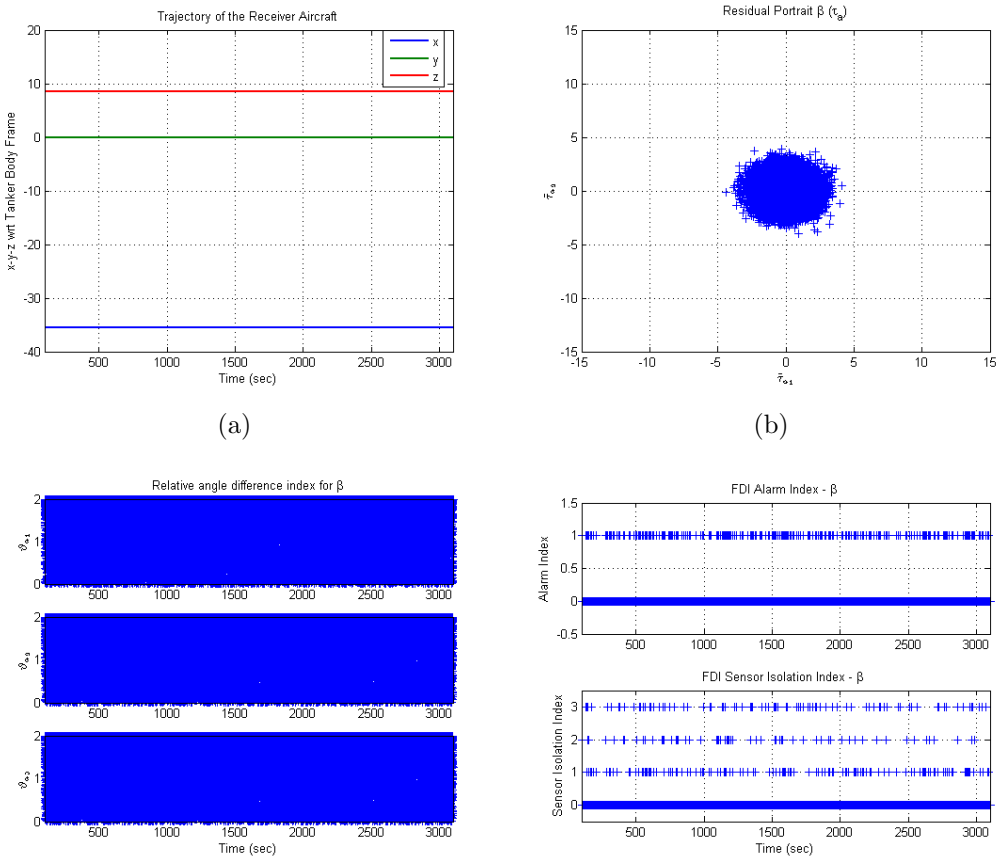


Figure 6.20. (a)Commanded Trajectory and the x-y-z Positions of the Receiver Aircraft (b) Residual Portrait of Side Slip Angle Data of the Receiver Aircraft during the Flight with the Wake Vortex Effect of Tanker Aircraft at Contact Position (c) Relative Angle Difference Index between Residual Vector and the Column Vectors of the Null Space (d) FDI Alarm Index and Sensor Isolation Index.

Table 6.3. Alarm Percentages of Robust FDI at Contact Position

		Alarm Percentages		
		Airspeed	**Sideslip Angle**	**Angle of Attack**
No Fault	***FAP***	1.01%	0.97%	0.95%
Fault	***TAP***	99.40%	100%	100%
Injected	***FAP***	1.04%	0.93%	0.91%

(bias-type) are introduced into measurement model by changing the fault vector, F(k). The faults are introduced on only one sensor at a time with the values of 3 m/s for airspeed and 1 *deg* for sideslip angle and angle of attack. Different variations of fault injection are applied in the simulations. In one case, a bias-type fault is introduced at sensor 2 between 500 and 1500 seconds; in other case, a fault is added to sensor 1 between 1125 and 2125 seconds; and in another simulation, a fault is injected on sensor 3 between 1750 and 2750 seconds. In all these cases, robust FDI algorithm successfully detects the fault and pinpoints the faulty sensor (Fig. 6.21). In Fig. 6.21(a), *AI* and *SII* values are shown for airspeed measurements when the fault is injected at the sensor 2 between 500 and 1500 seconds. During the fault occurrence, the *AI* notifies the fault detection by giving value 1 consistently. Also, *SII* marks the sensor number 2 as the faulty sensor during that period, again consistently. Similarly, Fig. 6.21(b) and 6.21(c) show the *AI* and *SII* for sideslip angle measurements where the fault is injected to the sensor 1 between 1125 and 2125 seconds, and for angle of attack measurements where the fault is injected to the sensor 3 between 1750 and 2750 seconds, respectively. The *TAP* and *FAP* values with fault injection are given in Table 6.3. It should be noted that the *FAP* in this case is calculated during the time with no fault. The FDI algorithm detects faults for sideslip angle and angle-of-attack measurement without missing any alarm case (*TAP*=100%), and for airspeed measurements it detects faults 99.40% of the time. And all *FAP* values which cover the remaining simulation period are around 1%. These results demonstrate that the robust FDI algorithm successfully overcome the false fault detection problem caused by nonuniform wind field, and it successfully detects and isolates any injected faults.

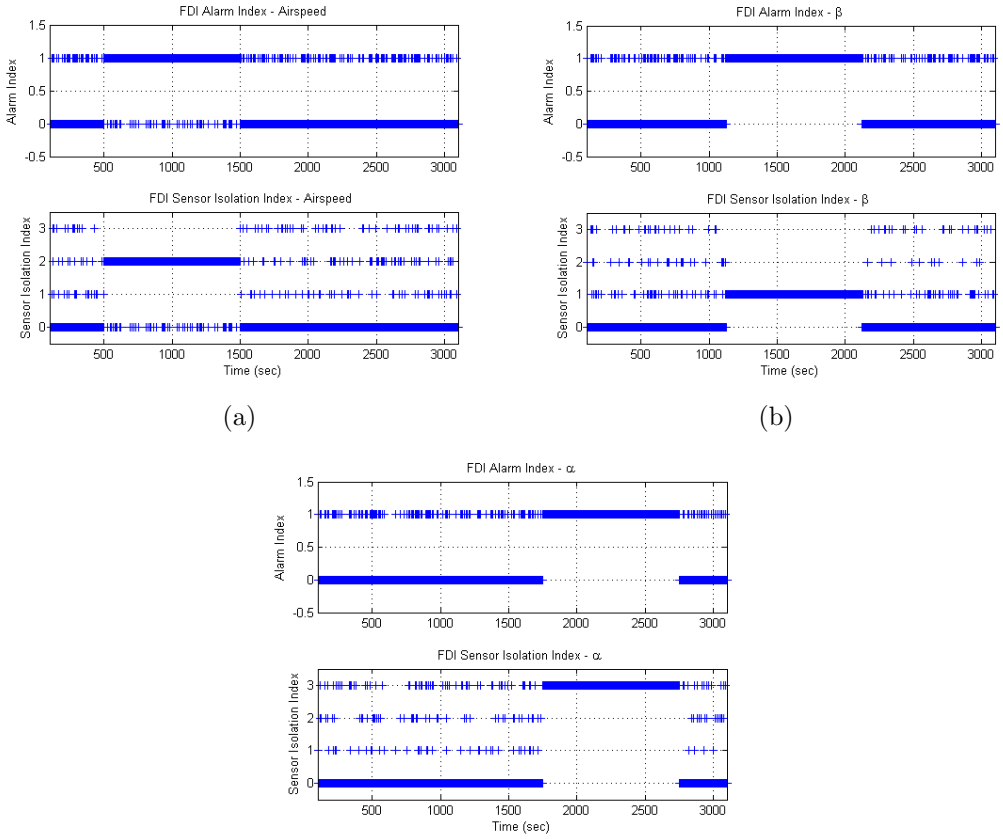


Figure 6.21. Robust FDI Alarm Index and Sensor Isolation Index (a) For Airspeed Measurements where Fault is Injected to Sensor 2 between 500 and 1500 sec (b) For Side Slip Angle Measurements where Fault is Injected to Sensor 1 between 1125 and 2125 sec (c) For Angle of Attack Measurements where Fault is Injected to Sensor 3 between 1750 and 2750 sec.

6.2.4 Aerial Refueling Flight Simulation Results using Robust FDI

Finally, aerial refueling flight simulations using robust FDI are performed. In this section, the robust FDI algorithm is analyzed first without any fault injection while the receiver aircraft maneuvers from the observation position to the contact position. During simulations, prevailing wind and turbulence effects are turned off, and only the wake vortex effect is kept on in order to observe the performance of robust FDI. Recall that the robust FDI uses expected airdata values obtained from

the nonuniform wind field maps based on the relative position. The following results are obtained with using the maps in the extended grid (Figs. 3.13 to 3.15).

The commanded position and actual position of the receiver aircraft relative to the tanker is shown in Fig. 6.22(a) in terms of the coordinates in the body frame of the tanker. The receiver aircraft flies at the observation position first, and then it moves from the observation position to the contact position. The receiver is exposed to the nonuniform wind field, when it starts its maneuver towards to contact position. Although the new robust FDI algorithm is used here to eliminate this effect, the effect of maneuver and cross winds appears in the normalized residuals plot (Fig. 6.22(b)) of side slip angle measurements as residuals move away from origin. Unlike the fault occurrence case where residuals are shifted from origin to another single point, in this case residuals move arbitrarily around the origin. Mainly it is due to the fact that the robust FDI is not affected by the nonuniform wind as much as the FDI, even during maneuver from the observation to contact position. This can be observed by comparing the relative angle difference index plot of FDI (Fig. 6.19(c)) and robust FDI (Fig. 6.22(c)). In the robust FDI results, the relative angle difference index does not show any obvious distinction in any sensor. Similarly, the AI and SII show no distinct difference throughout the simulation (Fig. 6.22(d)).

The resulting false alarm percentages are given in Table 6.4. As listed in table, an increase in false alarm percentage is observed during the maneuver from the observation position to the contact position due to the nonuniform wind distribution. This results in a false fault detection. False alarm percentages are calculated in three phases of the simulation. In the first phase, the receiver aircraft flies at the observation position, then in the second phase it maneuvers from the observation position to the contact position, and in the last phase it flies at the contact position. During the maneuver, the false alarm percentages are increased, yielding false fault detection.

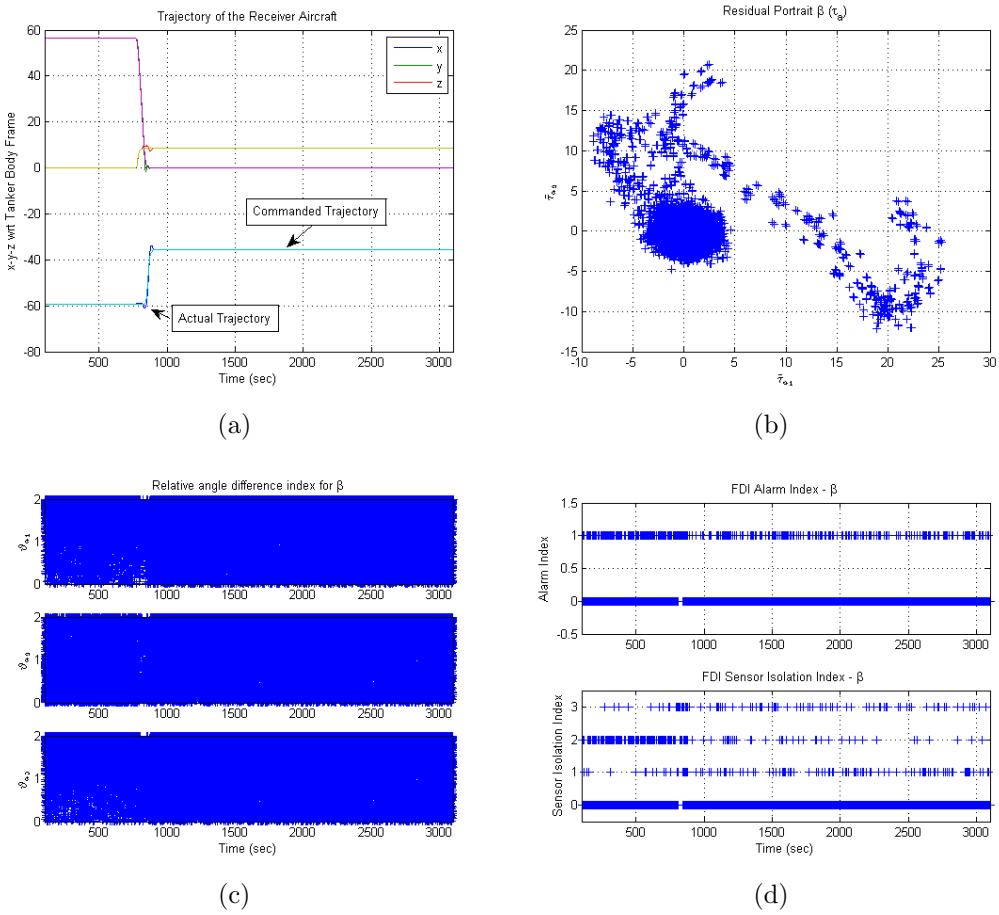


Figure 6.22. (a)Commanded Trajectory and the x-y-z Positions of the Receiver Aircraft (b) Residual Portrait of Side Slip Angle Data of the Receiver Aircraft during the Flight with Receiver Aircraft is Exposed to Wake Vortex of Tanker Aircraft (c) Relative Angle Difference Index between Residual Vector and the Column Vectors of the Null Space (d) FDI Alarm Index and Sensor Isolation Index.

However, by using robust FDI algorithm, the false alarm percentages are reduced considerably. In all three phases, the false alarm problem is avoided for airspeed measurements. For angle of attack measurements, the false alarm percentage is still slightly higher than 1% during the maneuver, but at the contact and observation positions, the false alarm percentages are less than 1%. Although the percentage value is reduced by more than half for side slip angle during the maneuver by using robust FDI, the value is still too high to be considered as false alarm free. Meanwhile, the

false alarm percentage of side slip angle at contact position is reduced to less than 1%, which means the false fault detection is avoided. This is particularly important because the FAP in the same phase is 100% if the FDI is used.

Table 6.4. False Alarm Percentages of Robust FDI in Aerial Refueling Flight

	False Alarm Percentages (*FAP*)		
	Airspeed	**Sideslip Angle**	**Angle of Attack**
Before Maneuver	1.02%	2.82%	0.94%
During Maneuver	1.07%	36.03%	2.33%
After Maneuver	1.00%	0.93%	0.98%

The detection and isolation performance of the robust FDI is also tested in the simulations. The fixed type faults (bias-type) are introduced into measurement model with the values of 3 m/s for airspeed and 1 deg for sideslip angle and angle of attack. In the first case, bias-type fault is added on sensor 2 between 500 and 1500 seconds, in the other case fault, a is added to sensor 1 between 1125 and 2125 seconds, and in the last simulation, a fault is introduced on sensor 3 between 1750 and 2750 seconds. In all these cases, the robust FDI algorithm successfully detects the fault and pinpoints the sensor with the fault (Fig. 6.23). In Fig. 6.23(a), AI and SII values are shown for airspeed measurements, in Fig. 6.23(b) for sideslip angle measurements, and in Fig. 6.23(c) for angle of attack measurements. The TAP and FAP values for aerial refueling flight with fault injection are given in Table 6.5.

It should be noted that the FAP in faulty case is calculated for the period except the fault injected interval. The robust FDI algorithm detects faults for side slip angle and angle of attack measurement without missing any alarm case (TAP=100%), and for airspeed measurements it detects faults 99.42% of the time. And all FAP values covering the remaining simulation period are around 1%, except for side slip angle

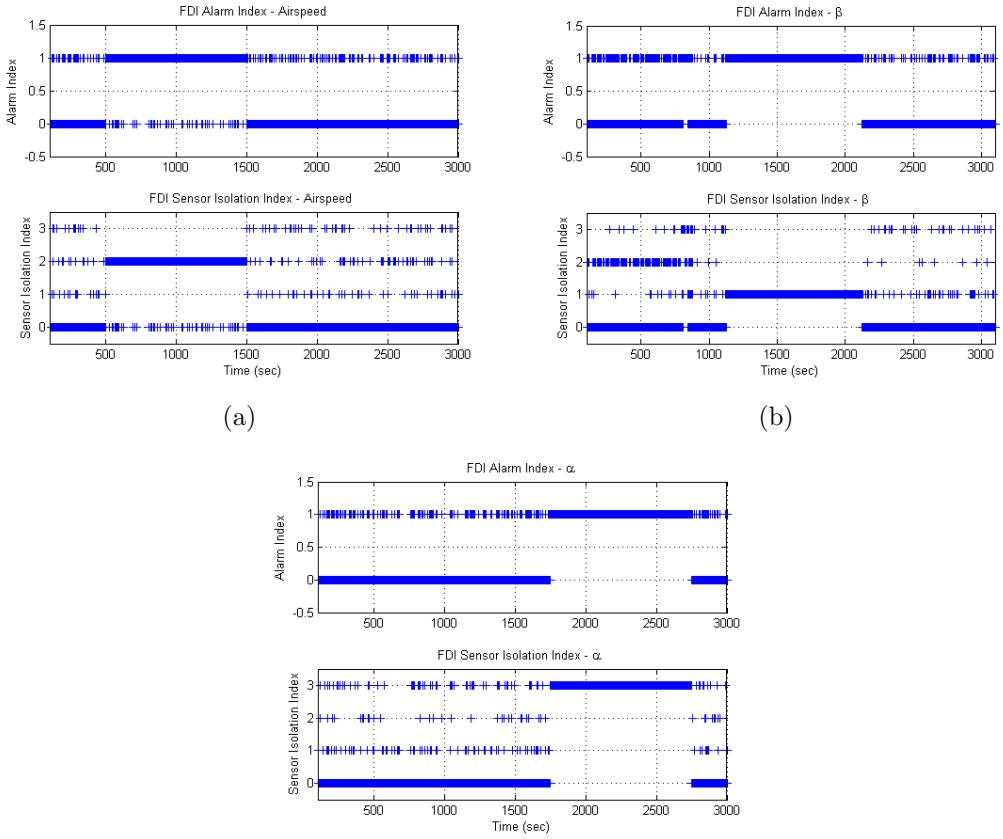


Figure 6.23. Robust FDI Alarm Index and Sensor Isolation Index (a) For Airspeed Measurements where Fault is Injected to Sensor 2 between 500 and 1500 sec (b) For Side Slip Angle Measurements where Fault is Injected to Sensor 1 between 1125 and 2125 sec (c) For Angle of Attack Measurements where Fault is Injected to Sensor 3 between 1750 and 2750 sec.

Table 6.5. Alarm Percentages of Robust FDI in Aerial Refueling Flight

		Alarm Percentages		
		Airspeed	**Sideslip Angle**	**Angle of Attack**
No Fault	***FAP***	1.00%	3.11%	1.04%
Fault	***TAP***	99.42%	100%	100%
Injected	***FAP***	1.03%	4.13%	1.04%

measurements. Even in side slip angle measurements, the FAP is 3.11% in no fault case, and it is 4.13% in the faulty case. Recall the fact that the reason for higher FAP is the maneuver of the receiver aircraft. Considering the straight flights of the receiver aircraft at observation position and at contact position, it is obvious that the false alarm in the FDI occurred due to the nonuniform wind field, is successfully avoided when the robust FDI is employed.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

In aerial refueling, the tanker aircraft induces a nonuniform wind field in its wake where the receiver aircraft needs to fly. The nonuniform wind field has wind velocity vector with varying magnitude and direction depending on the relative position with respect to the tanker. As a result, the receiver aircraft experiences varying induced aerodynamic forces and moments depending on its position relative to the tanker. The airdata sensors on the receiver aircraft measure varying airspeed, side slip angle and angle of attack as the receiver moves relative to the tanker. Further, multiple airdata sensors placed at different locations on the receiver aircraft measure different airdata variables as the nonuniform wind field has a wind variations in magnitude and direction over the length and span of the aircraft. This research develops a relative position estimation method by utilizing the nonuniform wind field as a signature of the tanker aircraft position. Further, this research develops a robust redundant-sensor-based FDI (Fault Detection and Isolation) method that can successfully detects and isolates faults in multiple airdata sensors despite the differences in airdata variable readings due to the nonuniform variation of the wind over the receiver aircraft.

The inspection of the nonuniform wind field generated from a model validated by wind tunnel and flight tests and CFD simulations shows that there are isolines on a relative position plot for each of the airdata variables. Thus, there is no one-to-one mapping between the relative position and each one of the airdata variables. This means the nonuniform wind field maps expressed in terms of the airdata variables

cannot be reversed to obtain relative position information for the measurements of an airdata variable by a single sensor. It is shown that isolines of different variables from a single sensor have intersections, but in a limited region behind the tanker aircraft, which implies that position estimation is possible with at least two variables from a single sensor. The simulations of the estimation algorithms reveal that the errors in the estimated position even in the limited region where estimation is possible are very large. The simulations of other algorithms that utilize multiple airdata variables from multiple sensors shows that the feasible region (where position estimation is possible without ambiguity) becomes larger and the accuracy of the estimation improves as more variables from more sensors are used in position estimation. The best ones among the algorithms developed and evaluated in this research use two variables (side slip angle and angle of attack) from three sensors located at the wing tips and the nose of the aircraft.

Two metrics are defined and used to quantify the performance of the algorithms: (i) the size of the feasible region where it can produce a position estimate and (ii) the accuracy of the estimation in the feasible region. These two metrics are also used to evaluate the sensitivity of the algorithms against the turbulence, measurement noise and prevailing wind. The simulation results show performance degradations to various degree in position estimation in the presence of such disturbances. This is because the nonuniform wind maps used by the algorithms as inverse maps are constructed in ideal conditions when there is no turbulence, measurement noise or prevailing wind. Performance degradation is also observed when the algorithms are evaluated on-line in dynamic simulations when the tanker and the receiver are flown by controllers. Specifically, the receiver aircraft is trimmed by the controller at the commanded positions relative to the tanker. Since the receiver is exposed to different aerodynamic force and moments in the nonuniform wind field, the controller trims

the aircraft with different orientations at different relative positions. The difference in orientation between the static setup that generates the maps and the dynamic simulation when the aircraft is trimmed results in different relative positions of the individual airdata sensors relative to the tanker. Such differences causing variation in sensor readings results in degradation in estimation accuracy or failure to generate estimation especially in the regions where the wind maps have high slopes.

The standard parity-space-based FDI algorithm is developed that relies on sensor redundacy (three airdata sensors), uses a chi-square hypothesis that for detection and employs a relative angle comparison in the residual space for isolation. The FDI algorithm can work with only one faulty sensor. The simulation experiments show that the FDI system can successfully detect and isolate additive (bias) fault in any of the three airdata sensors considered in the presence of atmospheric turbulence, measurement noise and prevailing wind when the receiver flies solo or outside the wake of the tanker. The simulations also reveal that the FDI generates false fault alarms when the receiver enters the wake of the tanker. This is because the nonuniform wind field causes the three sensors to measure different airdata variables and this deceives the FDI that is based on the assumption that the sensors when no fault is present provide identical readings of airdata variables. This weakness of the FDI is eliminated by utilizing the maps of the nonuniform wind field consisting of the data of the airdata variables as functions of relative position. By employing this maps to calculate the expected values of the airdata variables depending on the relative position, the FDI algorithm no longer assumes the identical readings from the sensors. This leads to successful operation of the FDI system even in the wake of the tanker. The performance limitations of the FDI system is investigated by a parameter study that varies the variance of the measurement noise and the magnitude of the bias error.

7.2 Future Work

The following topics are recommended for future work considering the results and conclusions obtained from this research. During construction of the maps of nonuniform wind field in the case of multiple intersection points are found, estimation algorithms either picks the first or last one found, or discard them all. In making such a decision, previous known or estimated location of the aircraft could be used considering the kinematics of the aircraft. Also, position estimation algorithms rely on the nonuniform wind maps constructed from static simulation in the absence of turbulence, measurement noise, prevailing wind and the kinematics of the aircraft. When the inverse of these maps are used for position estimation in the presence of such effects, the performance of the position estimation degrades as these factors causes mismatch. Incorporation of these effects in the construction of the nonuniform wind maps should be investigated. Another important point is the method used for construction of the nonlinear wind field maps. In this study, those maps are created in simulation environment. Instead of using an analytical model in the construction of wind field maps, wind tunnel and flight test experiments as well as CFD methods could be utilized. Such methods should yield more accurate wind field maps and this improve the performance of the position estimation in real-time implementations. In addition to that, in the static simulations used to generate wind field data to construct the nonuniform wind maps, the orientation of the receiver relative to the tanker is set to be fixed. However, in dynamic simulations representing actual flight, aircraft trim orientation varies with position because the receiver is to exposed to different induced aerodynamic moments. Any change in the orientation of the receiver moves the locations of the sensors in the wind field and different and different airdata variable readings are obtained at the same relative position. The effort of the trim orientation on the sensor readings should be incorporated into either the construction of the maps

or into the processing of the inverse maps. This should cause dynamic simulation to have closer performance to the static simulations. It is shown that the proposed techniques are affected highly by the measurement noise, turbulence and prevailing winds. The methods for eliminating of these effects can further be investigated. Filtering of the measured airdata signals should help attenuate the noise in the signals, which, in turn, improve the position estimation performance. Airdata sensor based position is intended to complement other relative position estimation methods. Data fusion methods that use airdata sensor based position estimation with other relative position measurements should be investigated to improve the performance and reliability of position estimation. Further, incorporation of the position estimation system into the flight control system should be considered in order to contribute for developing autonomous aerial refueling operation.

The FDI system studied herein is based on sensor redundancy. Software redundancy methods such as Kalman filtering and neural networks should also be investigated in order to combine additional residual generating process to the existing system. This research only considered single sensor failure. Using software redundancy or a hybrid approach, the feasibility of the FDI of multiple sensor faults can be investigated. Further, incorporation of the FDI system into the flight control system should be considered to develop fault tolerant flight control systems.

Finally, the proposed systems can be tested in real-time implementations. The developed algorithms can be implemented first by the Processor-in-the-loop (PIL) technique where it allows to test proposed system on the actual microprocessor. Also, the algorithms can be compiled to a microprocessor and can be tested by using relatively simple and small platforms, before being tested in actual aircraft.

REFERENCES

[1] A. Dogan, T. A. Lewis, and W. Blake, "Flight data analysis and simulation of wind effects during aerial refueling," *Journal Of Aircraft*, vol. 45, no. 6, pp. 2036–2048, November 2008.

[2] K. Ro and J. W. Kamman, "Modeling and simulation of hose-paradrogue aerial refueling systems," *Journal Of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 53–63, 2010.

[3] J. Waishek, A. Dogan, and W. Blake, "Derivation of the dynamics equations of receiver aircraft in aerial refueling," *Journal Of Guidance, Control, and Dynamics*, vol. 32, no. 2, pp. 585–597, 2009.

[4] J. Wang, N. Hovakimyan, and C. Cao, "Verifiable adaptive flight control: Unmanned combat aerial vehicle and aerial refueling," *Journal Of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 75–87, 2010.

[5] K. Enomoto, T. Yamasaki, H. Takano, and Y. Baba, "Guidance and control system design for chase uav," in *Proc. of AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, 18-21 August 2008, pp. AIAA 2008–6842.

[6] A. Dogan, E. Kim, and W. Blake, "Control and simulation of relative motion for aerial refueling in racetrack maneuvers," *Journal Of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1551–1557, 2007.

[7] S. Venkataramanan, A. Dogan, and W. Blake, "Vortex effect modelling in aircraft formation flight," in *Proc. of AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Austin, Texas, 11 - 14 August 2003, aIAA 2003-5385.

[8] A. Dogan and W. Blake, "Modeling of bow wave effect in aerial refueling," in *Proc. of AIAA Atmospheric Flight Mechanics Conference*, Toronto, Ontario Canada, 2-5 August 2010, aIAA 2010-7926.

[9] A. Dogan, T. A. Lewis, and W. Blake, "Wake-vortex induced wind with turbulence in aerial refueling - part b: Model and simulation validation," in *Proc. of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Honolulu, Hawaii, 18-21 August 2008, aIAA 2008-6697.

[10] D. Saban, J. F. Whidborne, and A. K. Cooke, "Simulation of wake vortex effects for uavs in close formation flight," *The Aeronautical Journal*, vol. 113, no. 1149, pp. 727–738, November 2009.

[11] C. Han, L. Cho, and J. Cho, "Wake shapes behind wings in close formation flight near the ground," *Journal of Mechanical Science and Technology*, vol. 19, no. 2, pp. 674–681, 2005.

[12] J. P. Nalepka and J. L. Hinchman, "Automated aerial refueling: Extending the effectiveness of unmanned air vehicles," in *Proc. of AIAA Modeling and Simulation Technologies Conference and Exhibit*, San Francisco, California, 15-18 August 2005, aIAA 2005-6005.

[13] B. S. Burns, P. A. Blue, and M. D. Zollars, "Autonomous control for automated aerial refueling with minimum-time rendezvous," in *Proc. of AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, South Carolina, 20-23 August 2007, aIAA 2007-6739.

[14] M. D. Tandale, R. Bowers, and J. Valasek, "Robust trajectory tracking controller for vision based probe and drogue autonomous aerial refueling," in *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, 15-18 August 2005, aIAA 2005-5868.

[15] M. Mammarella, G. Campa, M. R. Napolitano, M. L. Fravolini, Y. Gu, and M. G. Perhinschi, "Machine vision/gps integration using ekf for the uav aerial refueling problem," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 38, no. 6, pp. 791–801, November 2008.

[16] DARPA News/Event Releases, "Making Connections at 45,000 Feet: Future UAVs May Fuel Up in Flight," URL: http://www.darpa.mil/NewsEvents/Releases/2012/10/05.aspx (visited on April 2, 2013).

[17] J. Doebbler, T. Spaeth, and J. Valasek, "Boom and receptacle autonomous air refueling using visual snake optical sensor," *Journal Of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1753–1769, 2007.

[18] M. D. Tandale, R. Bowers, and J. Valasek, "Trajectory tracking controller for vision-based probe and drogue autonomous aerial refueling," *Journal Of Guidance, Control, And Dynamics*, vol. 29, no. 4, pp. 846–857, 2006.

[19] J. Valasek, J. Kimmett, D. Hughes, K. Gunnam, and J. L. Junkins, "Vision based sensor and navigation system for autonomous aerial refueling," in *Proc. of AIAA Technical Conference and Workshop on Unmanned Aerospace Vehicles*, Portsmouth, USA, 20-23 May 2002, pp. AIAA–2002–3441.

[20] R. V. DellAquila, G. Campa, M. R. Napolitano, and M. Mammarella, "Real-time machine-vision-based position sensing system for uav aerial refueling," *Journal of Real-Time Image Processing*, vol. 1, p. 213224, 2007.

[21] E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer, "Real-time vision-based relative aircraft navigation," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, pp. 707–738, 2007.

[22] J. Valasek, K. Gunnam, J. Kimmett, M. D. Tandale, and J. L. Junkins, "Vision-based sensor and navigation system for autonomous air refueling," *Journal Of Guidance, Control, And Dynamics*, vol. 28, no. 5, 2005.

[23] J. A. Curro II, "Automated aerial refueling position estimation using a scanning LiDAR," Master's thesis, Air Force Institute of Technology, 2012.

[24] T. Jiang and W. Chen, "The accuracy relative position algorithm for multi-aircrafts," in *Proc. of ICISE2009 International Conference on Information Science and Engineering*, Hefei, China, 26-28 December 2009, pp. 3864 – 3867.

[25] L. Pollini, M. Innocenti, and R. Mati, "Vision algorithms for formation flight and aerial refueling with optimal marker labeling," in *Proc. of AIAA Modeling and Simulation Technologies Conference and Exhibit*, San Francisco,USA, 15-18 August 2005, pp. AIAA 2005–6010.

[26] Z. Mahboubi, Z. Kolter, T. Wang, and G. Bower, "Camera based localization for autonomous uav formation flight," in *Proc. of AIAA Infotech@Aerospace Conference*, St. Louis, USA, 29-31 March 2011, pp. AIAA 2011–1658.

[27] A. L. Smith, "Proportional navigation with adaptive terminal guidance for aircraft rendezvous," *Journal Of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1832–1835, 2008.

[28] W. R. Williamson, G. J. Glenn, V. T. Dang, and J. L. Speyer, "Sensor fusion applied to autonomous aerial refueling," *Journal Of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 262–275, 2009.

[29] S. M. Khanafseh and B. Pervan, "Autonomous airborne refueling of unmanned air vehicles using the global positioning system," *Journal Of Aircraft*, vol. 44, no. 5, pp. 1670–1682, 2007.

[30] A. M. Fosbury and J. L. Crassidis, "Relative navigation of air vehicles," *Journal Of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 824–834, 2008.

[31] G. Campa, M. L. Fravolini, and A. Ficola, "Autonomous aerial refueling for uavs using a combined gps-machine vision guidance," in *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, Rhode Island, 16-19 August 2004, aIAA 2004-5350.

[32] S. Anwar and L. Chen, "An analytical redundancy-based fault detection and isolation algorithm for a road-wheel control subsystem in a steer-by-wire system," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2859–2869, 2007.

[33] N. Meskin, C. A. Rabbath, and K. Khorasani, "A hybrid fault detection and isolation strategy for a network of unmanned vehicles in presence of large environmental disturbances," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 6, pp. 1422–1429, 2010.

[34] M. M. Tousi, A. G. Aghdam, and K. Khorasani, "A hybrid fault diagnosis for a team of unmanned aerial vehicles," in *Proc. of SoSE 2009, IEEE International Conference on System of Systems Engineering*, Albuquerque, USA, 30 May-3 June 2009, pp. 1–6.

[35] E. Kiyak, A. Kahvecioglu, and F. Caliskan, "Aircraft sensor and actuator fault detection, isolation, and accommodation," *Journal of Aerospace Engineering*, vol. 24, no. 1, pp. 46–58, 2011.

[36] C. Hajiyev, "Sensor fault detection by testing the largest eigenvalue of the innovation covariance using tracy-widom distribution," in *Proc. of 2010 American Control Conference*, Baltimore, Maryland, 30 June-02 July 2010, pp. 5427–5432.

[37] P. Freeman, P. Seiler, and G. J. Balas, "Robust fault detection for commercial transport air data probes," in *Proc. of 18th IFAC World Congress*, Milano, Italy, 28 August-2 September 2011, pp. 13 723–13 728.

[38] K. Shaoping, H. Yefa, and Z. Zude, "A redundant-sensor-based fault reasoning technique for multi-sensors," in *Proc. of 2008 IEEE Pacific-Asia Workshop*

on Computational Intelligence and Industrial Application, Wuhan, China, 19-20 December 2008, pp. 722–726.

[39] M. H. Kim, S. Lee, and K. C. Lee, "Predictive hybrid redundancy using exponential smoothing method for safety critical systems," *International Journal of Control, Automation, and Systems*, vol. 6, no. 1, pp. 126–134, 2008.

[40] S. Dajani-Brown, D. Cofer, G. Hartmann, and S. Pratt, "Formal modeling and analysis of an avionics triplex sensor voter," *Model Checking Software*, p. 3448, 2003.

[41] T. Kerr, "Decentralized filtering and redundancy management for multisensor navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-23, no. 1, pp. 83–119, 1987.

[42] D.-S. Shim and C.-K. Yang, "Geometric fdi based on svd for redundant inertial sensor systems," in *Proc. of 5th Asian Control Conference*, vol. 2, Melbourne, Australia, 20-23 July 2004, pp. 1094 – 1100.

[43] D. Berdjag, A. Zolghadri, J. Cieslak, and P. Goupil, "Fault detection and isolation for redundant aircraft sensors," in *Proc. of 2010 Conference on Control and Fault Tolerant Systems*, Nice, France, 6-8 October 2010, pp. 137–142.

[44] R. D. Eubank, E. M. Atkins, and S. Ogura, "Fault detection and fail-safe operation with a multiple-redundancy air-data system," in *Proc. of AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, 2-5 August 2010, pp. AIAA 2010–7855.

[45] H. S. Kim, S. K. Park, Y. Kim, and C. G. Park, "Hybrid fault detection and isolation method for uav inertial sensor redundancy management system," in *Proc. of the 16th IFAC World Congress*, Prague, Czech Republic, 38 July 2005, pp. 265–270.

[46] S. Kim, Y. Kim, C. G. Park, and I. Jung, "Hybrid fault detection and isolation techniques for aircraft inertial measurement sensors," in *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, USA, 16-19 August 2004, pp. AIAA 2004–5419.

[47] B. D. Brumback and M. D. Srinath, "A fault-tolerant multisensor navigation system design," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-23, no. 6, pp. 738–756, 1987.

[48] M. Mattei and G. Paviglianiti, "Managing sensor hardware redundancy on a small commercial aircraft with h_∞ fdi observers," in *Proc. of the 16th IFAC World Congress*, Prague, Czech Republic, 38 July 2005, p. 347352.

[49] T. J. Wheeler, P. Seiler, A. K. Packard, and G. J. Balas, "Performance analysis of fault detection systems based on analytically redundant linear time-invariant dynamics," in *Proc. of ACC 2011 American Control Conference*, San Francisco, USA, 29 June-1 July 2011, pp. 214–219.

[50] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 636–653, 2010.

[51] J. J. Gertler and R. Monajemy, "Generating directional residuals with dynamic parity relations," *Automatica*, vol. 31, no. 4, pp. 627–635, 1995.

[52] J. J. Gertler, "Fault detection and isolation using parity relations," *Control Engineering Practice*, vol. 5, no. 5, pp. 653–661, 1997.

[53] A. Medvedev, "Fault detection and isolation by a continuous parity space method," *Automatica*, vol. 31, no. 7, pp. 1039–1044, 1995.

[54] F. Kratz, W. Nuninger, and S. Ploix, "Fault detection for time-delay systems: a parity space approach," in *Proc. of American Control Conference*, vol. 4, Philadelphia, USA, 24-26 June 1998, pp. 2009–2011.

[55] S. R. Hali, P. Motyka, E. Gai, and J. J. Deyst, "In-flight parity vector compensation for fdi," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-19, no. 5, pp. 668–676, 1983.

[56] E. Y. Chow and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Transactions on Automatic Control*, vol. AC-29, no. 7, pp. 603–614, 1984.

[57] M. A. Sturza, "Navigation system integrity monitoring using redundant measurements," *Navigation*, vol. 35, no. 4, pp. 69–87, 1988.

[58] E. J. de Oliveira, H. K. Kuga, and I. M. da Fonseca, "Fault detection and isolation in inertial measurement units based on χ^2-cusum and wavelet packet," in *Proc. of AIAA Guidance, Navigation, and Control Conference*, Minneapolis, USA, 13-16 August 2012, pp. AIAA 2012–4753.

[59] S. Guerrier, A. Waegli, J. Skaloud, and M.-P. Victoria-Feser, "Fault detection and isolation in multiple mems-imus configurations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2015–2031, 2012.

[60] Accident Investigation Board, "United States Air Force Class A Aerospace Mishaps," URL: http://usaf.aib.law.af.mil/ (visited on March 5, 2012).

[61] United States Air Force, "B-2 Accident Report Released," URL: http://www.acc.af.mil (visited on April 5, 2012).

[62] "Airbus a330-303 in-flight upset 154 km west of learmonth, wa," Australian Transport Safety Bureau, Interim Factual, March 2009.

[63] "Final report 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," Bureau d'Enqutes et d'Analyses, Final Report, July 2012.

[64] A. Dogan, S. Venkataramanan, and W. Blake, "Modeling of aerodynamic coupling between aircraft in close proximity," *Journal of Aircraft*, vol. 42, no. 4, pp. 941–955, 2005.

[65] J. Waishek, "Derivation of the dynamics equations for receiver aircraft in aerial refueling," Master's thesis, The University of Texas at Arlington, 2007.

[66] E. Kim, "Control and simulation of relative motion for aerial refueling in racetrack maneuver," Master's thesis, The University of Texas at Arlington, 2007.

[67] T. A. Lewis, "Flight data analysis and simulation of wind effects during aerial refueling," Master's thesis, The University of Texas at Arlington, 2008.

[68] U. M. Specification, "Flying qualitied of piloted airplane," Tech. Rep., 1980, MIL-F-8785C.

[69] I. Moir and A. Seabridge, *Aircraft systems: Mechanical, Electrical and Avionics Subsystems Integration*. Wiley, 2008.

[70] C. Edwards, "Air data system workbook," National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, Tech. Rep., 2006.

[71] R. P. Collinson, *Introduction to Avionics Systems*. Springer, 2011.

[72] G. Koppenwallner, "Controlled hypersonic flight air data system and flight instrumentation," Educational Notes RTO-EN-AVT-130, Flight Experiments for Hypersonic Vehicle Development, 2007, 17-1 – 17-30.

[73] E. A. Haering, "Airdata measurement and calibration," National Aeronautics and Space Administration, Dryden Flight Research Center, Tech. Rep., 1995, NASA Technical Memorandum 104316.

[74] J. H. Lee, H. E. Sevil, A. Dogan, and D. Hullender, "Estimation of receiver aircraft states and wind vectors in aerial refueling," in *Proc. of GNC 2012, AIAA Guidance, Navigation, and Control Conference*, Minneapolis, USA, 13-16 August 2012, AIAA 2012-4533.

[75] H. E. Sevil and A. Dogan, "False fault detection in airdata sensor due to non uniform wind in aerial refueling," in *Proc. of AFM 2011, AIAA Atmospheric Flight Mechanics Conference*, Portland, USA, 08-11 August 2011, pp. AIAA 2011–6446.

[76] ——, "Airdata sensor fault detection and isolation for receiver aircraft in aerial refueling," in *Proc. of ASM 2013, AIAA Aerospace Sciences Meeting*, Grapevine, USA, 7-10 January 2013, pp. AIAA 2013–0950.

[77] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy a survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[78] H. E. Sevil, "Fault detection and isolation using redundant sensors for a wind turbine with doubly-fed induction generator (dfig)," in *Proc. of the International Symposium of Mechanism and Machine Science (AzCIFToMM 2010)*, Izmir, Turkey, 5-8 October 2010, pp. 232–238.

BIOGRAPHICAL STATEMENT

Hakki Erhan Sevil received his B.Sc. and M.Sc. degrees in Mechanical Engineering from Izmir Institute of Technology, Turkey, in 2004 and 2006, respectively. Before joining to UTA, he had over 4 years of research experience as being a Graduate Research Assistant at Izmir Institute of Technology from 2005 to 2009, and he was with Service Automation and Systems Analysis Laboratory at Universite Libre de Bruxelles as a visiting researcher in 2009. Currently, he is affiliated as a Graduate Teaching Assistant at the University of Texas at Arlington (UTA), and he is conducting research at Computer Aided Control System Design Laboratory (CACSDL) and Autonomous Vehicles Laboratory (AVL). His research interests include, fault detection and isolation, autonomous navigation and obstacle avoidance of unmanned vehicles, condition monitoring, distributed sensing and intelligent control systems.