

Article

Implementation of the Rauch-Tung-Striebel Smoother for Sensor Compatibility Correction of a Fixed-Wing Unmanned Air Vehicle

Woei-Leong Chan * and Fei-Bin Hsiao

Institute of Aeronautics and Astronautics, National Cheng Kung University, Tainan City 701, Taiwan;
E-Mail: fbhsiao@mail.ncku.edu.tw

* Author to whom correspondence should be addressed; E-Mail: p4895701@mail.ncku.edu.tw;
Tel.: +886-6-275-7575 ext. 63642; Fax: +886-6-208-8214.

Received: 10 February 2011; in revised form: 15 March 2011/Accepted: 17 March 2011 /

Published: 28 March 2011

Abstract: This paper presents a complete procedure for sensor compatibility correction of a fixed-wing Unmanned Air Vehicle (UAV). The sensors consist of a differential air pressure transducer for airspeed measurement, two airdata vanes installed on an airdata probe for angle of attack (AoA) and angle of sideslip (AoS) measurement, and an Attitude and Heading Reference System (AHRS) that provides attitude angles, angular rates, and acceleration. The procedure is mainly based on a two pass algorithm called the Rauch-Tung-Striebel (RTS) smoother, which consists of a forward pass Extended Kalman Filter (EKF) and a backward recursion smoother. On top of that, this paper proposes the implementation of the Wiener Type Filter prior to the RTS in order to avoid the complicated process noise covariance matrix estimation. Furthermore, an easy to implement airdata measurement noise variance estimation method is introduced. The method estimates the airdata and subsequently the noise variances using the ground speed and ascent rate provided by the Global Positioning System (GPS). It incorporates the idea of data regionality by assuming that some sort of statistical relation exists between nearby data points. Root mean square deviation (RMSD) is being employed to justify the sensor compatibility. The result shows that the presented procedure is easy to implement and it improves the UAV sensor data compatibility significantly.

Keywords: Unmanned Air Vehicle (UAV); sensor data compatibility; Rauch-Tung-Striebel (RTS); Wiener type filter; noise estimation

1. Introduction

Air vehicle sensor compatibility is a specific field of study dedicated to the estimation of onboard sensor errors. The sensor errors are usually described as the scale factors and the biases. The field is often discussed in the context of flight path reconstruction. The term “compatibility” refers to the consistency of the measured data with the force equations that govern the airdata (airspeed, AoA, and AoS), and the kinematic equations that govern the attitude angles (roll, pitch, yaw). If flight path reconstruction is to be discussed jointly, the navigational equations will be included. Sensor compatibility is often the first step of the so-called “two-step method” of aircraft system identification. The first step is the sensor data correction/estimation; the second part is the parameter estimation.

Research on the topic of aircraft sensor compatibility started as early as the 1970s. In 1976, Jonker [1] pioneered the research in the area by investigating the applicability of the Kalman Filter for flight path reconstruction and estimation of instrumentation biases. A year later, the National Aeronautics and Space Administration (NASA) published a report on the measured aircraft responses compatibility check. As discussed in the report, Klien and Schiess [2] implemented the EKF with a fixed-point smoother to accomplish the task. In 1984, similar work was again published by NASA. This time, Whitmore *et al.* [3] calibrated the airdata error using the Linearized Kalman Filter (LKF). Up to that point, The Kalman Filter and EKF were proven to be sufficient in estimating the air vehicle sensor errors. However, the implementations of Kalman Filter and EKF are not easy without *a priori* knowledge of the measurement noise and process noise and this noise information is often not easily available. In response to that issue, Chu *et al.* [4] proposed the implementation of the modified recursive Maximum Likelihood adaptive filter in 1995, claiming that the adaptive filter is more robust than the EKF. Mendonça *et al.* [5] also proposed an adaptive method for estimating the noise properties in 2007. All these efforts were carried out using manned aircraft.

In the age of the UAV, due to the advances in discrete time system identification techniques and control theories, identifying an accurate model that is coherent with the force equations and kinematic equations seems to be unnecessary. However, discrete time system identification techniques are unable to provide a good insight of an aircraft’s dynamics because the discrete time models are usually physically meaningless, no matter how well they agree with the real flight data. In addition to that, UAVs are usually smaller than manned aircrafts, and fly at lower velocities, and thus are very sensitive to the atmospheric turbulence which will result in a higher noise ratio in the flight data. Due to that, sensor compatibility correction is more important than ever if the dynamic characteristics of an UAV are to be investigated.

Motivated by the discussion above, this paper attempts to derive a procedure in order to perform compatibility correction on the flight data of the SP-80 UAV. The SP-80 UAV [6-9] is an UAV research testbed developed and operated under the Spoonbill UAV Project by the Remotely Piloted Vehicle and Micro-Satellite Laboratory (RMRL) of the National Cheng Kung University (NCKU),

Taiwan. The Spoonbill UAV Project was established to fulfill educational research purposes of which it has brought forth the successful implementation of subspace and prediction-error method identification algorithms in the UAV system identification [7], implementation of the Linear Quadratic Gaussian (LQG) controller to the autopilot system [8] and a successful cross-sea automatic flight demonstration that covered 90.2 km [6]. As part of the continuing researches on the SP-80 UAV, the work presented in this paper intends to bridge the gap between the controller implementation and the understanding of the dynamics of the UAV.

The following sections in this paper are organized as follows: first, Section 2 introduces the Spoonbill UAV system. In particular, the sensor system is described in detail. Section 3 reviews the force equations and kinematic equations. This is followed by Sections 4 and 5 which briefly elaborate the Wiener Type Filter and the RTS smoother. Section 6 presents the flight test data and initial data reconstruction and the next section explains the sensor compatibility correction procedure in detail. Then, the result of correction is discussed in Section 8. Finally, Section 9 draws the conclusions for the paper.

2. The Hardware Architecture

2.1. The Spoonbill UAV System

The Spoonbill UAV system used in this work is designed and operated by the RMRL for educational research purposes. The air vehicle used is designated as the SP-80, which has a high-wing, twin-boom, pusher engine configuration and the power plant is an 80 c.c. two stroke gasoline engine. Figure 1 depicts the air vehicle. The 3.50 m wingspan air vehicle is capable of carrying 3 kg of fuel and it has a maximum endurance of an hour once fully refueled. The fuselage is designed such that it can carry various sensors, an onboard computer, telemetry devices, and a gimballed camera system. So far, the air vehicle is capable of performing automatic flight using a flight controller based on the LQG algorithm, and it automatically flew the longest distance of 90.2 km on 20 October 2009 [6].

Figure 1. The SP-80 UAV.



2.2. The Onboard System

Figure 2 shows the onboard avionics system and Figure 3 illustrates the Spoonbill UAV system architecture. The onboard computer acts as the central processing unit. It is a PC/104 embedded computer produced by Advantech Co., Ltd. The onboard computer is operating under a Windows-XP embedded operating system and the flight control computer program is written using

Borland C++ Builder. The onboard computer collects data from the peripheral sensors through RS-232 protocol via several communication ports (COM Port). In addition to that, the onboard computer also performs bi-directional communication with the Servo Management Board (SMB), and the wireless transceiver. The final COM Port is occupied by the rotation platform of the gimballed camera system. The flight control program operates under various modes depends on the system operator and the flight test objectives. The basic functions of the flight control program are: acquires data from the peripheral sensors, executes the automatic flight control algorithm, and records selected flight data. All of these operations are executed at the rate of 20 Hz.

Figure 2. The onboard system.

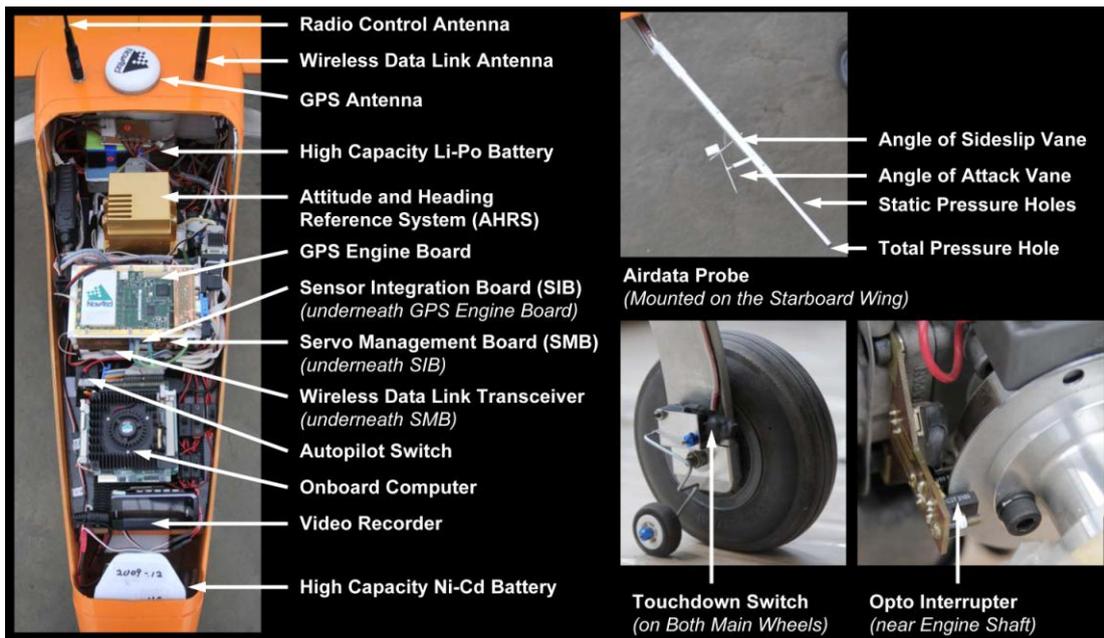
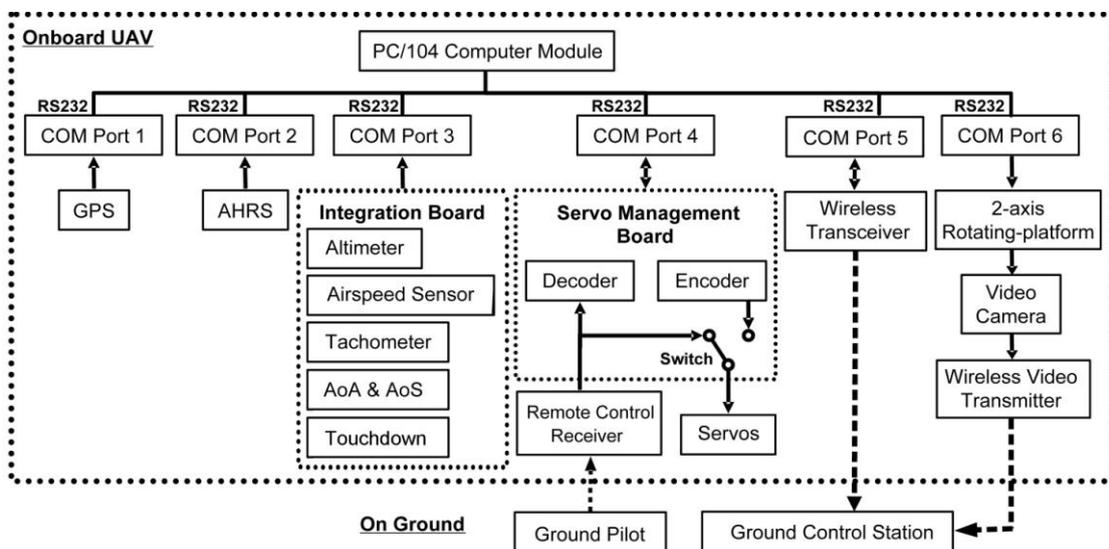


Figure 3. Spoonbill UAV System architecture.

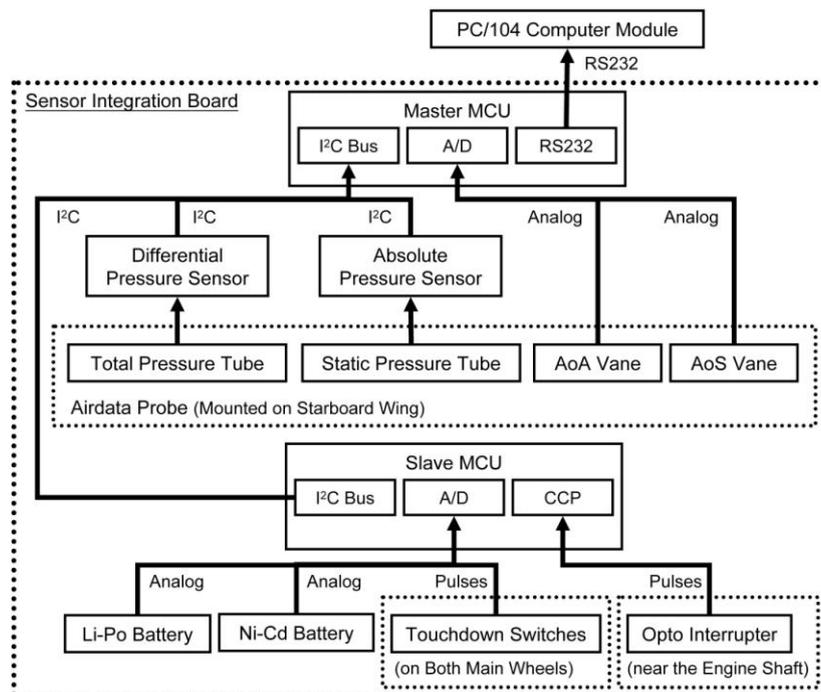


Before any discussion on the peripheral sensors can commence, it is worth mentioning that the GPS receiver, the AHRS, the wireless transceiver, and the video camera system are all off-the-shelf

products, while the Sensor Integration Board (SIB), and the SMB were designed and fabricated in-house by the RMRL. The SMB is a crucial component of an autopilot system for it decodes the control signal from the pilot in Pulse Width Modulation (PWM) form during the manual flight mode, and it encodes PWM signal to control the air vehicle control surface actuators (servo motors) according to the command given by the onboard computer. The SMB is capable of decoding and encoding nine channels of PWM signals at a 20 Hz update rate. The pilot can easily cede flight control authority to the onboard computer by flipping a switch on the remote control radio. Regaining control of the air vehicle from the onboard computer is just as easy.

The peripheral sensors are divided into three major groups. The first group is the GPS. The GPS receiver is the Novatel OEMV-3. Unlike any other GPS receivers built for automobile navigation that normally operating at a maximum of 5 Hz, The Novatel OEM-V3 is capable of providing global position and ground speed at the rate of 20 Hz and is thus consistent with the sampling rate of other onboard sensors. The second group is the AHRS. It is a Crossbow AHRS440. The AHRS measures 3-axis accelerations, angular rates, and attitude angles. The attitude angle measurement is augmented by GPS information from the Novatel OEM-V3. The final group is the SIB. The SIB integrates all other sensors using microcontroller units (MCU). It is designed and fabricated in-house by the RMRL. The MCUs on the SIB are responsible to collect data from the corresponding sensors, perform signal conditioning on the data (such as data filtering), and pass the data to the onboard computer via RS232 protocol. Sensors included in the SIB are the differential air pressure sensor for airspeed measurement, absolute air pressure sensor for barometric altitude measurement, engine speed tachometer, AoA and AoS, and the touchdown sensors. Figure 4 illustrates the SIB architecture.

Figure 4. Sensor Integration Board architecture.



The SIB consists of two MCUs: the Master MCU and the Slave MCU. The Master MCU is the core of the SIB. It controls the data flow within the SIB and sends data to the onboard computer. There are

two sensors built on the SIB: the SenSpecial SCPB-MB0/10D100i2c32667R5 digital differential pressure sensor, and the SenSpecial SCPB-mmHg525/825A100i2c32667R5 digital absolute pressure sensor. Both pressure sensors are connected to the air pressure tubes from the SpaceAge Control 101100-02 airdata probe. The airdata probe is mounted on the starboard wing, as shown in Figure 2. The Master MCU communicates with the pressure sensors via I²C protocol, and calculates the airspeed and barometric altitude using the pressure readings and the following equations.

$$h = h_b + \frac{T_b}{L_b} \left[\left(\frac{P}{P_b} \right)^{\frac{R^* L_b}{g_0 M}} - 1 \right] \quad (1)$$

$$V = \sqrt{\frac{2(P_0 - P)}{\rho}} \quad (2)$$

$$\rho = \frac{P}{RT} \quad (3)$$

$$T = T_b + L_b h \quad (4)$$

Where $g_0 = 9.80665 \text{ m/s}^2$ is the gravitational acceleration at sea level

h is the geopotential altitude in meter

$h_b = 0$ is the sea level altitude

$L_b = -6.5 \times 10^{-3} \text{ K/m}$ is the temperature lapse rate from sea level to altitude 11,000 m

$M = 0.0289644 \text{ kg/mol}$ is the molar mass of Earth's air

P is the static pressure in Newton per square meter (Pascal)

P_0 is the total pressure in Newton per square meter (Pascal)

$P_b = 1.01325 \times 10^5 \text{ N/m}^2$ is the sea level static pressure

$R = 287 \text{ J/(kg} \cdot \text{K)}$ is the specific gas constant

$R^* = 8.31432 \text{ N} \cdot \text{m}/(\text{mol} \cdot \text{K})$ is the universal gas constant

T is the temperature

$T_b = 288.16 \text{ K}$ is the sea level temperature

V is the airspeed in meter per second

ρ is the air density in kilogram per cubic meter

Equation 1 is called the barometric formula introduced by NASA in 1976 [10]. It has since become the standard equation for barometric altitude calculation. Equation 2 is derived from the Bernoulli's equation [11]. From Equation 2, the airspeed of the air vehicle can be determined once the total pressure, static pressure and air density are known. Due to the absence of an air density sensor, the air density is estimated using Equation 3. The temperature needed in Equation 3 is calculated by one of the standard atmosphere formulae shown here as Equation 4 [10]. Equation 4 is only valid from sea level to altitude 11,000 m. The absolute pressure sensor measures P , while the differential pressure sensor measures $(P_0 - P)$.

The airdata probe is also built with two vanes, one for the AoA measurement, and another for the AoS. The vanes are basically two delicate potentiometers. The SIB provides the operating voltage to

the vanes and senses the vanes' movement through the Master MCU 12 bit analog-to-digital converters (A/D).

The onboard avionics are powered by a high capacity Lithium-Polymer (Li-Po) battery and the air vehicle control surface actuators (servo motors) are powered by high capacity Nickel-Cadmium (Ni-Cd) batteries. The Slave MCU reads the battery voltages via a built-in 10 bit A/D. In addition, it reads the pulse signal from the touchdown switches mounted on both main wheels of the air vehicle; such to determine the instant the air vehicle lifts off the ground and touches down. The most important function of the Slave MCU is to calculate the engine rotation speed. The opto interrupter mounted near the engine shaft generates a pulse per engine revolution. The Capture/Compare/PWM (CCP) interrupt of the Slave MCU detects the pulse and calculates the time taken between two pulses before converts it to revolution per minute (rpm). The Slave MCU sends the data string to the Master MCU upon Master MCU's I²C request.

3. Equations of Motions

The air vehicle translation and rotation dynamics are governed by two sets of ordinary differential equations (ODE), namely the force equations and the kinematic equations. Each set of these equations consists of three equations, and each equation governs the motion of one axis. The detailed derivation of the equations has been described well by Nelson [12].

3.1. Force Equations

The force equations as shown in Equation 5 describe the variation of the three body axis velocities. The ODEs are functions of the velocities, accelerations, angular rates, and gravity components:

$$\dot{u} = rv - qw + a_x - g \sin \theta \quad (5a)$$

$$\dot{v} = pw - ru + a_y + g \cos \theta \sin \phi \quad (5b)$$

$$\dot{w} = qu - pv + a_z + g \cos \theta \cos \phi \quad (5c)$$

where: a_x is the acceleration along the body X-axis
 a_y is the acceleration along the body Y-axis
 a_z is the acceleration along the body Z-axis
 $g = 9.81ms^{-2}$ is the gravitational acceleration
 p is the roll rate
 q is the pitch rate
 r is the yaw rate
 u is the velocity along the body X-axis
 v is the velocity along the body Y-axis
 w is the velocity along the body Z-axis

It is also possible to describe the force equation in terms of the variation of airspeed, AoA, and AoS. The airspeed, AoA, and AoS are related to the body axis velocities as in Equation 6. Thus, the force equations can be transformed into Equation 7 that describe the variation of airspeed, AoA, and AoS:

$$V = \sqrt{u^2 + v^2 + w^2} \quad (6a)$$

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \quad (6b)$$

$$\beta = \sin^{-1}\left(\frac{v}{V}\right) \quad (6c)$$

$$\dot{V} = (a_x \cos \alpha + a_z \sin \alpha) \cos \beta + a_y \sin \beta + g (\cos \theta \cos \phi \sin \alpha \cos \beta + \cos \theta \sin \phi \sin \beta - \sin \theta \cos \alpha \cos \beta) \quad (7a)$$

$$\dot{\alpha} = \frac{1}{V \cos \beta} \{a_z \cos \alpha - a_x \sin \alpha + g (\cos \theta \cos \phi \cos \alpha + \sin \theta \sin \alpha)\} + q - \tan \beta (p \cos \alpha + r \sin \alpha) \quad (7b)$$

$$\dot{\beta} = \frac{1}{V} \left\{ a_y \cos \beta - (a_x \cos \alpha + a_z \sin \alpha) \sin \beta + g [\cos \theta \sin \phi \cos \beta + (\sin \theta \cos \alpha - \cos \theta \cos \phi \sin \alpha) \sin \beta] \right\} + p \sin \alpha - r \cos \alpha \quad (7c)$$

where: V is the airspeed

α is the AoA

β is the AoS

As a matter of fact, even though Equation 7 is much more complicated compared to Equation 5, it is more intuitive to use Equation 7 rather than Equation 5 because the airspeed, AoA, and AoS are normally measured by the onboard instruments instead of using the body axis velocities.

3.2. Kinematic Equations

The kinematic equations as shown in Equation 8 describe the variation of the three attitude angles, namely the roll angle, pitch angle, and yaw angle. The ODEs are functions of the attitude angles, and angular rates:

$$\dot{\phi} = p + \tan \theta (q \sin \phi + r \cos \phi) \quad (8a)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (8b)$$

$$\dot{\psi} = \frac{q \sin \phi + r \cos \phi}{\cos \theta} \quad (8c)$$

where: ϕ is the roll angle

θ is the pitch angle

ψ is the yaw angle

4. Wiener Type Filter

The Wiener Type Filter [13] is an optimal filter in the frequency domain. Equation 9(a) indicates that the measurement is a summation of the filtered/true signal and noise. In order to simplify the transformation to the frequency domain, the measurement is modified using Equation 9(b). Equation 9(b) is an odd function with its end point discontinuity removed. Such function allows the Fourier sine series expansion. The Fourier sine series coefficients are given by Equation 9(c):

$$z_k = y_k + n_k, \quad k = 1, 2, \dots, N \quad (9a)$$

$$g_k = z_k + z_1 - (k-1) \left[\frac{z_N + z_1}{N-1} \right], \quad k = 1, 2, \dots, N \quad (9b)$$

$$b_l = \frac{2}{N-1} \sum_{i=2}^{N-1} g_i \sin \left[\frac{l\pi(k-1)}{N-1} \right], \quad i = 1, 2, \dots, N-1 \quad (9c)$$

where: b_l is the Fourier sine series coefficient at l^{th} frequency index

g_k is the modified measurement at k^{th} time step

N is the data size

n_k is the noise at k^{th} time step

y_k is the filtered/true at k^{th} time step

z_k is the measurement at k^{th} time step

The next step is to define the true signal model and noise model. It is important to know that the filter consists of weighting indices range from 0 to 1, where 0 indicates total rejection and 1 indicates otherwise. Thus, it is ideal when the filter equals to 0.5 as the frequency index then corresponds to the cutoff frequency. Plus, as discussed by Morelli, it is best to describe the true signal model being proportional to l^{-3} . Given the form of the Wiener Type Filter as in Equation 10(a), yields the true signal model and noise model as shown in Equation 10(b,c). Equation 10(d) shows how the frequency index being related to the frequency. Similarly, Equation 10(e) determines the frequency index corresponds to the desired cutoff frequency. The filter is applied in Equation 10(f) to obtain the filtered modified measurement. Finally, Equation 10(g) reconstructs the filtered measurement:

$$\Phi_l = \frac{Y_l^2}{Y_l^2 + N_l^2}, \quad 0 \leq l \leq N-1 \quad (10a)$$

$$Y_l = l_c^3 l^{-3} \quad (10b)$$

$$N_l = 1 \quad (10c)$$

$$f_l = \frac{l}{2(N-1)\Delta t} \quad (10d)$$

$$l_c = 2f_l(N-1)\Delta t, \quad \text{if } f_l \leq f_{c,d} < f_{l+1} \quad (10e)$$

$$\tilde{g}_{s,k} = \sum_{l=1}^{N-1} \Phi_l b_l \sin \left[\frac{l\pi(k-1)}{N-1} \right], \quad k = 1, 2, \dots, N \quad (10f)$$

$$y_k = \tilde{g}_{s,k} + z_1 + (k-1) \frac{z_N - z_1}{N-1}, k=1, 2, \dots, N \quad (10g)$$

where: f_l is the frequency at l^{th} frequency index

$f_{c,d}$ is the desired cutoff frequency

$\tilde{g}_{s,k}$ is the filtered modified measurement at k^{th} time step

l_c is the frequency index corresponds to the cutoff frequency

N_l is the noise model in discrete frequency domain at l^{th} frequency index

Y_l is the true signal model in discrete frequency domain at l^{th} frequency index

Φ_l is the filter at l^{th} frequency index

5. Rauch-Tung-Striebel Smoother

The RTS Smoother [14-16] is a two pass algorithm. The forward pass is a standard EKF [15] while the backward recursion is introduced to reduce the inherent bias in the EKF estimates.

5.1. Forward Pass (Extended Kalman Filter)

The EKF is literally one of the most widely used state estimation methods due to its simplicity. The EKF is derived from the standard Kalman Filter. The Kalman Filter is derived based on linear systems, but the EKF works for nonlinear systems by performing linearization about the current estimates.

Over the years, different forms of EKF have been proposed, namely the continuous time EKF, discrete time EKF, and the continuous-discrete EKF. Since the force equations and kinematic equations are continuous time equations and the flight test measurements are discrete, the continuous-discrete EKF is used for the work presented in this paper. The continuous-discrete EKF is presented here as Equations 11–12. Often, the difficulty of implementing the EKF is mainly due to the lack of *a priori* knowledge of the noise covariance.

Given a nonlinear continuous time dynamic system with discrete time measurement, of which the measurement and process noise are Gaussian distributed:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(t), \mathbf{w}(t) \sim N(0, \mathbf{Q}(t)) \quad (11a)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \mathbf{v}_k \sim N(0, \mathbf{R}_k) \quad (11b)$$

where: $f(\mathbf{x}(t), \mathbf{u}(t))$ is the nonlinear ordinary differential equations

$h(\mathbf{x}_k)$ is the measurement equations

k is the number of time step

\mathbf{Q} is the process noise covariance matrix

\mathbf{R} is the measurement noise covariance matrix

\mathbf{u} is the input vector

\mathbf{v} is the Gaussian distributed measurement noise vector

\mathbf{w} is the Gaussian distributed process noise vector

\mathbf{x} is the state vector

\mathbf{z} is the measurement vector

The prediction phase:

$$\hat{\mathbf{x}}(t) = \int_t^{t+\Delta t} f(\hat{\mathbf{x}}(t), \mathbf{u}(t)) dt, \quad \mathbf{F}(t) = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}(t), \mathbf{u}(t)} \quad (12a)$$

$$\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(t)^T + \mathbf{Q}(t) \quad (12b)$$

$$\hat{\mathbf{z}}_k = h(\hat{\mathbf{x}}_{k|k-1}), \quad \mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}} \quad (12c)$$

The updating phase:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (12d)$$

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k \quad (12e)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad (12f)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \quad (12g)$$

where: $\mathbf{P}(t_k) = \mathbf{P}_{k|k-1}$

$$\mathbf{P}(t_{k-1}) = \mathbf{P}_{k-1|k-1}$$

$$\hat{\mathbf{x}}(t_k) = \hat{\mathbf{x}}_{k|k-1}$$

$$\hat{\mathbf{x}}(t_{k-1}) = \hat{\mathbf{x}}_{k-1|k-1}$$

\mathbf{F} is the Jacobian of the nonlinear ordinary differential equations

\mathbf{H} is the Jacobian of the measurement equations

\mathbf{K} is the gain matrix

\mathbf{P} is the state error covariance matrix

$\hat{\mathbf{x}}$ is the estimated state vector

$\tilde{\mathbf{y}}$ is the measurement residual

$\hat{\mathbf{z}}$ is the estimated measurement

Equation 12(b) is a continuous time Riccati equation. In order to solve the equation, the discrete state transition matrix can be determined using Equation 12(h). With the state transition matrix available, Equation 12(i) provides the solution of the state error covariance matrix. Note that Equation 12(h) is applied under the assumption that \mathbf{F} is a constant from t_{k-1} to t_k :

$$\Phi_{k-1} = e^{\mathbf{F}(t)\Delta t} \quad (12h)$$

$$\mathbf{P}_{k|k-1} = \Phi_{k-1} \mathbf{P}_{k-1|k-1} \Phi_{k-1}^T + \mathbf{Q}_{k-1} \quad (12i)$$

where Φ is the state transition matrix

It is important to know that the EKF needs predefined \mathbf{P}_{00} , \mathbf{R} , \mathbf{Q} , and $\hat{\mathbf{x}}_{00}$ for initialization. It is acceptable to assign \mathbf{P}_{00} arbitrarily, however, the values must be large enough to allow good tracking of the parameters. If the states are measured, $\hat{\mathbf{x}}_{00}$ can be specified by taking the average of the first few data points.

5.2. Backward Recursion

The backward recursion works backwards in time. The combination of a forward pass estimator (EKF) and a backward recursion is considered to have utilized all available information [15]. Thus the scheme is capable of performing better estimations than the EKF. Equation 13 summarizes the backward recursion of the RTS smoother:

$$\mathbf{A}_k = \mathbf{P}_{k|k} \mathbf{\Phi}_k^T \mathbf{P}_{k+1|k}^{-1} \quad (13a)$$

$$\hat{\mathbf{x}}_{k|n} = \hat{\mathbf{x}}_{k|k} + \mathbf{A}_k \left(\hat{\mathbf{x}}_{k+1|n} - \hat{\mathbf{x}}_{k+1|k} \right), \quad k = N-1, \dots, 0 \quad (13b)$$

$$\mathbf{P}_{k|n} = \mathbf{P}_{k|k} + \mathbf{A}_k \left[\mathbf{P}_{k+1|n} - \mathbf{P}_{k+1|k} \right] \mathbf{A}_k^T \quad (13c)$$

where: \mathbf{A} is the smoother gain matrix
 N is the final time step
 $\mathbf{P}_{k|n}$ is the corresponding state error covariance matrix
 $\hat{\mathbf{x}}_{k|n}$ is the smoothed states of k^{th} time step

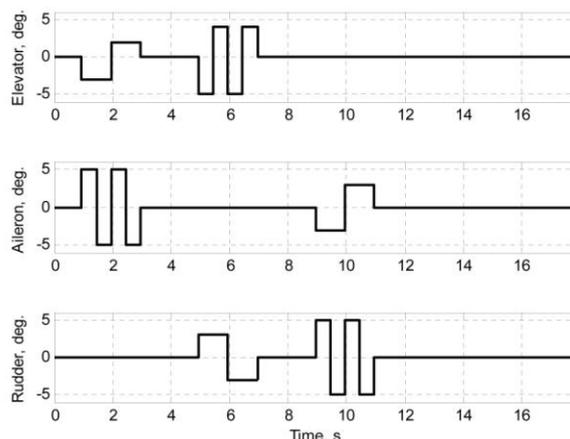
6. Flight Tests

6.1. The Flight Maneuver

The flight test data were collected through a series of flight tests. In the flight data acquisition process, it is desirable to maneuver the air vehicle such that the flight data contain sufficient information to represent the dynamics of the air vehicle. The maneuvers chosen to fulfill the objective of this work were a multiple-input design. Due to its simplicity, time-skewed doublet inputs like those used in 2008 by Lee *et al.* [7,8] are always a favorable choice, but in order to effectively excite the sufficient dynamic information, including the coupling of the longitudinal and lateral motion, the orthogonal square-wave inputs [17] were applied. The orthogonal square-wave inputs are depicted in Figure 5.

As shown in Figure 5, at any instant, two control surfaces are deflected simultaneously. One is deflected at half the rate of another. The simultaneous deflections are meant to excite couple dynamics, while the separate deflection rates are meant to distinguish the corresponding control surfaces' effects on the air vehicle motion.

The initial condition of the designed flight maneuver has to be trimmed flight condition. Due to the complexity of the maneuver and the difficulty for the ground pilot to ensure a trimmed flight initial condition, the maneuver was executed by the onboard computer. The ground pilot flew the SP-80 UAV to a desired altitude and remained straight and level flight before switching the control authority to the onboard computer. Once switched, the onboard computer maintained the air vehicle in trimmed straight and level flight for 5 seconds. After that, the designed maneuver was executed and the ground pilot switched it back to manual flight no less than 5 seconds after the maneuver. The desired altitude was about 300 m above ground. Since the air vehicle remained at open-loop control during the maneuver, the 300 m altitude gave the ground pilot enough time to react to any unexpected outcome while the air vehicle remained in the pilot's eyesight.

Figure 5. Orthogonal square-wave inputs.

It is desirable to perform the flight test under absolutely windless conditions, but this is utterly impossible. Thus, the flight tests were conducted in nearly windless environment and the maneuvers was executed in both head wind and tail wind directions to minimize the crosswind effect.

6.2. The Flight Data and Reconstruction

The flight data reconstruction is accomplished by implementing the 4th order Runge-Kutta method. The accelerations and angular rates are the inputs; the airdata, and attitude angles are the outputs. The 4th order Runge-Kutta method is summarized as Equation 14:

$$\begin{aligned}
 k_1 &= h \cdot f(\mathbf{x}(t), \mathbf{u}(t)) \\
 k_2 &= h \cdot f\left(\left[\mathbf{x}(t) + \frac{k_1}{2}\right], \mathbf{u}(t)\right) \\
 k_3 &= h \cdot f\left(\left[\mathbf{x}(t) + \frac{k_2}{2}\right], \mathbf{u}(t)\right) \\
 k_4 &= h \cdot f\left([\mathbf{x}(t) + k_3], \mathbf{u}(t)\right) \\
 \mathbf{x}(t+h) &= \mathbf{x}(t) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}
 \end{aligned} \tag{14}$$

where h is the time step size

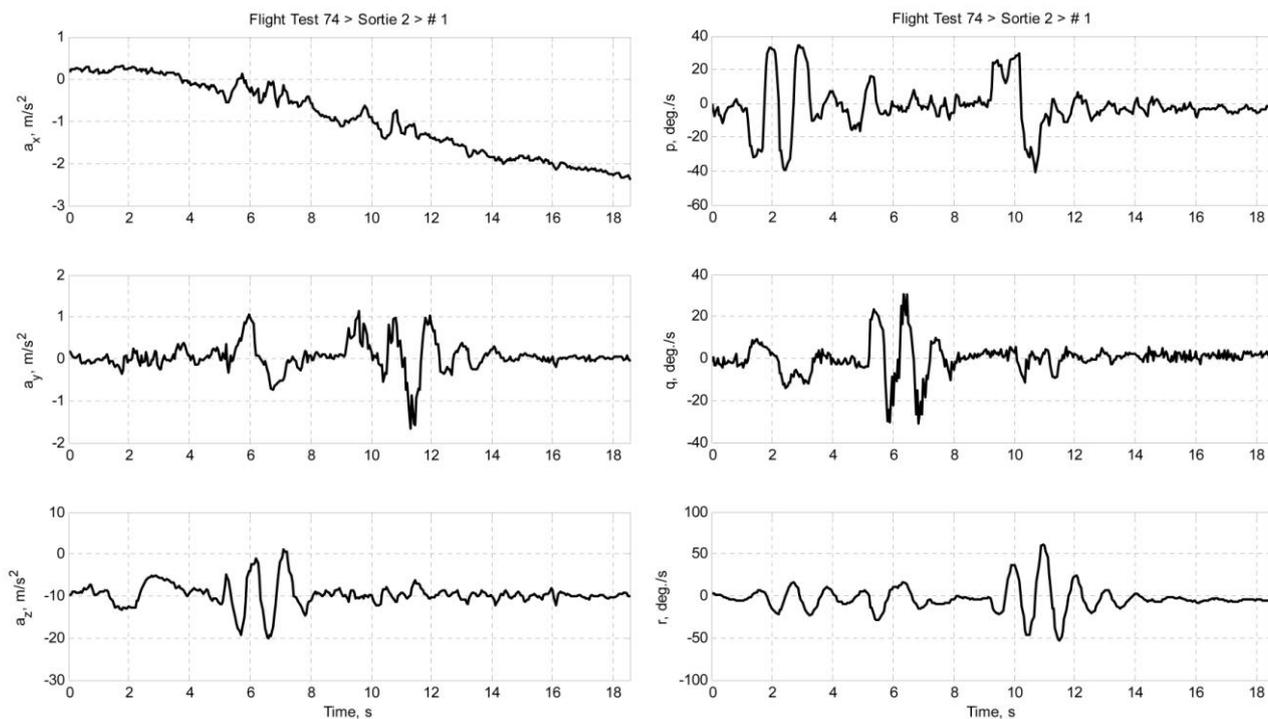
Multiple sets of flight data were collected and each data set is designated in this paper in the following format: *ftFFsrtSSrunRR*, where FF is the flight test number of the Spoonbill UAV project, SS is the sortie number, and RR is the number of maneuver in that particular sortie. This paper presents two data sets as examples: *ft74srt02run01* and *ft74srt02run05*. Data of *ft74srt02run01* is depicted in Figure 6, and Figure 7 corresponds to *ft74srt02run05*. Both data sets show similar trends. The reconstructed attitude angles demonstrate very good compatibility while the airdata compatibility is poor.

From Equation 7, it is legitimate to infer that the airspeed, AoA, and AoS variations are dominated by X-axis acceleration, Z-axis acceleration, and Y-axis acceleration, respectively. The reconstructed

airspeed is over estimated in both data sets. It is obvious that the reconstructed airspeeds are increasing at a higher rate compared to the measured airspeed, which suggest the presence of a bias on the X-axis acceleration. Fluctuations are observed on the measured airspeeds at the 9th to 12th second, but not on the reconstructed airspeeds. That could be the result of instrumentation errors of the airdata probe due to rapid yawing motion of the air vehicle. The AoA and AoS demonstrate slightly better compatibilities compared to the airspeed. The reconstructed data seem to follow the trend of the measured data. This suggests that the biases of Y-axis acceleration and Z-axis acceleration are less significant. Thus, the instrumentation error, measurement noises, and atmospheric turbulent are the major causes of the AoA and AoS incompatibilities.

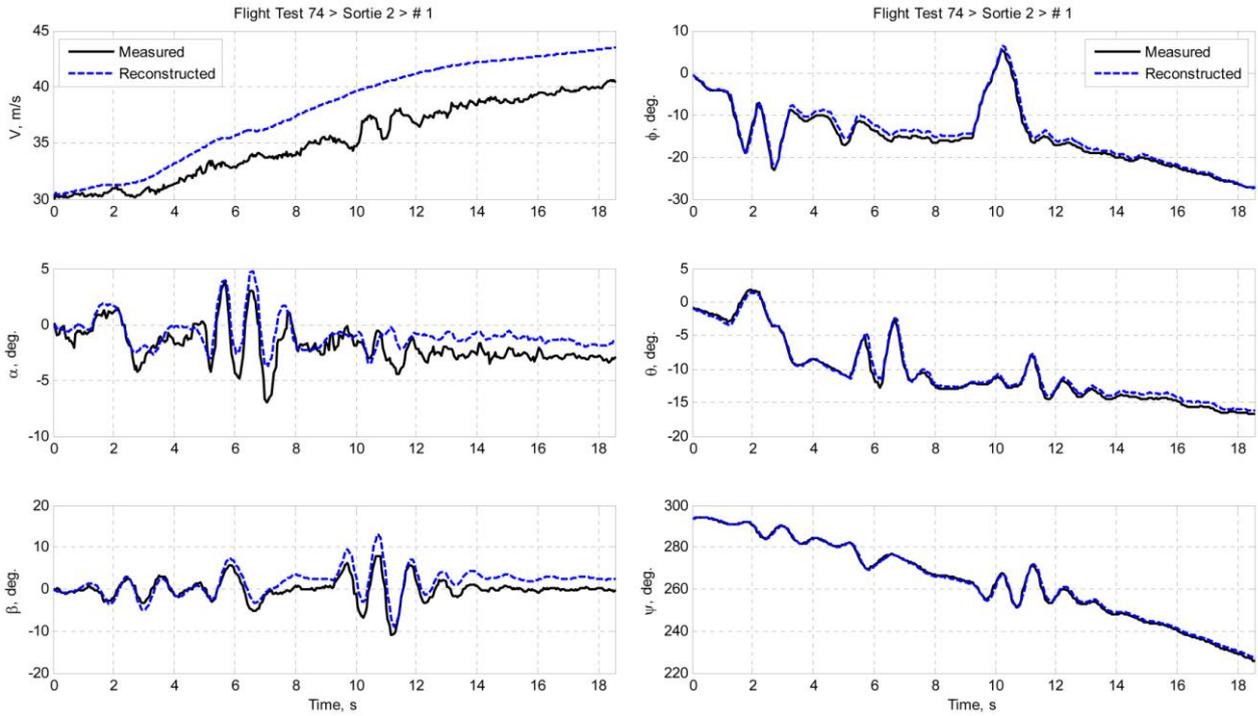
Similar inferences were made on Equation 8. The roll angle, pitch angle, and yaw angle variations are dominated by the roll rate, pitch rate, and yaw rate, respectively. It is well understood that the attitude angles are estimated through complex algorithms using the angular rates, and accelerations [18]. The principle idea is that the attitude angles are estimated through integration of the angular rates, and the accelerations prevent the drifting of the estimation. That explains the high compatibilities of the attitude angles. Since the attitude angles estimation is augmented by GPS data [19,20], it is more likely that the attitude angles are relatively accurate and noises of the angular rates are the reason of the slight incompatibilities.

Figure 6. (a) Accelerations and angular rates of *ft74srt02run01*. (b) Airdata and attitude angles of *ft74srt02run01*.



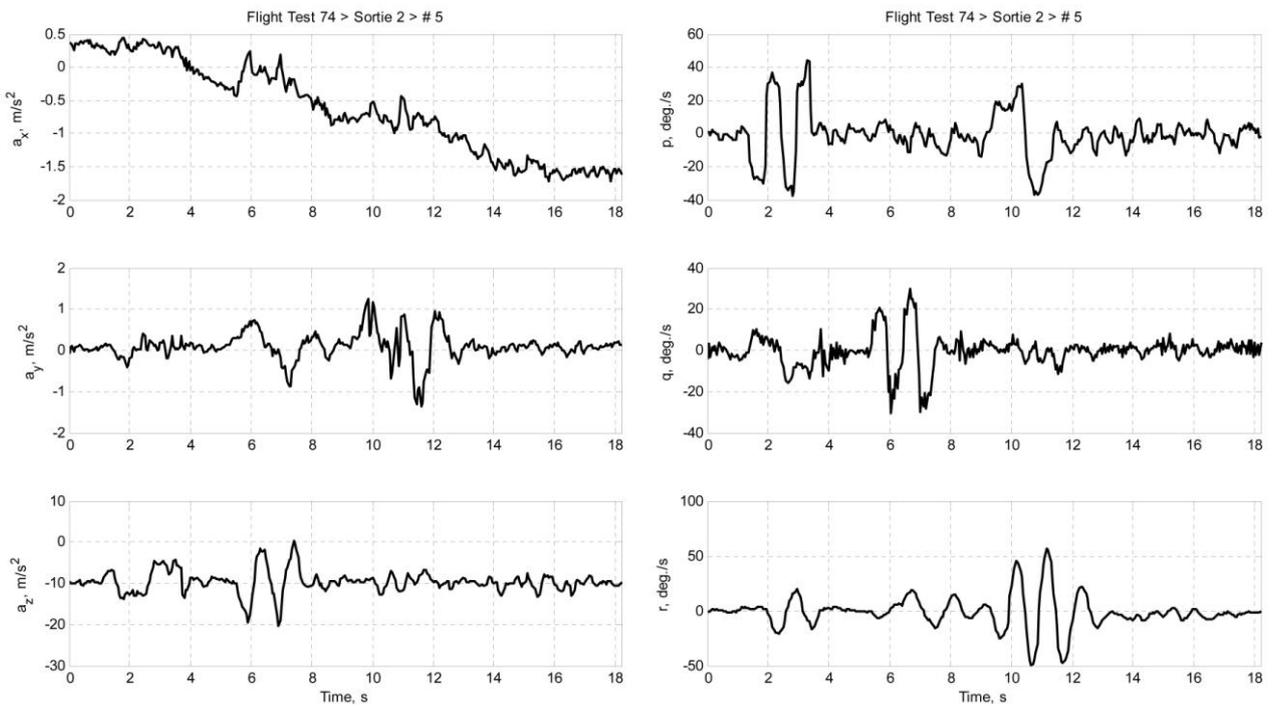
(a)

Figure 6. Cont.



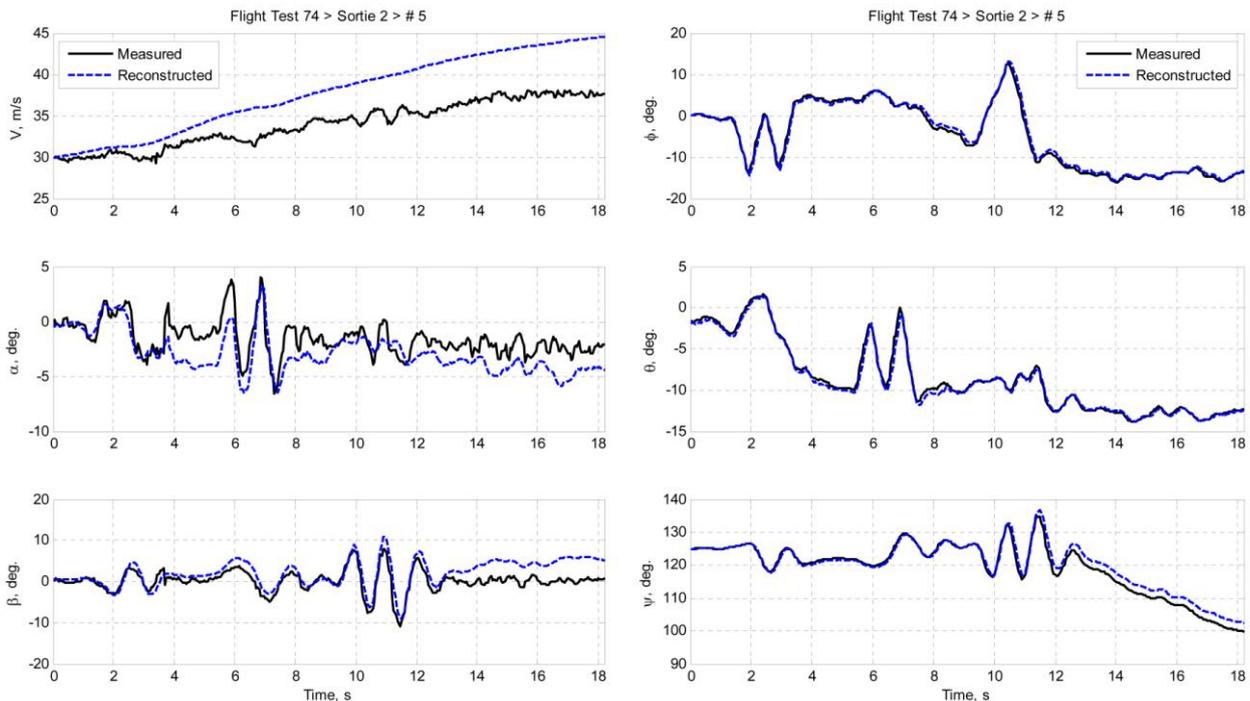
(b)

Figure 7. (a) Accelerations and angular rates of *ft74srt02run05*. (b) Airdata and attitude angles of *ft74srt02run05*.



(a)

Figure 7. Cont.



(b)

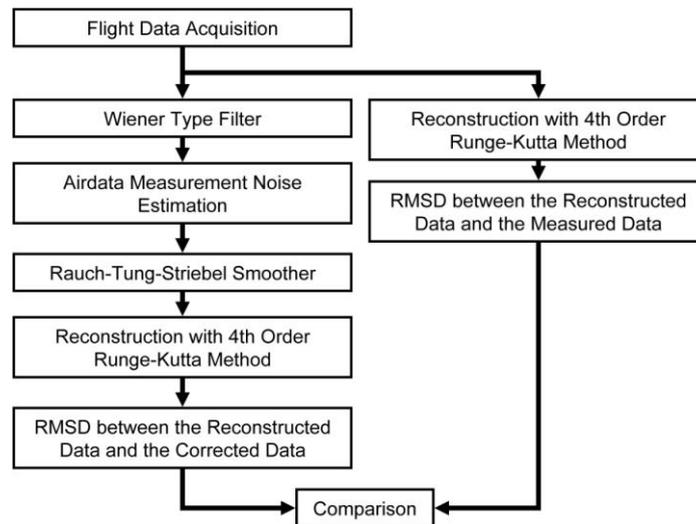
7. Sensor Data Compatibility Correction

7.1. The Compatibility Correction Procedure

The sensor compatibility correction procedure as illustrated in Figure 8 was designed under several assumptions:

- The AHRS body axes and the air vehicle body axes coincide.
- The airdata probe is located near to the center of gravity such that position correction is unnecessary.
- Since the AHRS is not exposed to the environment, the AHRS readings (accelerations, angular rates, and attitude angles) are less affected by atmospheric turbulence, and the noises (not the biases) are high frequency noises that can be effectively filtered by a frequency filter.
- Due to the high compatibility of the attitude angles, the biases of the angular rates are negligible.
- The noises or errors of different variables are not inter-correlated.
- The noises distributions are Gaussian.

As seen in the right portion of Figure 8, the flight data undergo reconstruction using the 4th order Runge-Kutta Method. The reconstructed results of *ft74srt02run01* and *ft74srt02run05* are presented in the previous section as Figure 6 and Figure 7. The RMSD shown here as Equation 15 is then implemented to quantify the quality of compatibility. The smaller the RMSD is, the better the compatibility is.

Figure 8. The compatibility correction procedure.

The RMSD between the measured and reconstructed airdata and attitude angles are calculated. This is the RMSD before the correction:

$$RMSD(\gamma_1, \gamma_2) = \sqrt{\frac{\sum_{i=1}^{n_d} (\gamma_{1,i} - \gamma_{2,i})^2}{n_d}} \quad (15)$$

where: n_d is the data size

γ_1 is the first data

γ_2 is the second data

$\gamma_{1,i}$ is the i^{th} element of the first data

$\gamma_{2,i}$ is the i^{th} element of the second data

The left portion of Figure 8 illustrates the correction process. The RTS smoother is the core of the whole process. The flight data (airspeed, AoA, AoS, attitude angles, angular rates, and accelerations) is first passed through a Wiener Type Filter to suppress the high frequency noises of the flight data. It then followed by the airdata measurement noise estimation that will be discussed in depth in Section 7.3. RTS is implemented after that to correct the flight data such that the data is compatible to the force equations and kinematic equations. The corrected accelerations and angular rates are then used as inputs to perform airdata and attitude angle reconstruction. If successful, the corrected and the reconstructed airdata and attitude angles should show a high degree of similarity. After that, the RMSD between the corrected and the reconstructed outputs (airdata and attitude angles) are calculated. This is the RMSD after the correction. Finally the RMSDs before and after the correction are compared to demonstrate the sensor data compatibility improvement. For the work presented in this paper, the RTS was executed with 0.05 s sampling time, which is consistent with the 20 Hz sampling rate of the onboard system.

7.2. RTS Problem Formulation

Under the assumptions discussed in the previous section, the RTS problem formulation is as shown in Equations 16–18. Equation 16 shows the relation between the measured accelerations and the corresponding biases. As in Equation 17, the biases are formulated as the state variables such that the RTS smoother could estimate them. The airdata and the attitude angles are part of the state variables and they are also the measurement. The accelerations and angular rates are the inputs. While the force equations and kinematic equations serve as the ordinary differential equations of corresponding states, the acceleration biases are assumed constant as shown in Equation 18:

$$a_{x_M} = a_x + b_{a_x} \quad (16a)$$

$$a_{y_M} = a_y + b_{a_y} \quad (16b)$$

$$a_{z_M} = a_z + b_{a_z} \quad (16c)$$

where: a_{x_M} is the measured X-axis acceleration
 a_{y_M} is the measured Y-axis acceleration
 a_{z_M} is the measured Z-axis acceleration
 b_{a_x} is the X-axis acceleration measurement bias
 b_{a_y} is the Y-axis acceleration measurement bias
 b_{a_z} is the Z-axis acceleration measurement bias

$$\mathbf{x} = [V \quad \alpha \quad \beta \quad \phi \quad \theta \quad \psi \quad \mathbf{b}]^T, \quad \mathbf{b} = [b_{a_x} \quad b_{a_y} \quad b_{a_z}]^T \quad (17a)$$

$$\mathbf{z} = [V \quad \alpha \quad \beta \quad \phi \quad \theta \quad \psi]^T \quad (17b)$$

$$\mathbf{u} = [a_{x_M} \quad a_{y_M} \quad a_{z_M} \quad p \quad q \quad r]^T \quad (17c)$$

where \mathbf{b} is the acceleration measurement bias vector:

$$\dot{\mathbf{b}} = \mathbf{0}_{3 \times 1} \quad (18)$$

7.3. Measurement Noise & Process Noise Covariance Matrix Estimation

It was assumed that the noises are not inter-correlated in order to simplify the noise covariance matrices estimation. Hence, the task is to estimate the diagonal terms of the matrices, namely the variances. The measurement noise variances of the airdata are estimated from the speeds measured by the GPS. Unlike the airdata, the GPS speed measurements are less noisy. That is because the GPS doesn't measure speed by interacting with the relative airflow like the airdata probe does. In addition, the atmospheric turbulence is the major contributor to the airdata noise or error. One might argue that it is impossible to accurately estimate the airdata measurement noise variances from the GPS speed measurements because the airdata are relative to the wind, but the GPS measurements are relative to the ground. But it is later demonstrated that the GPS speed measurements are sufficient to fulfill the purpose.

The GPS speed measurement would have to go through a series of coordinate transformations before any noise variances estimation is possible. Equation 19 transforms the horizontal ground speed and ascent rate measured by the GPS to the North-East-Down (NED) local horizontal coordinate frame. The NED speed components are then transformed to the air vehicle body-axis using Equation 20. After that, the body-axis speeds are substituted into Equation 21 to obtain estimations of total speed, AoA, and AoS. It is essential to take note that these estimations are not meant to be accurate, but since they are far much less sensitive to atmospheric turbulent, these estimations provide a good baseline reference for the airdata measurement noise variances estimation:

$$V_{N, gps} = V_{hor, gps} \cos \psi \quad (19a)$$

$$V_{E, gps} = V_{hor, gps} \sin \psi \quad (19b)$$

$$V_{D, gps} = -V_{U, gps} \quad (19c)$$

where: $V_{hor, gps}$ is the horizontal speed given by GPS

$V_{E, gps}$ is the ground speed to the East

$V_{N, gps}$ is the ground speed to the North

$V_{U, gps}$ is the ascent rate given by GPS

$V_{D, gps}$ is the descent rate

$$\begin{bmatrix} u_{gps} \\ v_{gps} \\ w_{gps} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{N, gps} \\ V_{E, gps} \\ V_{D, gps} \end{bmatrix} \quad (20)$$

where: u_{gps} is the body X-axis speed estimated from GPS measured speeds

v_{gps} is the body Y-axis speed estimated from GPS measured speeds

w_{gps} is the body Z-axis speed estimated from GPS measured speeds

$$V_{gps} = \sqrt{u_{gps}^2 + v_{gps}^2 + w_{gps}^2} \quad (21a)$$

$$\alpha_{gps} = \tan^{-1} \left(\frac{w_{gps}}{u_{gps}} \right) \quad (21b)$$

$$\beta_{gps} = \tan^{-1} \left(\frac{v_{gps}}{V_{gps}} \right) \quad (21c)$$

where: V_{gps} is the GPS total speed

α_{gps} is the GPS estimated AoA

β_{gps} is the GPS estimated AoS

The airdata estimations given by Equation 21 serve as reference for the airdata measurement noise estimations. The most intuitive way of estimating the noise is determining the residual between the airdata estimations with the airdata measurements. However, the airdata estimations come with inherent biases due to the difference between airspeed and ground speed. Instead, the measurement noise estimations are derived based on the deviation of the airdata at any particular time with their

local expected values. The idea is as shown in Equation 22. Take Equation 22(a) as an example, the first portion of the equation tells how far is the k^{th} airspeed measurement deviate from its local expected value. The local expected value is the mean of airspeed measurements from $(k - m)^{\text{th}}$ to $(k + m)^{\text{th}}$ time steps. Similarly, in the second portion of the equation, the deviation of the k^{th} GPS total speed from its expected value is calculated. The difference between these two deviations gives the estimated noise of the k^{th} airspeed measurement. One can control the local expected values by changing m . If the air vehicle is highly dynamic, a smaller m gives more legitimate local expected values:

$$\varepsilon_{V,k} = \left[V_k - \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} V_i \right] - \left[V_{gps,k} - \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} V_{gps,i} \right] \quad (22a)$$

$$\varepsilon_{\alpha,k} = \left[\alpha_k - \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} \alpha_i \right] - \left[\alpha_{gps,k} - \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} \alpha_{gps,i} \right] \quad (22b)$$

$$\varepsilon_{\beta,k} = \left[\beta_k - \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} \beta_i \right] - \left[\beta_{gps,k} - \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} \beta_{gps,i} \right] \quad (22c)$$

where: ε is the estimated noise/error

m is the design parameter

Once the airdata measurement noises are estimated using Equation 22, the noise variances can be calculated. Using the same idea of data regionality, the noise variances are calculated by taking the nearby data points into account. The equations are presented as Equation 23. Taking Equation 23(a) as an example, it is a standard variance equation; the k^{th} airspeed measurement noise variance is calculated by accounting a sample with sample size of $(2m + 1)$. The larger the m is, the larger the sample size is, and thus the less intense the variation of the noise variance is. Figures 9 and 10 depict the estimated airdata measurement noise variance of *ft74srt02run01* and *ft74srt02run05* at $m = 5$:

$$\sigma_{\varepsilon_{V,k}}^2 = \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} (\varepsilon_{V,i} - \bar{\varepsilon}_V)^2, \quad \bar{\varepsilon}_V = \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} (\varepsilon_{V,i}) \quad (23a)$$

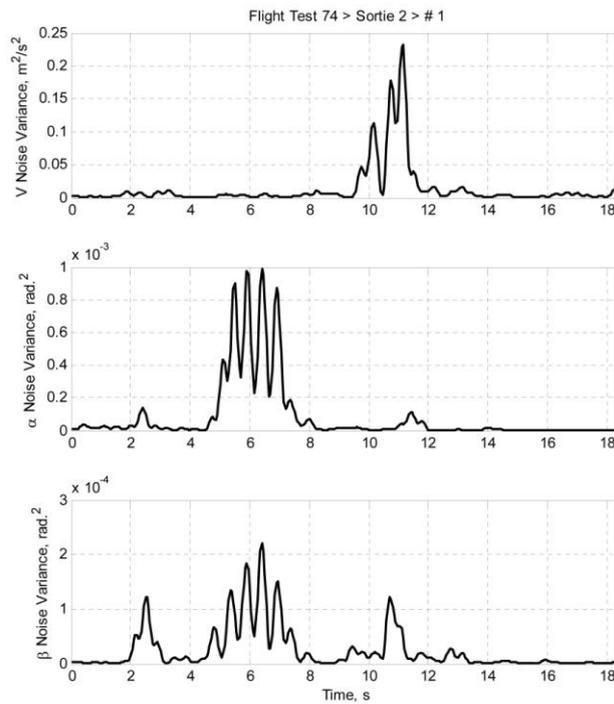
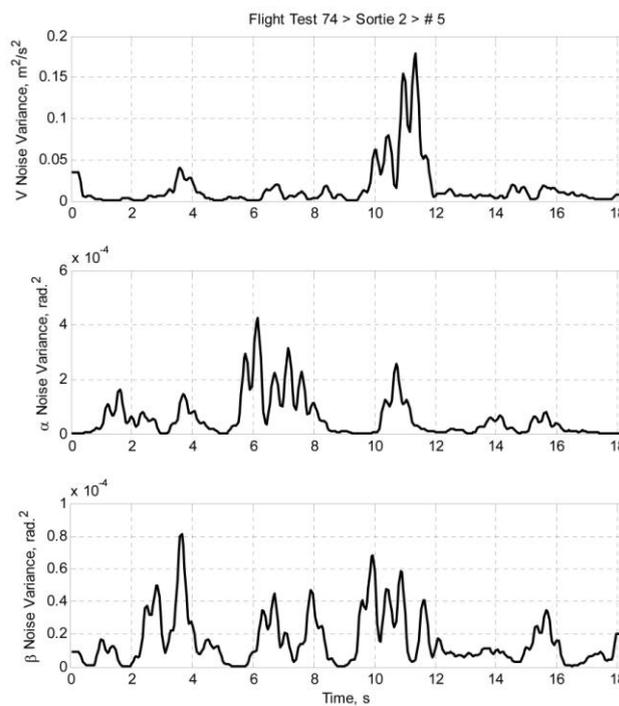
$$\sigma_{\varepsilon_{\alpha,k}}^2 = \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} (\varepsilon_{\alpha,i} - \bar{\varepsilon}_{\alpha})^2, \quad \bar{\varepsilon}_{\alpha} = \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} (\varepsilon_{\alpha,i}) \quad (23b)$$

$$\sigma_{\varepsilon_{\beta,k}}^2 = \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} (\varepsilon_{\beta,i} - \bar{\varepsilon}_{\beta})^2, \quad \bar{\varepsilon}_{\beta} = \frac{1}{2m+1} \sum_{i=k-m}^{2m+1} (\varepsilon_{\beta,i}) \quad (23c)$$

where: $\sigma_{\varepsilon_V}^2$ is the estimated error variance of airspeed

$\sigma_{\varepsilon_{\alpha}}^2$ is the estimated error variance of AoA

$\sigma_{\varepsilon_{\beta}}^2$ is the estimated error variance of AoS

Figure 9. Airdata noise variance of *ft74srt02run01*.**Figure 10.** Airdata noise variance of *ft74srt02run05*.

Under the assumptions given in Section 7.1, the measurement noise covariance is a diagonal matrix and the attitude angles measurement noise variances are zeros after the implementation of the Wiener Type Filter. Hence, the measurement noise covariance matrix is in the following form:

$$\mathbf{R}_k = \begin{bmatrix} \sigma_{\varepsilon_V,k}^2 & 0 & 0 & \\ 0 & \sigma_{\varepsilon_\alpha,k}^2 & 0 & \mathbf{0}_{3,3} \\ 0 & 0 & \sigma_{\varepsilon_\beta,k}^2 & \\ & \mathbf{0}_{3,3} & & \mathbf{0}_{3,3} \end{bmatrix} \quad (24)$$

On the other hand, the process noise is mainly contributed by the accelerations and the angular rates. If the second assumption in Section 7.1 is valid, the process noise covariance matrix shall be a zero matrix since the data has already been filtered by the Wiener Type Filter. However, a zero process noise covariance matrix will lead the RTS smoother to a singularity. Thus, the process noise covariance matrix was chosen to be a diagonal matrix with very small diagonal terms as shown in Equation 25:

$$\mathbf{Q}_k = \mathbf{I} \cdot 1 \times 10^{-5} \quad (25)$$

8. Results and Discussion

Figures 11 and 12 present the results of the sensor compatibility correction of *ft74srt02run01* and *ft74srt02run05*, respectively. The figures contain correction results using the EKF (the RTS forward pass), the RTS (the RTS forward pass and backward recursion), and the measured data (after Wiener Type Filtering at cutoff frequency of 2 Hz). As shown in Figure 11(a) and Figure 12(a), the EKF took some time before the acceleration biases converged, and the biases remain constant in the backward recursion. This is due to the inherent characteristics of the RTS backward recursion whereby it is unable to update the estimations with zero change rates. However, the biases are meant to be constants, and as long as the EKF provide converged estimations, the biases are valid and the RTS backward recursion helps applying the converged biases to all data points.

High degree of similarities is demonstrated on the angular rates and attitude angles. Comparing Figure 6 and Figure 11, and Figure 7 and Figure 12, the compatibilities of the attitude angles improved slightly. The EKF and RTS have done little correction on the attitude angles; the result is mainly due to the high frequency noises suppression by the Wiener Type Filter. This is coherent with the assumption that the AHRS measurement contains only high frequency noises that can be effectively filtered by a frequency filter, and the biases of the angular rates are negligible. Also demonstrated is the significant improvement of the airdata compatibility. The reconstructed airspeed tracks the corrected airspeed with good consistency. The AoA and AoS also show significant improvement. One of the major reasons of the significant improvement of the airspeed compatibility is that the RTS has successfully identified a significant bias on the X-axis acceleration as suggested in Section 6.2. The improvements on the AoA and AoS compatibilities are due to the noise reduction on the angular rates and the biases estimation on the corresponding accelerations.

Figure 11. (a) Accelerations and angular rates of *ft74srt02run01* before and after correction. (b) Airdata and attitude angles of *ft74srt02run01* before and after correction.

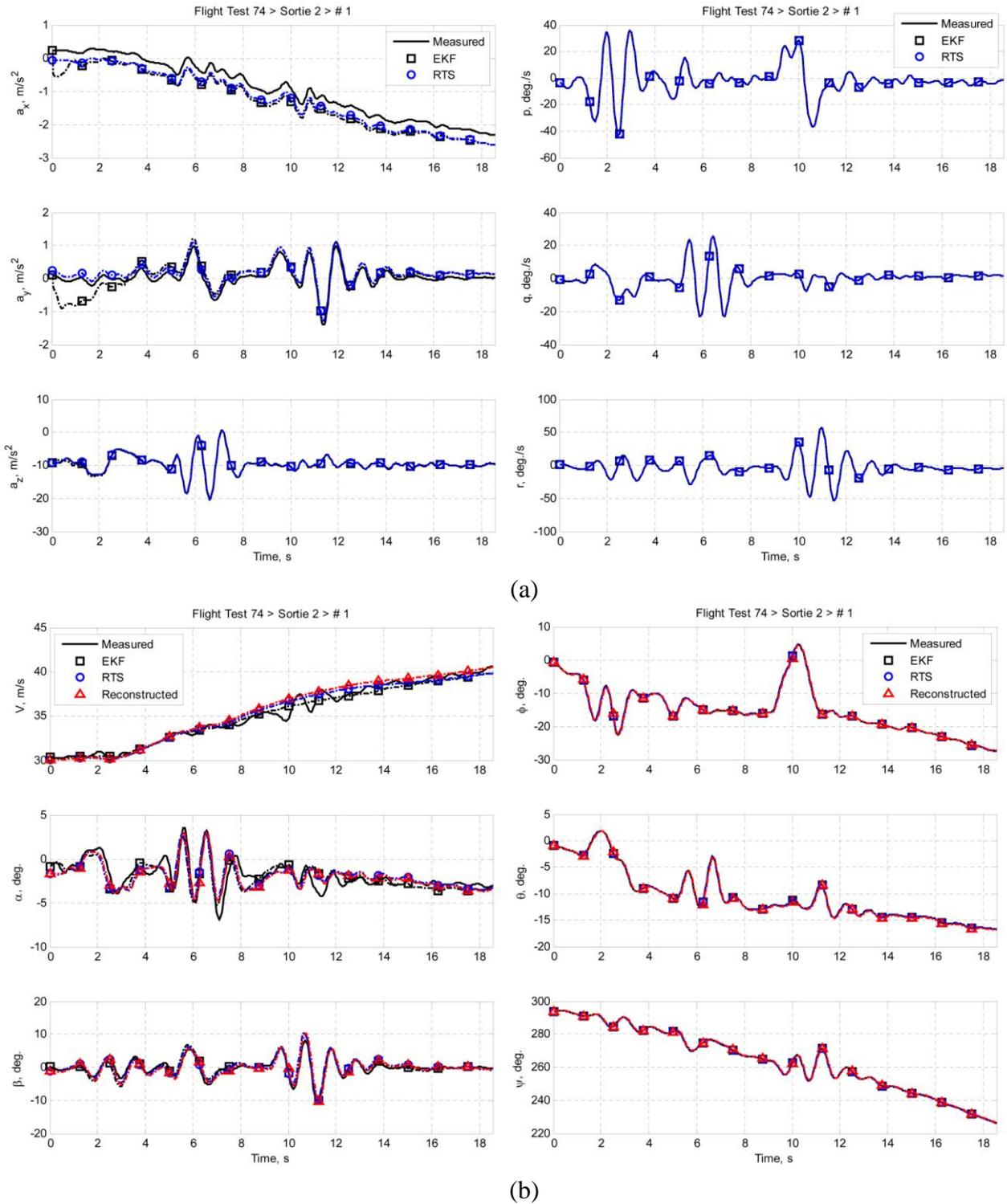
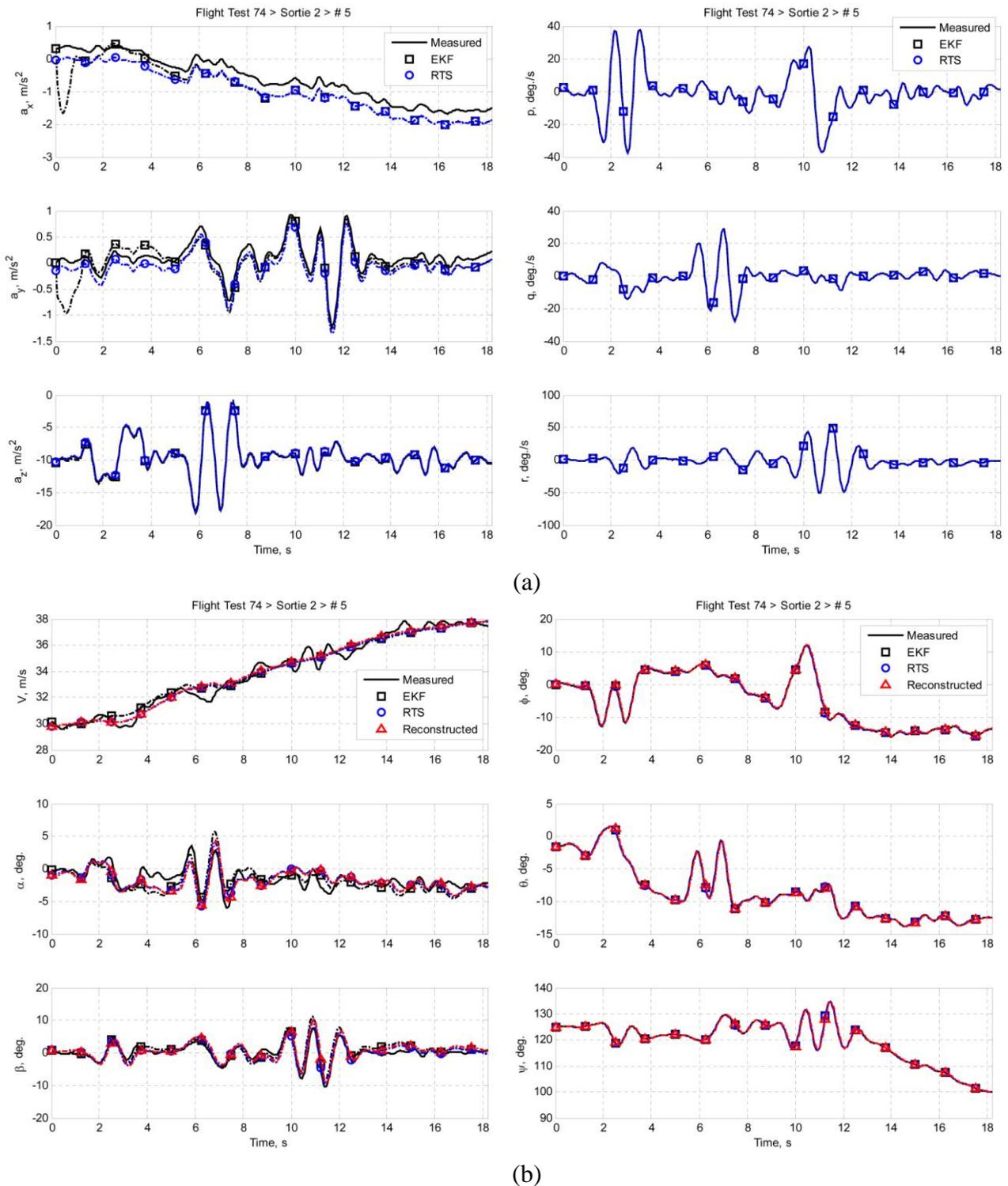


Figure 12. (a) Accelerations and angular rates of *ft74srt02run05* before and after correction. (b) Airdata and attitude angles of *ft74srt02run05* before and after correction.



Tables 1 and 2 demonstrate the quantitative improvement by comparing the RMSDs before and after correction. The compatibilities of all the airdata and attitude angles have improved. The airspeed has the best improvement with RMSD reduction of as high as 97.76%, which is consistent with the significant X-axis acceleration bias estimated in both *ft74srt02run01* and *ft74srt02run05*. Although the

improvements of the attitude angles compatibilities are not apparent in Figures 11 and 12, all of the attitude angles have shown at least 47.43% of RMSD reduction.

Table 1. Comparison of RMSD before and after correction of *ft74srt02run01*.

	Root Mean Square Deviation (RMSD)		RMSD Reduction, %
	Before Correction	After Correction	
Airspeed (V), m/s	2.9855	0.3373	88.70
AoA (α), deg.	1.3855	0.3490	74.81
AoS (β), deg.	2.3143	0.7520	67.51
Roll Angle (ϕ), deg.	1.0314	0.3086	70.08
Pitch Angle (θ), deg.	0.5413	0.2121	60.82
Yaw Angle (ψ), deg.	0.8486	0.4190	50.62

Table 2. Comparison of RMSD before and after correction of *ft74srt02run05*.

	Root Mean Square Deviation (RMSD)		RMSD Reduction, %
	Before Correction	After Correction	
Airspeed (V), m/s	3.3453	0.0751	97.76
AoA (α), deg.	1.7380	0.4319	75.15
AoS (β), deg.	2.6914	0.9741	63.81
Roll Angle (ϕ), deg.	0.7183	0.3625	49.53
Pitch Angle (θ), deg.	0.4090	0.2138	47.73
Yaw Angle (ψ), deg.	1.6547	0.3729	77.46

9. Conclusions

This paper presented an easy and straightforward procedure for implementing sensor compatibility corrections for fixed-wing UAVs. The procedure implemented the Wiener Type Filter, the RTS Smoother, and an airdata noise variances estimation method. The Wiener Type Filter was introduced to filter out the high frequency noises of the flight data, thus simplifying the measurement noise matrix and process noise matrix estimation. The RTS Smoother serves as the main correction algorithm. It is a two pass algorithm that combines the EKF and a backward recursion. The airdata noise variances estimation method accomplishes its purpose using the GPS speed measurements.

The sensor data compatibility correction procedure has been successfully implemented on real flight data recorded on the SP-80 UAV of the RMRL. Initial data reconstruction shows that the compatibility is poor and the airdata is far much noisier than the AHRS data due to the exposure of the airdata probe to the atmospheric turbulence. It was assumed that the AHRS data noises only consist of high frequency noises and the noises can be effectively filtered by the Wiener Type Filter. Also, due to the high compatibility of the attitude angles, it was assumed that the biases of the angular rates are negligible. Since the airdata is corrupted by atmospheric turbulence, the Wiener Type Filter was unable to suppress all the noises. Thus, relatively accurate airdata noise variances are needed. The airdata noise variances estimation method brings about the concept of data regionality. Under this

concept, the data around a particular data point are assumed to have statistical relation with the data point. This has allowed the estimation of the expected value and variance at any particular data point by taking the data around the data point into account. This concept might not apply universally to all problems, but it has adequately fulfilled its purpose for the work presented in this paper.

After the correction, significant sensor compatibility improvement, especially on the airspeed data has been demonstrated. The improvement of the airdata compatibilities is mainly contributed by the estimation of the acceleration biases by the RTS. On the other hand, the Wiener Type Filter contributes to the improvement of attitude angle compatibilities.

The main benefit of the sensor data compatibility correction is to provide better data reliability, which leads to a better understanding of the dynamics of an air vehicle. Such a correction procedure is essential if an accurate dynamic model is to be identified using the flight data. The Wiener Type Filter and RTS are both batch (offline) methods. Future improvement to the procedure could include the implementation of a real-time (online) algorithm. Being able to perform sensor compatibility correction in real-time could help improve the robustness of the autopilot system for both manned and unmanned vehicles.

Acknowledgements

This work is supported by the National Cheng Kung University, Taiwan, R.O.C. under grant number, D98-3340 and D99-3340, under the Ministry of Education's 5Y50B Special Funding, and by National Science Council of Taiwan under the Contract No. NSC 98-2221-E-006-134-MY3.

References

1. Jonker, H.L. *Application of the Kalman Filter to Flight Path Reconstruction from Flight Test Data Including Estimation of Instrumental Bias Error Corrections*; Report VTH-162; Department of Aerospace Engineering, Delft University of Technology: Delft, The Netherlands, 1976.
2. Klien, V.; Schiess, J.R. *Compatibility Check of Measured Air Craft Responses Using Kinematic Equations and Extended Kalman Filter*; NASA Technical Note TN D-8514; NASA: Washington, DC, USA, 1977.
3. Whitmore, S.A.; Larson, T.J.; Ehernberger, L.J. *Air Data Position-Error Calibration Using State Reconstruction Techniques*; NASA Report TM-86029; NASA: Washington, DC, USA 1984.
4. Chu, Q.P.; Mulder, J.A.; van Woerkom, P.Th.L.M. Aircraft flight path reconstruction with nonlinear adaptive filters. In *Proceedings of the American Control Conference*, Seattle, WA, USA, June 1995; pp. 1196-1200.
5. de Mendonça, C.B.; Hemerly, E.M.; Góes, L.C.S. Adaptive stochastic filtering for online aircraft flight path reconstruction. *J. Aircraft* **2007**, *44*, 1546-1558.
6. Lee, C.S.; Chan, W.L.; Jan, S.S.; Hsiao, F.B. A linear-quadratic-Gaussian approach for automatic flight control of fixed-wing unmanned air vehicles. *Aeronaut. J.* **2011**, *115*, 29-41.
7. Lee, C.S.; Chan, W.L.; Hsiao, F.B. Implementation of system identification on unmanned aerial vehicle via subspace and prediction error method. *J. Aeronaut. Astronaut. Aviat. Series A*; **2010**, *42*, 87-98.

8. Lee, C.S.; Hsiao, F.B.; Jan, S.S. Design and implementation of linear-quadratic-Gaussian stability augmentation autopilot for unmanned air vehicle. *Aeronaut. J.* **2009**, *113*, 275-290.
9. Lee, C.S.; Chan, W.L.; Hsiao, F.B. The development of Spoonbill UAV and LPV modeling of longitudinal dynamics. In *Proceedings of 23rd Bristol International UAV Systems Conference*, Bristol, UK, April 2008.
10. *U.S. Standard Atmosphere*; 1976. NOSA, National Aeronautics and Space Administration, United States Air Force: Washington, DC, USA, 1976; pp. 10-12.
11. Anderson J.D., Jr. *Introduction to Flight*, 4th ed.; McGraw-Hill: Singapore, 2000; pp. 147-157.
12. Nelson R.C. *Flight Stability and Automatic Control*, 2nd ed.; McGraw-Hill: Singapore, 1998; pp. 96-105.
13. Morelli, E.A. Estimating noise characteristics from flight test data using optimal Fourier smoothing. *J. Aircraft* **1995**, *32*, 689-695.
14. Rauch, H.E.; Tung, F.; Striebel, C.T. Maximum likelihood estimates of linear dynamic systems. *AIAA J.* **1965**, *3*, 1445-1450.
15. Gelb A.; Kasper, J.F., Jr.; Nash, R.A., Jr.; Price, C.F.; Sutherland, A.A., Jr. *Applied Optimal Estimation*; Gelb, A., Ed.; The MIT Press: Cambridge, MA, USA, 1974; pp. 156-170, 180-189.
16. Jategaonkar, R.V. *Flight Vehicle System Identification: A Time Domain Methodology*; Lu, F.K., Ed.; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2006; pp. 344-350.
17. Klein, V.; Morelli, E.A. *Aircraft System Identification: Theory and Practice*; Schetz, J.A., Ed.; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2006; pp. 289-329.
18. Lai, Y.C.; Hsiao, F.B. Development of a low-cost attitude and heading reference system using a three-axis rotating platform. *Sensors* **2010**, *10*, 2472-2491.
19. *Crossbow 440 Series Inertial Systems*; Crossbow Technology: San Jose, CA, USA, 2007; pp. 17-27.
20. Lai, Y.C.; Hsiao, F.B. Application of fuzzy logic controller and pseudo-attitude to the autonomous flight of an unmanned aerial vehicle. *J. Chin. Inst. Engineer.* **2010**, *33*, 387-396.