AN ALGORITHM FOR SENSOR DATA FUSION FOR UNMANNED MINI AERIAL VEHICLES

Corneliu AXENTE, Adrian COMAN, Mircea BOŞCOIANU, Nicolae JULA

Military Technical Academy, Bucharest Romania, cornell2you@yahoo.com, adrico59@yahoo.com, mircea_boscoianu@yahoo.co.uk, njula@mta.ro

Abstract **– This paper presents a sensor fusion algorithm for estimating the flight parameters of an Unmanned Mini Air Vehicle is presented. The sensor fusion algorithm is illustrated through simulation using a nonlinear six-degree-of-freedom model of the aircraft and simple sensor models.**

Keywords: *mini-UAV, sensor fusion, GPS errors, INS errors*

1. INTRODUCTION

Sensor fusion is a method for conveniently integrating data provided by various sensors, in order to obtain the best estimate for a dynamic system's states. Sensor fusion algorithms are particularly useful in low-cost UAV applications, where acceptable performance and reliability is desired, given a limited set of inexpensive sensors. The sensor fusion system can provide: filtered high-rate navigation and control data for increased performance, estimation of the flight parameters which are not measured directly (i.e. attitude angles, angle-of-attack, sideslip), detection of significant changes in aircraft dynamics (i.e. icing, airframe damage), and the ability to replace failed sensor outputs with estimates (graceful degradation).

The aim of this research work is to study the applicability of various sensor fusion algorithms for different airplane configurations. The performance of the sensor fusion algorithms will be tested in simulation using models of the aircrafts and sensors.

2. MODELS

For simulation purposes, a nonlinear 6-degree of freedom model was developed, for use in the Matlab/Simulink environment – Fig.1. The model includes Simulink blocks for the equations of motion, aerodynamics, propulsion, inertia, standard atmosphere, background wind, turbulence, and a WGS-84 Earth model [1]. The aerodynamics is based on look-up tables of wind tunnel test results. The aerodynamic coefficients of the models were obtained from the Slope Soaring Simulator [1]. The Slope Soaring Simulator is an open-source flight.

The propulsion model is based on propeller wind tunnel tests and engine experimental data.

The simulator model takes the following inputs:

- Actuators: control surfaces - flap, elevator, aileron, rudder deflections, as well as throttle setting;
- Atmospheric conditions: background wind vector, turbulence intensity, sea-level pressure and temperature.

The model outputs the following parameters:

- Aircraft states: body-axes velocities, angular rates, attitude angles, position and engine speed;
- Aircraft sensors: position and groundspeed, accelerations, angular rates, and air data (static pressure, dynamic pressure and outside air temperature);
- Aerodynamic coefficients;
- Propeller coefficients;
- Engine parameters;
- Wind-axes velocity components: airspeed, sideslip angle and angle-of-attack;
- Linear position - East, North, Up components, relative to the starting point.

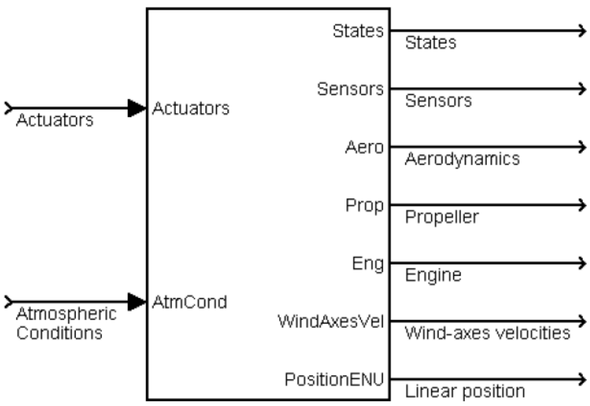

Figure 1: The Simulink Model

The basic sensor set which will be used includes a low cost GPS receiver providing position and groundspeed information at a rate of 1 Hz, 3 accelerometers and 3 rate gyros providing a complete 6-degree of freedom inertial solution, and an air data system which outputs static and dynamic pressure as well as the outside-air temperature.

Simple dynamic models for the sensors were created for simulation purposes. The sensor models were implemented as Simulink blocks and have the following features: white noise, offset drift, scale factor variation, and saturation limits.

The simulation set-up is shown in Fig. 2. The aircraft model can be controlled by fixed actuator commands, manual control using a joystick, or automatic control from an autopilot block. The ideal sensor signals that the aircraft model outputs are corrupted by the sensor model blocks, and then fed to the sensor fusion block, which estimates the aircraft states. The estimated states are then plotted against the actual aircraft states returned by the simulator. This simple comparative plot has the advantage that not only can we see the magnitude of the estimation error relative to the magnitude of the signal, but we can correlate the variation in estimation performance with various aircraft maneuvers during the flight.

Using this simulation set-up we will analyze the performance of each of the sensor fusion algorithm which is presented next.

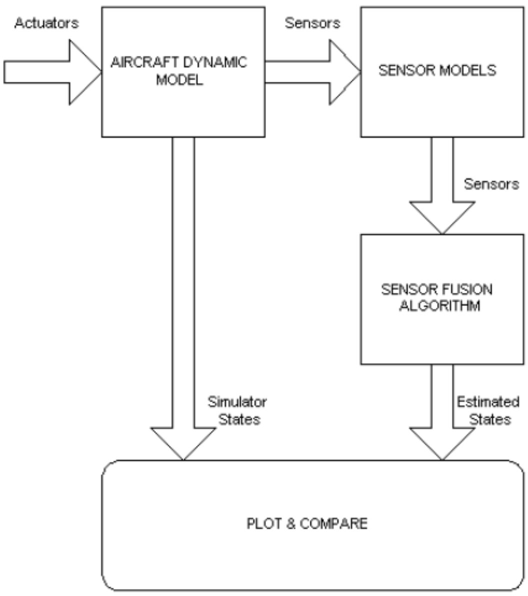

Figure 2: The simulation set-up

3. CONSIDERATIONS ABOUT THE GPS / INS INTEGRATION

The INS algorithm integrates the accelerations and angular rates provided by an Inertial Measurement Unit (IMU) to compute the position, velocity, and attitude (PVA) of the vehicle. The algorithm takes into account the Earth rotation rate and geodic shape, and it also includes a gravity model.

An INS algorithm by itself is seldom useful since the inertial sensor biases and the fixed-step integration errors will cause the PVA solution to diverge quickly. The navigation system must account for these error sources to be able to correct the PVA estimate.

A low-cost GPS (Global Positioning System) receiver can output the aircraft position and groundspeed. The measurement will be corrupted by time-correlated noise and provided at a low rate, typically 1 Hz - not fast enough for some flight control applications. Also, the GPS signal is susceptible to jamming. However, the position and velocity measurements do not drift over long periods of time.

A low-cost IMU (Inertial Measurement Unit) can output the aircraft accelerations and angular rate which can be integrated by an INS to obtain the aircraft position, velocity, and attitude. The IMU measurements are corrupted by noise, scale factor and bias variations with temperature (nonlinear, difficult to characterize). By integrating the IMU measurements with the INS algorithm, the errors will accumulate, leading to significant drift in the position and velocity outputs. One advantage of the IMU is that it can be sampled at high-rates, therefore it is capable to capture the fast dynamics of the aircraft. But the main advantages over GPS is that the INS is autonomous (does not rely on any external aids), it is immune to jamming and inherently stealthy (does not emit nor receives any detectable radiation).

The disadvantages include the following:

1. Mean-squared navigation errors increase with time.
2. Cost, including:
    a) Acquisition cost, which can be an order of magnitude (or more) higher than GPS receivers.
    b) Operations cost, including the crew actions and time required for initializing position and attitude. Time required for initializing INS attitude by gyrocompass alignment is measured in minutes. Time-to-first-fix for GPS receivers is measured in seconds
    c) Maintenance cost.
3. Power requirements, which have been shrinking along with size and weight but are still higher than those GPS receivers.
4. Heat dissipation, which is proportional to and shrinking with power requirements [2].

The advantages and disadvantages of the GPS and INS sensor systems makes them complementary, and the best estimates of the aircraft position, velocity and attitude can be obtained by combining both GPS and INS measurements using the GPS/INS integration method presented below.

4. THE SIMULATION SETUP AND RESULTS

To setup the simulation we had to implement two models:

1. The GPS model
2. The IMU model

The data obtained is processed by a navigation filter.

The outputs of the GPS model are the position and the velocity. The IMU model defines the biases and the noise for the accelerometers and gyros.

The estimated parameters of the UAV are computed by the navigation filter, using the outputs of the GPS and IMU model. Also, the GPS and IMU outputs are ploted to compare the simulated results with the one obtained from the navigation filter.
The model takes as input commands the airspeed command which was setup to the constant value of 26m/s and the bank angle command which is set to the value of 0^0. The wind velocity is set to zero for all the three axes, but it can be modified. The simulation time is set to 50 seconds. The results are presented in Figure 3.

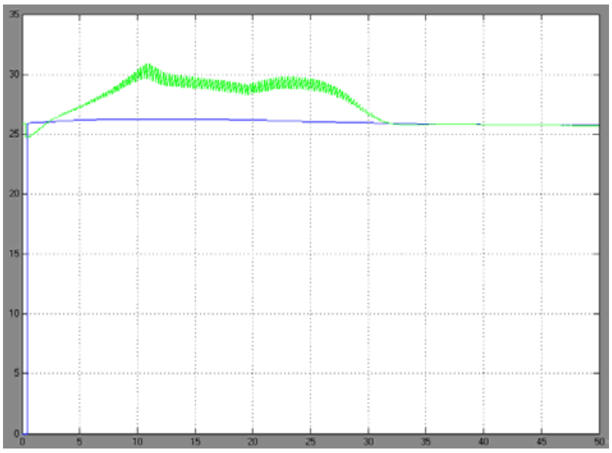

Figure 3: Simulation results for 0^o bank angle

The blue line represents the simulated value for the groundspeed on x-axis, and the red line is the one obtained from the navigation filter.
Similar results are obtained for the y-axis and z-axis.
A second simulation was made, in which the command for the bank angle is set to 15^0. The results are shown in Figure 4.

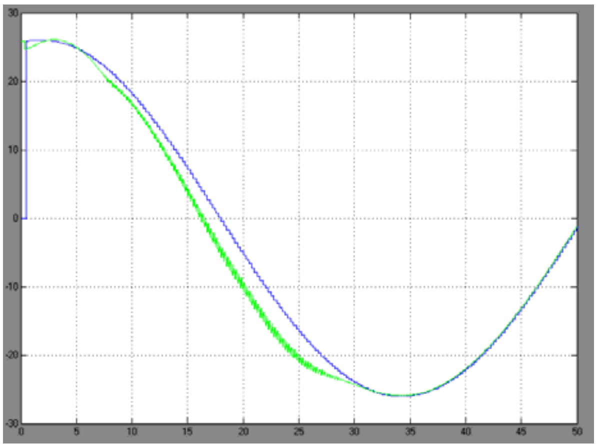

Figure 4: Simulation results for 15^0 bank angle

5. CONCLUSIONS

From Fig. 3 we can see that for a time period of about 30 seconds the speed estimated by the navigation filter is greater with about 4m/s than the speed computed by the GPS model. After this period the filter stabilize itself to a value near to 26m/s. Figure 4 shows that the results obtained with the navigation filter are following closely enough the simulated results. The navigation filter has a satisfactory behavior, but future work must be done for optimizing it.

Acknowledgments

We would like to acknowledge mister engineer Marius NICULESCU from Unmanned Dynamics for his priceless support which was decisive in conceiving this paper.

References

[1] AEROSIM BLOCKSET Version 1.2 User's Guide Unmanned Dynamics, LLC.

[2] Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews, Global Positioning Systems, Inertial Navigation and Integration, John Willy & Sons Inc., 2001.