A Flight Evaluating System using Flight Gear

Zhang Huamin
Department of Aerial Ammunition Engineering
The First Aeronautical College of Air Force
Xinyang, Henan, China
E-mail:zyrll@126.com

Li Li
Department of Aerial Instrument and Electric Engineering
The First Aeronautical College of Air Force
Xinyang, Henan, China
E-mail:zyrll@163.com

***Abstract*—This paper describes a system for evaluating flight of fixed-wing aircraft. During the flight of an aircraft, flight parameters are obtained and transmitted to the ground station by means of wireless communication. These flight data are received by the ground station and are calculated so as to get the input of flight gear, by internet interfaces flight gear can simulate the real-time state of the aircraft on the ground. The system can be used to evaluate the flight performance of a fixed-wing aircraft.**

Keywords- evaluating; flight; UAV; air data system; ground station; flight gear

I. INTRODUCTION

Recent developments in sensor technology, data processing hardware, and software algorithms have made use of the Uninhabited Aerial Vehicle (UAV) a highly feasible approach to achieving a variety of aerial mission objectives at lower risk and cost. UAV technology has the potential for use in many applications such as aerial surveys, meteorological data collection, autonomous target identification, and reconnaissance missions. Additionally, the UAV provides an inexpensive and efficient experimental platform for flight control and planning research.

An ongoing project to develop and demonstrate UAV technology is described in this paper. Equipped with an embedded on-board control system and R/F serial link to a monitored ground station, UAV uses a variety of sensors and software algorithms for real-time trajectory planning, guidance, control, and system identification.

This paper begins with a description of UAV instrumentation and computer systems, followed by an outline of the real-time software architecture and processes employed during flight. We discuss UAV research topics in system identification and dynamic flight planning with a focus on fault detection and recovery, then conclude with a description of our flight test program and the current project status.

II. RELATED COMPONENTS OVERVIEW

UAV carries an extensive instrumentation package, including an Inertial Measurement Unit (IMU), air data system (ADS), differential GPS, hall-effect engine tachometer, strain gauges for measuring engine thrust, and potentiometers to measure control surface deflections. While state estimation and system identification software incorporate data from all sensors.

The on-board inertial measurement package consists of 6 solid state accelerometers, 3 aerospace solid state rate gyroscopes, and a 3 axis solid state fluxgate magnetometer.

The Air Data System (ADS) measures angle of attack, angle of side-slip, and dynamic pressure. Angles are sensed using low-friction potentiometers connected to vanes that align themselves with the local airflow. The dynamic pressure is sensed with a Pitot probe connected to a pressure transducer. The resulting system is capable of providing vehicle airspeed and wind direction during flight.

The ADS is configured on a boom in front of the aircraft to avoid flow interference from the vehicle. The vane shafts are positioned in a plane perpendicular to the axial direction of the boom and the Pitot probe extends out the end of the boom. The ADS is designed so that angles of attack and side-slip can be measured at speeds in excess of 20 miles per hour, which is slightly below the predicted stall speed of UAV.

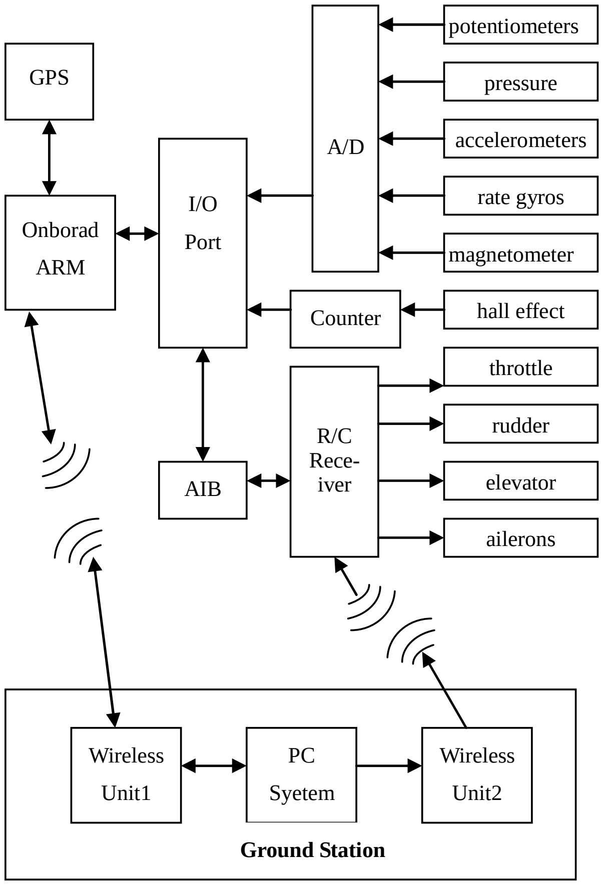
UAV is flown using a combination of on-board and ground-based computers. The aircraft contains ARM single-board computers, while the ground station computer is a Pentium laptop. The on-board ARM serially communicates with the ground station via wireless unit. The ground station performs only non-critical tasks with respect to real-time control deadlines, including the user interface, long-term data storage between flights, and development of the baseline flight plan to accomplish mission goals. The on-board processors read all instrumentation, perform all real-time control tasks, output actuator commands to the aircraft, and control communications with the ground station.

Fig. 1 shows the interface between the flight computers and aircraft instrumentation. The on-board 486 reads all A/D, hall effect (tachometer), and actuator interface board (AIB) data from I/O ports through PC104 bus, and also outputs actuator commands to the AIB. The on-board Pentium handles all serial links, including that from the PC104-based GPS and to/from the ground station. UAV actuators are controlled manually using a standard PPM (Pulse Position Modulation) R/C transmitter/receiver pair.

The AIB is a custom-built board mounted on the PC104 computer stack which has two functions. First, it reads transmitted pilot commands, and second, it outputs the computer-generated actuator commands (for augmented or fully-automated control). Pilot-transmitted commands are

Published by Atlantis Press, Paris, France.

intercepted by the R/C receiver, decoded by a microcontroller on the AIB, then read by the 486 from the PC104 bus. Actuator commands are output by encoding current actuator signals into a single PPM line that is fed into the R/C receiver and subsequently to aircraft actuators. A control switch on the pilot's transmitter allows quick switching between computer and pilot control modes. This feature allows the pilot to override the computer manually if necessary.



Data acquisition and controller interface

III. Software Design

UAV software integrates procedures that read sensors, estimate state, develop flight plans, perform model ID, and compute actuator commands. Except that, also require that on-board parameters be communicated to/from a user via the ground station computer. In this section, outline UAV software procedures and their real-time constraints, as well as describe how achieve hard real-time execution for critical tasks.

Fig. 2 illustrates the software procedures of fully-automate the aircraft. For a typical flight, the ground station begins by building, scheduling, and serially transmitting a flight plan to the flight computers. Once this plan is received, the guidance procedure uses the way-point trajectory plan and aircraft state estimate to issue the initial reference trajectory signal. During flight, identification uses state estimates and actuator commands to update the aircraft model, while the controller uses the current state estimate and reference signal to compute actuator commands.

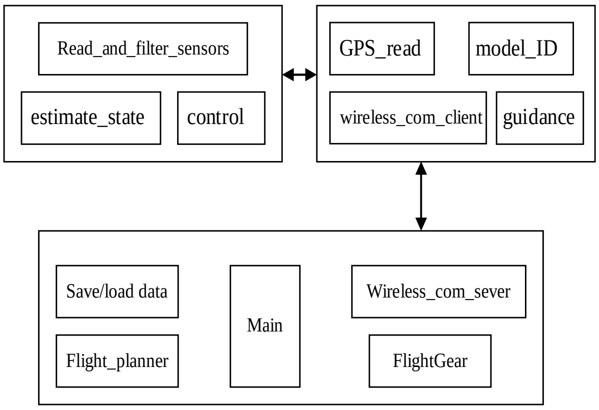


Figure 1. Software procedures of fully-automate the aircraft.

Throughout the flight, the ground station will run a graphical user interface (GUI), gives the user real-time access to a limited set of aircraft data and allows the user to input a limited set of high-level control commands. All flight data will be stored with harddisk during flight, and then downloaded for permanent storage between flights. Except for serial data handling and the GUI, the ground-based processor may remain idle for much of the flight, although dynamic alterations in the flight plan may be required. An important aspect of embedded control system design is guaranteeing real-time execution of critical tasks. The dynamics associated with aircraft flight require meeting hard real-time deadlines both to maintain stability and react quickly and safely to the large variety of normal and anomalous situations that can occur. The computers use the QNX real-time OS which supports strict adherence to hard real-time task schedules. In order to guarantee response times for critical tasks (e.g., reading sensors and executing the control loop), CPU and serial communication resources in the system is carefully allocated. On the UAV, primary time-critical procedures execute on-board the aircraft. Processes executing on the ARM (shown in Fig. 1) execute at fixed period and worst-case execution time. This results in predictable CPU usage, so we use a static schedule to execute these tasks. Reading the GPS will take near-constant execution time. However, the guidance and ID processes may require computations with large execution time variance, so they will be monitored by a dynamic scheduler that can preempt execution if a more critical task (e.g. reading GPS) must be performed.

The wireless connection uses a client/server model with the aircraft CPU acting as the client. This design allows the on-board ARM to optimize CPU utilization by controlling both the quantity and type of messages transmitted between ground station and aircraft. The on-board client uses execution times that have been required by high-variance tasks (e.g., model ID) to compute message transfer parameters. For example, if the last model ID iteration was very slow, there will be little or no communication with the ground station, but if model ID and guidance are both fast, serial communication will be given a large percentage of the on-board CPU time.

The data from aircraft on-board can be used by Flight-Gear online or offline. Flight Gear is an open-source flight Simulator. It provides a native network interface for interacting with the internal states of the flight simulation such as airspeed, altitude, accelerations, and position. To produce Flight Gear data from engineering, bridging modules was developed. Depending upon the data item, raw data can be in engineering units (data from the GPS) or in raw counts (data from the accelerometers, gyros, and pressure transducers). The bridge application reads and converts the flight data which is saved or realtime, feeding back into Flight Gear by network port. Then Flight Gear displays the flight in real time or offline. Fig. 3 shows a screenshot simulating flight.

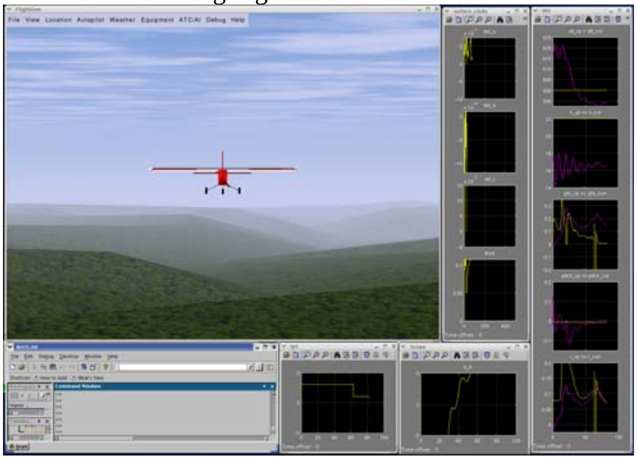

File View Location Autopilot Weather Equipment ATC/AI Debug Help
MATLAB
Command Window

Figure 2. Flight simulation

To driver Flight Gear with extern data, configure file model-set.xml must be modified, remove all FDM data file, and add sentence as follow: <flight-model>network</flight-model>. These means that FDM data is from network, flight data can be input to Flight Gear via network and control simulating model.

IV. CONCLUSIONS

This project has described a system focusing on the instrumentation, computer hardware and software that is required to allow an embedded control system to fly autonomously. UAV research involves system identification, reconfigurable control, and dynamic flight planning, with UAV tests illustrating fault detection and recovery capabilities. With flight data, UAV flight can be simulated and evaluated by Flight Gear. In addition, flight training for pilots can also be evaluated.

REFERENCES

[1] J. H. Chen, S. C. Lee, D. B. Debra. “Gyroscope Free Strapdown Inertial Measurement Unit by Six Linear Accelerometers,” Journal of Guidance, Control, and Dynamics, vol. 17, no. 2, 1994, pp.286-290.

[2] T. H. Van Pelt and D. S. Bernstein. “Least Squares Identication Using μ -Markov Parameterizations,” Proc. IEEE Conf. Decision and Control, December 1998.

[3] R.W. Deters, G. A. Dimock, M. S. Selig, “Icing Encounter Flight Simulator with an Integrated Smart Icing System,” AIAA Modeling and Simulation Technologies Conference, AIAA Paper 2002-4599, Monterey, CA, August 2002.

[4] G. A. Dimock, R.W. Deters, Selig, M.S., “Icing Scenarios with the Icing Encounter Flight Simulator,” AIAA 41th Aerospace Sciences Meeting, AIAA Paper 2003-0023, Reno, NV, January 2003.

[5] Berndt,Jon S,”JSBSim: An Open Source Flight Dynamics Model in C++”, AIAA Modeling and Simulation Technologies Conference and Exhibil,AIAA Paper 2004-4923,Providence,RI,August 2004.