Solus: An Autonomous Aircraft for Flight Control and Trajectory Planning Research

Ella M. Atkins** Robert H. Miller‡1 Tobin Van Pelt‡2 Keith D. Shaw+
William B. Ribbens‡ Peter D. Washabaugh‡
Dennis S. Bernstein‡
** Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109
‡ Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI 48109
+ Department of Biophysics, University of Michigan, Ann Arbor, MI 48109

Abstract

The University of Michigan has developed a fixed-wing model aircraft (Solus) with an embedded control system to develop and demonstrate UAV technology. The analytical objective of this project is the development of intelligent flight control and trajectory planning techniques, focusing on automated fault detection and recovery. Our experimental objective is to implement and evaluate these techniques on Solus for a variety of mission and fault scenarios.

1. Introduction

Recent developments in sensor technology, data processing hardware, and software algorithms have made the 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.

We describe an ongoing project at the University of Michigan to develop and demonstrate UAV technology. Solus, the University of Michigan UAV test bed shown in Figure 1, is a 1/4 scale Citabria built using standard model aircraft technology. Equipped with an embedded on-board control system and R/F serial link to a monitored ground station, Solus uses a variety of sensors and software algorithms for real-time trajectory planning, guidance, control, and system identification.

Figure 1: Solus - University of Michigan UAV.

The University of Michigan UAV can be operated in three modes, including remotely piloted vehicle (RPV) mode, augmented (pilot-assist) mode, and fully-automated mode. Our project objectives are:

- Accurately model UAV flight dynamics
- Test and evaluate intelligent flight control and state estimation software
- Implement online identification software
- Automate mission and trajectory planning
- Develop and test fault detection, isolation, and recovery techniques

In this paper we begin with a description of Solus instrumentation and computer systems, followed by

[1]Supported in part by NASA under grant NGT4-52404
[2]Supported in part by AFOSR under AASERT grant F49620-97-1-0406S

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.

2. Instrumentation

Solus 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, we focus on the IMU and ADS in this section. Both of these systems were designed, constructed, and calibrated at the University of Michigan.

2.1. Inertial Measurement Unit

The on-board inertial measurement package consists of 6 Analog Devices solid state accelerometers, 3 British Aerospace solid state rate gyroscopes, and a Honeywell 3 axis solid state fluxgate magnetometer. All these instruments are mounted on a 5 inch cube in the plane's fuselage. This design was motivated by [1] as well as integration constraints. Calibration of the IMU cube was conducted as a stand alone unit using an Ideal Aerosmith rate table and Singer Scorsby table. The calibration was then verified using the combination of the rate and Scorsby tables.

2.2. Air Data System

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 mph, which is slightly below the predicted stall speed of Solus. The ADS system was calibrated and tested in the 2 ft $\times$ 2 ft wind tunnel at the University of Michigan.

3. Computer and Interface Systems

Solus is flown using a combination of on-board and ground-based computers. The aircraft contains two PC104 single-board computers, a 486 and a Pentium, while the ground station computer is a Pentium laptop. The two flight computer processors communicate via dual-port RAM, while the on-board Pentium serially communicates with the ground station via R/F modem link at 56 Kbps. Because of serial communication bandwidth limitations, 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.

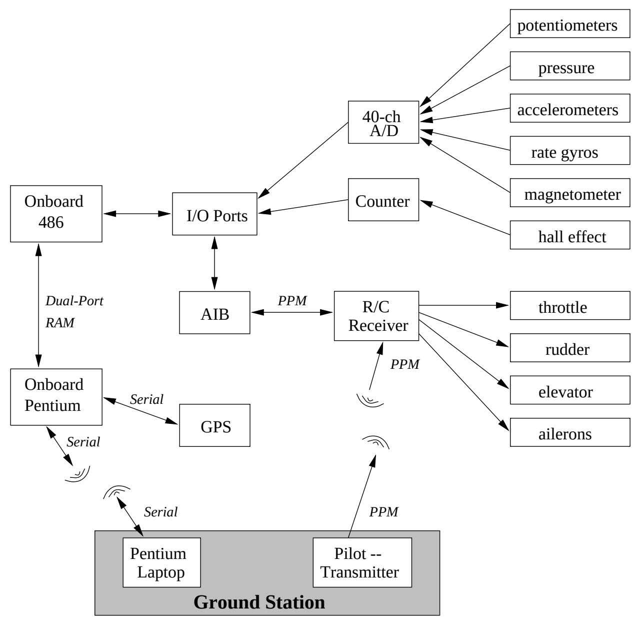

Figure 2: Data Acquisition and Controller Interface.

Figure 2 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 on its 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. Solus 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

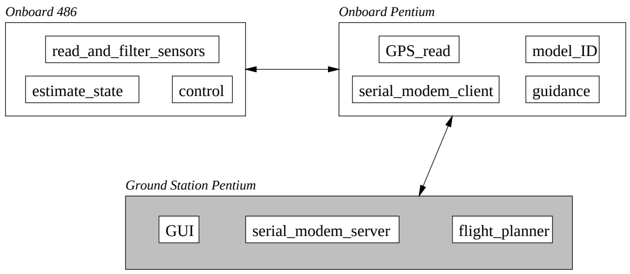

Figure 3: UAV Software Processes.

second, it outputs the computer-generated actuator commands (for augmented or fully-automated control). Pilot-transmitted commands are 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.

4. Software

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

Figure 3 illustrates the software processes required to 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 process 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.

Throughout the flight, the ground station will run a graphical user interface (GUI) that 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. Due to serial communication bandwidth limitations, all flight data will be stored on-board the aircraft during flight, 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. Our computers run the QNX real-time operating system which supports strict adherence to hard real-time task schedules. We carefully allocate CPU and serial communication resources in our system to allow guaranteed response times for critical tasks (e.g., reading sensors and executing the control loop).

On our UAV, primary time-critical processes execute on-board the aircraft. Processes executing on the 486 (shown in Figure 3) 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. On-board Pentium tasks are also critical, but have lower required execution frequencies. 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 pre-empt execution if a more critical task (e.g., reading GPS) must be performed.

The R/F modem serial connection uses a client/server model with the aircraft Pentium acting as the client. This design allows the on-board Pentium to optimize CPU utilization by controlling both the quantity and type of messages transmitted between ground station and aircraft. The on-board serial 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 Pentium's CPU time.

5. Research Objectives

We are focusing UAV research toward fault detection, isolation, and recovery. In this section, we discuss on-line model identification for reconfiguring flight control as well as forecasting aircraft performance capabilities. We describe methods to incorporate model identification techniques for the fault detection and isolation of system anomalies such as airframe icing.

Once each fault class is described from the identified model, we will investigate methods for fault recovery, focusing on trajectory re-planning to accommodate controllability changes. This will be done while continuing to satisfy the largest set of mission goals that are still achievable after the fault has occurred.

5.1. On Line System Identification and Reconfigurable Flight Control

In recent years, off line identification techniques have been successful at building aircraft models from flight data. The identification is performed off line to obtain a dynamic model that is used to design a gain scheduled controller [2]. While this is a viable method of building flight control architectures, the UAV project aims to maintain system operation in the presence of unforeseen changes in aircraft dynamics. Since trajectory planning and reconfiguration of flight controls is to be performed online, it requires fast and efficient implementation of identification algorithms. The UAV will use indirect adaptive control methods, such that an explicit model of the aircraft dynamics is generated on-line and then used for controller design. In addition, this dynamic model will be used to forecast vehicle performance capabilities for use in online trajectory planning.

Quite often measurement and process noise as well as inaccurate approximation of system order make identification difficult. For these reasons, the online identification will be performed using μ-Markov parameterized models [3]. μ-Markov parameterizations are non-minimal transfer functions that have sparse denominator polynomial structure and Markov parameters as coefficients of the numerator polynomial. These Markov parameters can then be used directly for controller design or indirectly by building state space models [4]. When performing least squares identification using μ-Markov parameterizations [3], the estimate of the system's Markov parameters are consistent, or unbiased, in the presence of correlated measurement and process noise. Moreover, the estimates are unbiased even with inaccurate knowledge of system order. This makes it possible to accurately identify aircraft dynamics even when anomalies during flight result in system order changes.

5.2. Fault Detection and Isolation

Fault detection and isolation will be a key technology for autonomous vehicles. Research in this area will also benefit today's piloted vehicles by supplying pilots with additional information about the aircraft they are flying. An example of this is the use of detection filters to indicate the presence of icing by the change in aircraft dynamics. A study by the United Kingdom's Civil Aviation Authority indicated that all the manufacturers of aircraft, and a sizeable portion of commercial pilots interviewed for the study "regarded ice detectors as unreliable" [5]. This is an area that would benefit greatly from an effective online ice detection and identification algorithm. Presently ice detectors are mechanical devices that can only detect icing at specific points on the aircraft.

One part of our research will focus on developing a systems level approach to ice detection that will give a more global picture of the presence and amount of icing. This research can be generalized to identify non-additive changes in dynamics. Improving current methods by making them more robust to the uncertainties inherent in aircraft dynamical models will be a major focus.

5.3. Flight Planning and Fault Recovery

Flight planning in commercial aircraft typically begins by retrieving a pre-computed way-point trajectory based on origin and destination airports. Current weather/wind conditions and air traffic restrictions are used to complete the flight plan with values such as fuel consumption and time en route. Our UAV research in flight planning begins with the incorporation of knowledge-based planning and real-time plan scheduling techniques such as those in [6] to build way-point trajectories that accomplish mission objectives (e.g., surveillance over a specified area for our UAV) given an initial dynamic model and wind estimates. In near-term research, we will concentrate on implementing a planner that incorporates a dynamic model of sufficient accuracy to build feasible flight plans for our UAV.

A major benefit of a knowledge-based planning system is flexibility, gained from a user-friendly world model and the ability to incorporate state feedback into planning, even if the current state was previously "unplanned-for" [7]. As discussed above, model ID research will focus on fault detection and isolation. By performing dynamic trajectory replanning when necessary, we will address the problem of fault recovery. Examples of fault recovery are plans which follow a best-glide trajectory to a desirable landing site upon [simulated] engine failure, or plans which attempt standard techniques such as altitude adjustment or heading reversal to depart icing conditions. The ID algorithm will describe the fault (e.g. icing) in terms of a new dynamic model and environmental parameter changes, which are then fed back to the planner. If the current way-point trajectory can still be achieved, the plan remains unchanged. However,

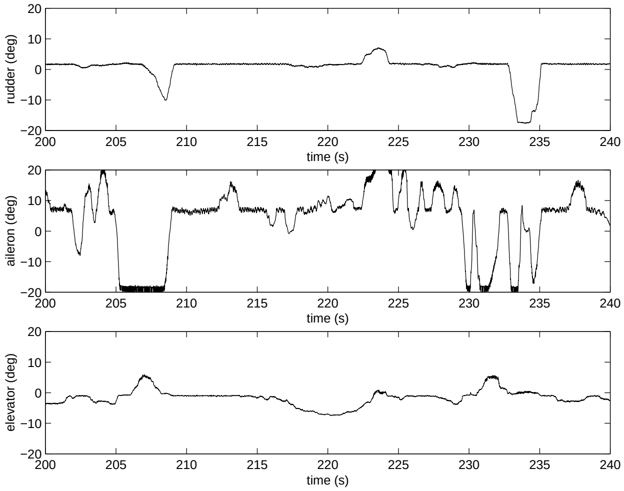

Figure 4: Control Surface Deflections for Continuous (t=205) and 4-point (t=229) Rolls.

if it cannot (i.e. the old plan violates controllability constraints required within the new model), the planner must efficiently build a new plan that achieves as many mission objectives as possible and is dynamically feasible with the new model.

6. UAV Flight Test Program

The University of Michigan UAV can be operated in three modes: as a remotely piloted vehicle (RPV), in augmented (pilot-assist) mode, and fully-automated mode. In RPV mode, a pilot will manually control the aircraft using the R/C transmitter. In addition to performing initial airframe and instrumentation tests, RPV mode will always be used as a backup safety mode and will be useful for conducting identification and estimation research without automatic control. Pilot-assist mode will augment pilot commands with computer-controlled outputs to allow effector mixing, control smoothing and limiting, and control allocation. This mode of operation can address many research issues involving pilot flying qualities, operations safety, and mission optimization while reconfiguring flight controls. During autonomous flight mode, the computer will completely control the aircraft, although the standby pilot will always have the ability to override computer control and enter RPV mode.

Our first test phase, conducted in RPV mode, performs data acquisition to debug hardware, as well as provide experimental data for off-line dynamic model estimates. An example of such data is shown in Figure 4, which contains flight test data illustrating control surface deflections during a roll maneuver sequence. In the next phase, the aircraft remains in RPV mode and uses the previously estimated dynamic model to perform state estimation, as well as test online identification software. We continue with this bottom-up testing strategy, building an initial controller to autonomously fly simple predefined high-altitude cruise trajectories, followed by tests with more complex guidance and flight planning tasks. Once we have a set of working processes for "normal" cruise flight, we will begin to introduce faults that make flight control even more challenging by forcing dynamic adaptation in one or more of the control, model ID, and planning processes.

7. Summary

We have described the University of Michigan's UAV project, focusing on the instrumentation, computer hardware, and software that is required to allow an embedded control system to fly autonomously. Our UAV research involves system identification, reconfigurable control, and dynamic flight planning, with UAV tests illustrating fault detection and recovery capabilities. To-date, we have calibrated sensors and operated the UAV in RPV mode while collecting flight data. We plan to begin testing the UAV in the augmented and fully-autonomous modes for high-altitude flight maneuvers later this year.

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*, 17(2):286–290, March-April 1994.

[2] J. G. Batterson. STEP and STEPSPL - Computer Programs for Aerodynamic Model Structure Determination and Parameter Estimation. Technical report, NASA TM-86410, 1986.

[3] T. H. Van Pelt and D. S. Bernstein. Least Squares Identification Using μ-Markov Parameterizations. To appear in *IEEE Conf. Decision and Control*, December 1998.

[4] J. Juang. *Applied System Identification.* Prentice-Hall, Englewood Cliffs, NJ, 1994.

[5] P.M. Render, L.R. Jenkinson. Investigation into Ice Detection Parameters for Turboprop Aircraft. *Journal of Aircraft*, 33(1):125–129, January-February 1996.

[6] D. J. Musliner, E. H. Durfee, K. G. Shin. World Modeling for the Dynamic Construction of Real-Time Control Plans. *Artificial Intelligence*, 74:83–127, 1995.

[7] E. M. Atkins, E. H. Durfee, K. G. Shin. Detecting and Reacting to Unplanned-for States. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 571–576, July 1997.