Software Enabled Control
for
Intelligent Uninhabited Aerial Vehicles (UAVs)

Sponsored by DARPA
Software Enabled Control Program
Contract #F33615-98-C-1341

Status Report
November 1, 1998 - January 31, 1999

Principal Investigators:

Georgia Tech Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150
(404) 894-6257
daniel.schrage@ae.gatech.edu

George Vachtsevanos
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250
(404) 894-6252
george.vachtsevanos@ece.gatech.edu

Key Personnel:

Georgia Tech Bonnie Heck, J.V.R. Prasad, Linda Wills
Boeing Bryan Doerr, Kevin Wise

Project Objectives:

The objective of this project is to develop software-enabled control methods for complex dynamic systems with application focus on intelligent UAVs. The approach is to use a hierarchical control structure where mission planning and situation awareness are at the highest level, the flight control (or stability augmentation control) is at the lowest level, and a mid-level controller coordinates the transitions between modes. The focus in this project is to develop mid-level control algorithms for smooth mode-switching. The software integration methodology is based on developing and implementing a plug-and-play, real-time software architecture. The technology will be demonstrated on a VTOL UAV hardware-in-the-loop simulation.

Status of Project:

There are three main tasks in this project:

- Task I: Mid-Level Coordination for Mode Switching and Reconfigurable Control
- Task II: Control Integration and Simulation Demonstration
- Task III: VTOL UAV Demonstration

The recent accomplishments for each task are given below. The project is on schedule for each of the subtasks.

Task I: Mid-Level Coordination for Mode Switching and Reconfigurable Control

<u>Mode Switching Control Subtask:</u>

We are developing a mode-switching controller methodology for helicopters. The features of the control are that it uses an optimization scheme to devise a trajectory for the state variables to follow when changing modes, and it blends local mode controllers in such a way as to achieve the desired trajectory. We are currently using the Apache helicopter for the model, but are automating the design process as much as possible so that the design for a different type of helicopter (or even an entirely different type of plant) would not take significant time. Currently, we are focussing on designing a controller to go from a hover mode of operation to a forward flight (50 knots) in minimum time for Apache Helicopter model. The 'hover to forward flight' controller will be analyzed using robust stability and sensitivity analysis tools. In addition, we have submitted two journal papers on this topic:

1. "Design of Mode to Mode Fuzzy Controllers," *International Journal of Intelligent Systems*.
2. "Robust Stability of Mode to Mode Fuzzy Controllers," *Journal of Guidance, Controls and Dynamics*.

<u>Fault Tolerant Control Subtask:</u>

We have been proceeding along two paths in the area of fault tolerance. On the design side, the applicability of the mode-switching controller as a tool for fault tolerance is being investigated. In this scheme, the system configurations before and after the fault occurrence are considered 'modes'. The mode-switching controller then attempts to transfer the system stabley from the normal mode to the fault mode. On the analysis side, the scenario assessment techniques are being used to give more precise meanings to ideas such as *isolatability*, *reconfigurability*, and *fault tolerant*. These metrics will assist the initial design of system architectures to insure some level of fault tolerance is inherent in the system. Also, we have been designing a generic simulation, consisting of both mechanical and electrical components, to assist both the analysis and design approaches to fault tolerance.

Task II: Control Integration and Simulation Demonstration

Open Control Platform Subtask:

In collaboration with Boeing, we are developing an open platform that will facilitate integration and reuse of new software enabled control algorithms and other hardware/software components. During this reporting period, we have designed a first generation demonstration system which we plan to implement and evaluate in the coming quarter. This first generation system has as its core Boeing's BoldStroke system which allows distributed heterogeneous application components to communicate asynchronously *in real-time* using the Common Object Request Broker Architecture (CORBA) standard without CORBA's high overhead services. The specific real-time issues addressed are priority-based event dispatching, scheduling, and preemption, and efficient event filtering and correlation.

We have installed BoldStroke on Windows-NT machines at Georgia Tech and we are defining a reference architecture on which to incrementally build increasing functionality. Boeing Phantom Works spent two days at Georgia Tech transferring their BoldStroke Open Systems Architecture. The complete source code and documentation was delivered. Since BoldStroke was mainly designed for Operational Flight Program implementation, discussions on how BoldStroke may be adapted to Control applications were held. Particular strengths of BoldStroke, such as event priority management, were identified and, more importantly, its limitations were discussed. We also recognized that exposing the entire BoldStroke API would overburden the controls developers.

To reduce the complexity of interacting with the raw BoldStroke API, we decided to wrap BoldStroke in a small number of C++ classes that provide the minimum functionality required to implement control algorithms. Additionally, this new simplified API resembles the controls domain representation of objects as systems that respond to stimuli. We are currently constructing simple systems using this new simplified API. Additional facilities will be added incrementally as required by the control algorithms. The simplified API will eventually evolve into the mid-level control OCP API. This method will decouple development of control algorithms from the intricacies of BoldStroke, as well as decouple the BoldStroke API from the mid-level OCP APIs.

Simulation And Modeling Subtask

We have received the simulation package FLIGHTLAB and have begun using it for simulating the vehicle including the stabilizing controllers. Because of the modular features of this package, we can use it for simulating faults and analyzing controller reconfiguration schemes. In particular, FLIGHTLAB was set –up for the verification of the proposed main rotor RPM control to be used in failure modes of Helicopter-UAV's. It was possible to simulate changes in main rotor RPM of the model by changing the reference speed of the main rotor governor. Our simulation results to date indicate that the main rotor RPM Control can be used as a substitute for the collective pitch control. Also, the RPM control can partially substitute for main rotor cyclic control and for tail rotor control for controlling of the vehicle attitudes. Both FLIGHTLAB and TMAN simulation programs have been setup for integration of the adaptive neural net controller (which serves as the stabilizing controller for the vehicle). Scenarios including main rotor collective pitch actuator failure, stuck tail rotor collective pitch actuator, stuck cyclic pitch control actuator and total tail rotor failure are currently being investigated to demonstrate the effectiveness of the proposed reconfiguration and replanning methods.

Task III: VTOL UAV Demonstration

We have outlined a mission scenario that will help drive the development of our first generation demonstration system. This scenario involves the integration of prototype mode transition management and fault tolerant control components which handle an initially small set of flight mode transitions and failure conditions. The demonstration system focuses on the use of BoldStroke for distributed client/server communication, priority management, and asynchronous event processing. In the coming quarter, we will define use cases and detailed scenarios and based on these, derive component interfaces, timing constraints, task prioritization and event dependencies. In transitioning our control algorithms to BoldStroke, we are investigating ways to make the overall control system more broadly applicable, scalable, and evolvable. We expect to have a suitable demonstration of part of the overall mission scenario for the DARPA meeting in May. This demonstration will highlight some of the key features of the open control platform as well as the mode-switching and fault tolerant algorithms.