IAV2004 - PREPRINTS
5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles
Instituto Superior Técnico, Lisboa, Portugal
July 5-7, 2004

INTELLIGENT CONTROL OF UNMANNED AERIAL VEHICLES FOR IMPROVED AUTONOMY AND RELIABILITY

George Vachtsevanos, Liang Tang, Graham Drozeski, and Luis Gutierrez

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia, 30332-0250, USA

Abstract: This paper reviews aspects of Unmanned Aerial Vehicle (UAV) autonomy as suggested by the Autonomous Control Logic chart of the U.S. DoD UAV Autonomy Roadmap; levels of vehicle autonomy addressed through intelligent control practices and a hierarchical/intelligent control architecture is presented for UAVs. Basic modules of the control hierarchy and their enabling technologies are reviewed; of special interest, from an intelligent control perspective, are the middle and high echelons of the hierarchy. Here, mission planning, trajectory generation and vehicle navigation routines are proposed for the highest level. At the middle level, the control role is portrayed by mode transitioning, envelope protection, real-time adaptation and fault detection/control reconfiguration algorithms which are intended to safeguard the UAV's integrity in the event of component failures, extreme operating conditions or external disturbances. The UAV thus exhibits attributes of robustness and operational reliability assuring a satisfactory degree of autonomy. The control technologies are demonstrated through flight testing results and the paper concludes with brief remarks on recent research directions regarding coordinated/cooperative control of multiple UAVs.

Keywords: Intelligent control, autonomous vehicle, flight control, adaptive control, fault detection, fault tolerant control, cooperative control

1 INTRODUCTION

Recent world events have highlighted the utility of unmanned aerial vehicles (UAVs) for both military and potential civilian applications. However, the reliability of these systems has been disappointing in practice. According to a recent report, nearly half of the current-generation unmanned surveillance aircraft built have been lost. This loss-rate is about ten times worse than manned combat aircraft. Clearly these numbers are driven in part by the dangerous missions these aircraft are tasked with, but there are other factors at work here. In manned aircraft, the pilot functions as the central integrator of the onboard subsystems and works to mitigate problems when they occur. Although "human error" is attributed as the most common cause of aviation accidents, human pilots are also simultaneously the most important safety-enhancing component on a manned aircraft.

To address this and other related UAV control issues, the Defense Advanced Research Projects Agency (DARPA) and the U.S. Air Force Research Laboratory (AFRL) have launched a major initiative to develop revolutionary new software-enabled control (SEC) systems with applications to intelligent UAVs.

Beyond the reliability of responding to unexpected system faults, the SEC program is also charged with making these machines more agile, thus helping them avoid hostile actions without exceeding critical flight parameters. This has the potential to improve the loss-rate for even the most dangerous missions.

The SEC program includes 16 organizations divided into SEC technology developers of control-related algorithms and SEC developers of the software infrastructure platform that enables the design and implementation of advanced control methods. The organizations include: Boeing Phantom Works, University of California at Berkeley, California Institute of Technology, the Charles Stark Draper Laboratory, Cornell University, Georgia Institute of Technology, Honeywell Laboratories, Massachusetts Institute of Technology, Northrop-Grumman Corporation, University of Minnesota, Oregon Research Institute, Rockwell Science Center, Stanford University, Stanford Research Institute, Scientific Systems Company, Inc., and Vanderbilt University.

Improved performance of UAVs is expected to be achieved when such vehicles are endowed with levels of autonomy that will allow them to operate safely and robustly under external and internal disturbances, to be able to accommodate fault conditions without significant degradation of their performance, to adapt to unexpected events and to coordinate/cooperate among them to accomplish mission objectives. Figure 1 depicts the expected UAV autonomy capabilities according to the U.S. DoD's UAV autonomy roadmap (DoD, 2002).

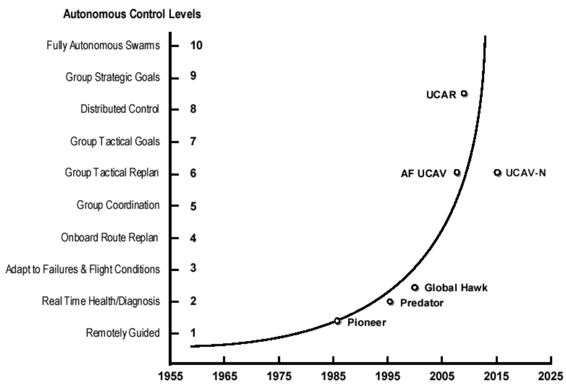


Figure 1. Autonomous control level trend

We suggest below hardware and software technologies aimed to achieve such autonomy objectives.

2 MISSION INTELLIGENT FLOW

A hierarchical control structure for mission intelligence flow is illustrated in Figure 2. Situation Awareness is used for Mission Planning and Flight Mode Selection which constitutes the high level control elements. For inhabited aircraft the pilot and other crewmembers provide the intelligence for interpreting the data from a variety of sources to execute these functions. Much of this data is used in pre-flight or pre-mission planning and is updated on-board as the mission proceeds. As the mission segments are executed and abnormal events are encountered, Flight Mode Switching takes place which constitutes the mid level control element. On an inhabited aircraft the pilot flies the aircraft and makes necessary mode switching and control reconfiguration decisions for implementation through the use of the Flight Control System. This constitutes the low level control element and is used to execute the smooth transition between modes of flight, i.e. transition from hover or takeoff to level flight, etc., and stay within the flight envelope of the UAVs. External Abnormal Conditions cause the pilot to take corrective action, such as avoiding an obstacle or evading a target or threat. Internal Abnormal Conditions can also occur, such as a failure or malfunction of a component on board the aircraft. Once again the pilot provides the intelligence to take the corrective action by reconfiguring his/her set of controls to safely continue to fly or land the aircraft.

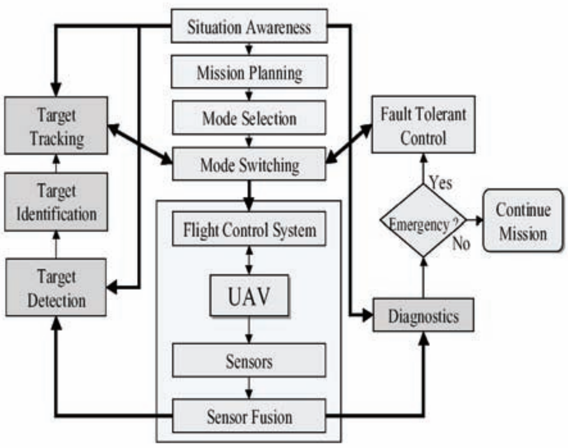


Figure 2. Mission intelligent flow

Without a pilot onboard the aircraft a UAV must either be controlled from the ground by a radio control ground pilot or the UAV must have its own intelligence to fly autonomously. Executing a VTOL UAV mission autonomously has been demonstrated by both the Georgia Tech and Sikorsky Aircraft UAVs in the Army's Advanced Scout Rotorcraft Testbed (ASRT) Project (Schrage, et al. 1997). However, both of the aircraft weren't able to use the entire flight envelope capability of the UAVs, largely limited by the control algorithms implemented. In addition, the control algorithms were very much customized for the particular vehicle's characteristics and were developed in very much of a trial and error approaches. Also, the computing architecture onboard the aircraft did not provide the environment for reusability and reconfigurability, let alone for plug and play of different SEC algorithms (Schrage and Vachtsevanos, 1999).

3 MISSION PLANNING

The high level supervisory controller provides the start and the destination points to the route planner. Upon given a start and a destination points, the route planner generates the "best" route in the form of waypoints for the UAV to follow. A database of the terrain in the form of a digitized map is available to the route planner.

The waypoints are downloaded next to the fuzzy navigator which ensures that the vehicle follows indeed the planned route. The route planner module is designed using a modified A* search algorithm which attempts to minimize a suitable cost function consisting of the weighted sum of distance, hazard and maneuverability measures. The cost elements are expressed as fuzzy membership functions reflecting the inherent uncertainty associated with the planned trajectory, the obstacles along the path and the maneuvers the vehicle is required to perform as it navigates through the terrain. A* uses heuristic knowledge about the closeness of the goal state from the current state to guide the search. Given a search state space, an initial state (start node) and final state (goal node), A* will find the optimal (least cost) path from the start node to the goal node, if such a path exists (Vachtsevanos, et al., 1997a).

The A* routine receives as an input a 2-D digitized map, a start point and a destination point and outputs a listing of 2-D waypoints which, if connected with straight line segments, form a feasible route from the start point to the destination point. Figure 3 illustrates in a flow chart the route planning implementation steps. A typical route for the ASRT with actual mapping data is shown in Figure 4.

The efficiency of the A* algorithm may be improved if heuristic knowledge of the costs associated with the graph nodes is available. A fuzzy inference engine has been designed to accelerate the convergence of the algorithm to the goal state. Simulation results indicate that substantial improvements can be achieved when the degree of confidence in the heuristic knowledge is exploited (Vachtsevanos, et al., 1997b).

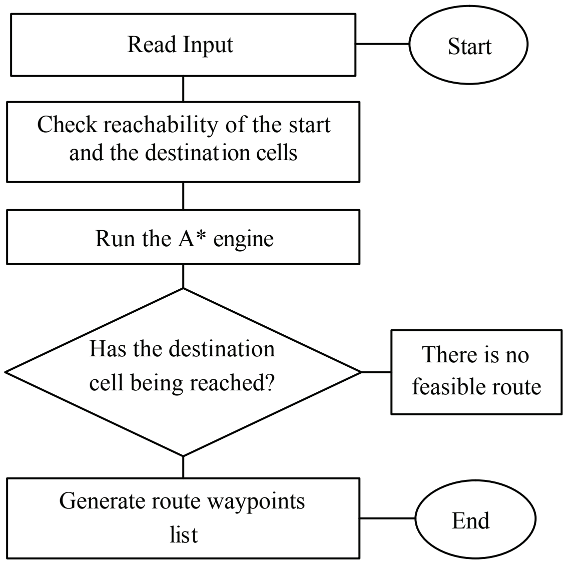


Figure 3. Flow chart of the route planner

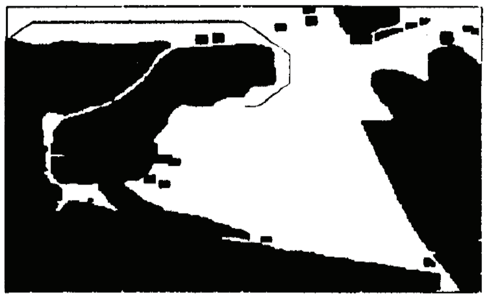


Figure 4. A typical ASRT planned route

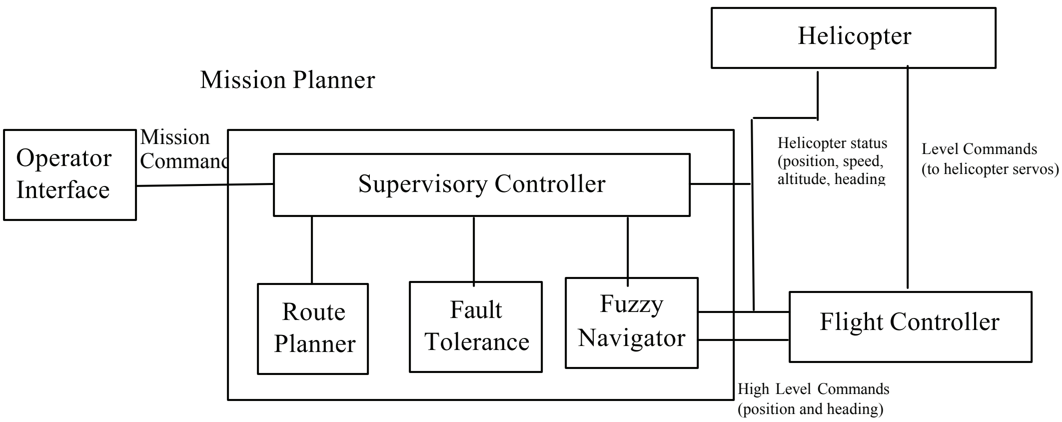


Figure 5 The mission planning configuration

4 FLIGHT TESTING: GTMAX RESEARCH UAV

The GTMax research UAV system, Figure 6, developed at the Georgia Institute of Technology to support SEC and other ongoing programs, utilizes a Yamaha R-Max helicopter, a modular/open avionics system, Open Control Platform, a set of baseline onboard software, and a series of simulation tools. The baseline systems enable autonomous flight of the normally remotely piloted aircraft. The R-Max configured with these systems is known as the GTMax, a highly effective for UAV research vehicle that has a design based on lessons-learned from UAV research at academic institutions such as Georgia Tech, University of California at Berkeley, Massachusetts Institute of Technology, and Carnegie Mellon University for more than ten years.

Figure 6. GTMax helicopter in flight, Georgia Tech's test bed research UAV.

Developed in Japan, the basic Yamaha R-Max helicopter has a rotor diameter of 10.2 feet, a 21 hp two-cylinder engine, and weighs 125 pounds. The weight increases to 160 pounds when configured with typical GTMax avionics. It is capable of carrying approximately 50 additional pounds of research equipment. It also has a generator, starter, and can be flown manually by a remote pilot in sight of the helicopter or by an onboard autopilot.

• Basic Yamaha R-Max Dimensions:
– Max. Length: 3630 mm. (Rotor blade included)
– Fuselage Length: 2750 mm.
– Width: 720 mm.
– Height: 1080 mm
– Fuel Tank: 6 Liter
– Main Rotor Diameter: 3115mm.
– Tail Rotor Diameter: 545mm.
– Max Gross Weight: 93*g N.
– Max. Payload: 30*g N.
• Powerplant:
– Type: Gasoline 2 cycles
– Cylinder configuration: Horizontal opposition 2 cylinder
– Displacement: 246cc.
– Engine RPM. 6350 RPM (Hovering)
– Max Power Ouput: 15.4 KW (21PS)
– Max Torque: 25.5Nm
– Cooling Type: Liquid Cooling
– Fuel: Auto Gas

The GTMax avionics system hardware consists of a set of modules that can be added/removed as required for a flight test. All modules include electro-magnetic interference protection and their own power regulation. The modules are mounted in a vibration-isolated rack within an enclosure under the fuselage. The basic system includes a general-purpose computer, Differential Global Positioning System (D-GPS), an inertial measurement unit, an ultra-sonic altimeter, a 3-axis magnetometer, and two wireless data links. Other flight configurations used to date have also utilized a second general purpose computer, cameras, a radar altimeter, and video capture/compression hardware. The basic ground equipment includes the data links, a GPS reference station, and one or more laptop computers.

The baseline onboard software includes interfaces to the sensors, an integrated navigation filter, and a nominal trajectory-following autopilot. This allows the baseline system to fly a prescribed mission on its own, including the takeoff and landing. The role of the human operator can be to set the desired flight path, start the engine, monitor the flight, and to shut down the engine after landing. This enables a large number of relevant flight control scenarios to be tested, from purely manual control to autonomous operations.

5 UAV ADAPTIVE MODE TRANSITION CONTROL

Control of Autonomous Aerial Vehicles presents unique challenges not only in the design of control algorithms, but also in the strategies and methodologies used to integrate and implement those algorithms on the actual vehicles. We propose an approach to the adaptive mode transition control of UAVs. The main objective of this architecture is to improve the degree of autonomy/intelligence of the UAV and its performance under uncertain conditions, for instance when external perturbations are present. The architecture is based on concepts developed in (Rufus, et al., 2002, 2000a, 200b, Rufus, 2001) where the adaptive mode transition control scheme was first introduced. Here, we suggest a new approach to the adaptive mode transition control problem and we are introducing a hierarchical architecture to implement it. The algorithms have been implemented and tested using the Open Control Platform (Gutierrez, et al., 2003a, 2003b)

The proposed architecture for the control of UAV's consists of a hierarchy of three levels (Figure 7). At the highest level, a mission planning component stores information about the overall mission, generates a low level representation of that mission, and coordinates its execution with the middle level. The middle level includes a trajectory planning component, which receives information from the high level in terms of the next task to be executed to fulfill the mission, and generate the trajectory (set points) for the low level controller. At the lowest level, an adaptive mode transition controller coordinates the execution of the local controllers or the active control models, which stabilize the vehicle and minimize the errors between the set points generated by the middle level and the actual state of the vehicle. The adaptive mode transition control consists of the mode transition control component and the adaptation mechanism component.

The mode transition control component consists of several subcomponents: the local controllers (one for each local mode), the active control models (one for each transition), and the mode transition manager. The mode transition manager decides which controller to use at a given time (a local controller or an active control model) based on the actual state of the UAV.

The local controllers are of the discrete time tracking variety running at a fixed sample rate. The control law for these controllers is given by:

$$u(k) = K_i e(k) + u_{trim,i} \tag{1}$$

where k represents the discrete time, $u(k)$ is the actuator command vector, $e(k)$ is the error between the desired state (set point) generated by the trajectory planning component $(x_d(k))$ and the actual state of the vehicle obtained from on-board sensors $(x(k))$. The parameters for local controller i are the matrix gain K_i, and the trim value of the actuator command $u_{trim,i}$.

The state of the vehicle is given by

$$x(k) = [x, y, z, \phi, \theta, \psi, u, v, w, p, q, r]^T \tag{2}$$

where x, y, z represent the positions, ϕ, θ, ψ the attitudes, u, v, w the velocities, and p, q, r the angular rates.

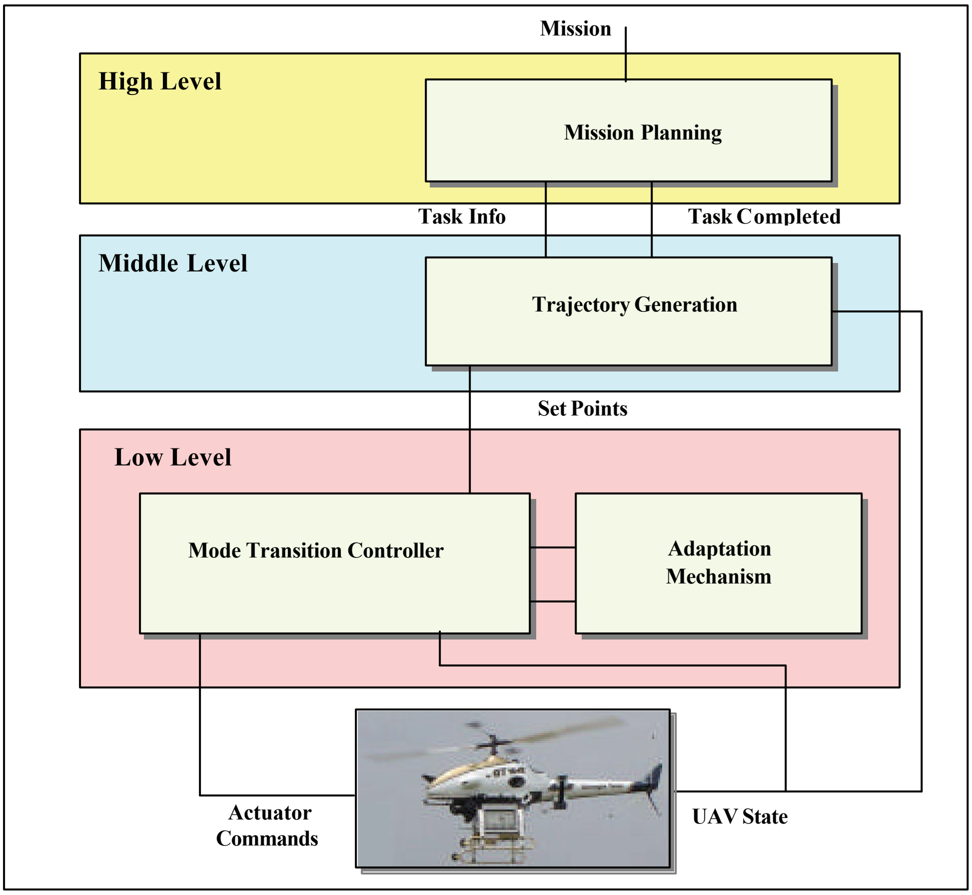


Figure 7. Overall architecture for the Adaptive Mode Transition Control

Once the operating state of a mode is decided, an approximate model of the vehicle is linearized about that state, and then discretized. A linear quadratic regulator is computed for the matrix gain K_i and the same design procedure is used for each mode. When an approximate model of the vehicle is not available, the linearized model could be obtained from a Fuzzy Neural Net model trained with input-output data from the actual vehicle.

The Mode Transition Manager (MTM) coordinates the transitions automatically based on the actual state of the vehicle. In order to accomplish this task, a Mode Membership Function is defined for each local mode and the MTM determines which local mode or transition should be activated relying upon these costructs.

When a local mode is active, the corresponding local controller is used to compute the control output whereas when a transition is active, the corresponding ACM is used to compute the control output.

The active control models are in charge of the transitions between local modes. The function of an active control model (ACM) is to blend the outputs of the local controllers corresponding to one transition in a smooth and stable way, that is, the blending of the local controllers should not deteriorate the overall performance of the closed loop system. Every ACM is linked to the local controllers corresponding to the transition, has access to their outputs, and also includes a Fuzzy Neural Net (FNN) that generates the blending gains to compute the control output. At run time, the FNN of the ACM is adapted on-line by the control adaptation mechanism.

The adaptation mechanism component calls the adaptation routines of the mode transition control and also includes the Active Plant Models (APMs, one for each transition), which serve as partial models of the plant in the transitions.

The APMs provide the sensitivity matrices required to adapt the ACMs and include a FNN that is trained to represent the dynamics of the vehicle in the transition. When the vehicle is in a transition, the input/output information from its sensors is used by the plant adaptation mechanism to train this model by calling the recursive least squares training routine from the FNN.

The control adaptation mechanism provides the adaptation function to the ACM's. When an ACM is active and the control adaptation mechanism is enabled, an optimization routine is used to find the optimal control value at each time step; the optimal blending gains that minimize the error between the optimal control and the control produced by the ACM are also computed. These optimal blending gains constitute the desired outputs for the recursive least squares training algorithm in the FNN, corresponding to that ACM, which is in turn called by the control adaptation mechanism.

The architecture has been implemented using the OCP. Figure 8 shows a software-in-the-loop simulation environment used to implement the architecture. Hardware-in-the-loop simulations and flight tests have been performed to validate the control algorithms. Typical results are depicted in Figure 9.

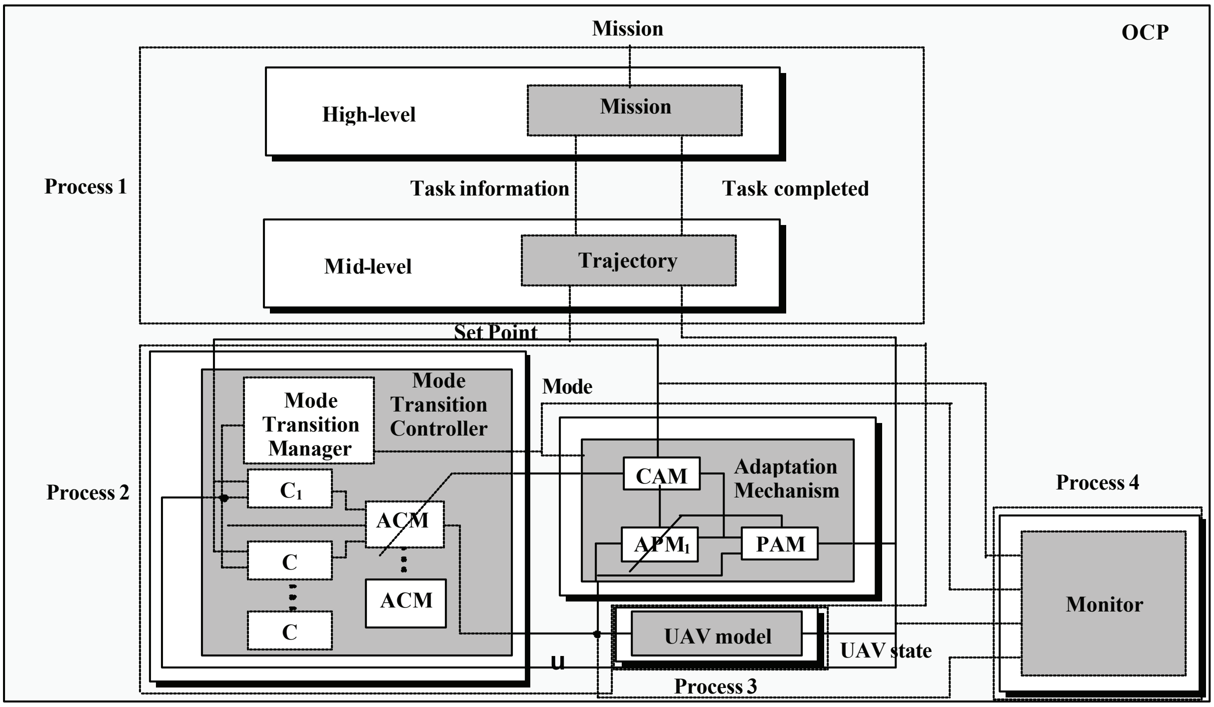
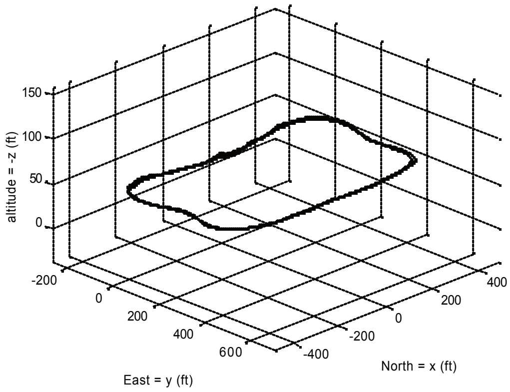
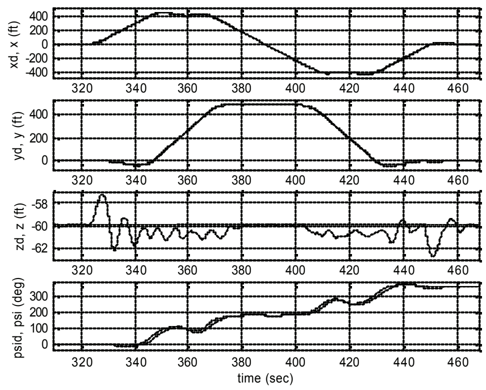


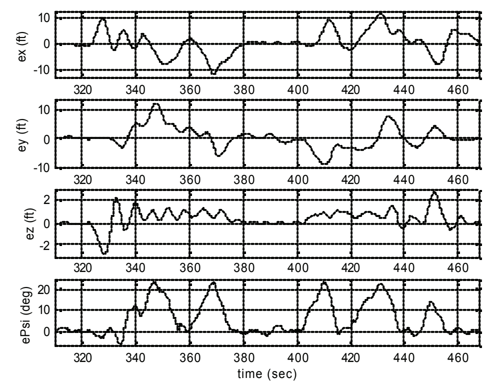
Figure 8. Adaptive Mode Transition Control Implementation on the OCP



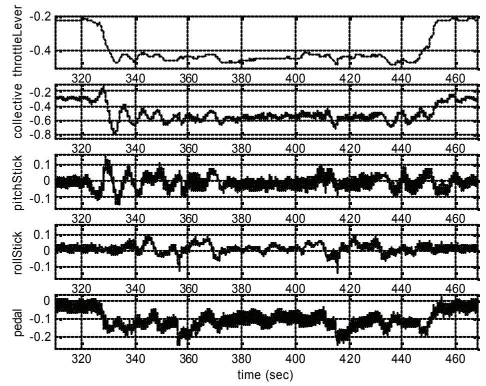
(a) Desired and actual 3D trajectory



(b) Desired and actual position and heading



(c) Position and heading errors



(d) Actuator commands

Figure 9. AMTC algorithms simulation results

6 FAULT TOLERANT CONTROL

UAVs are often subjected to failure modes that may lead to a catastrophic event resulting in loss of the vehicles. It is desired, therefore, to develop and implement technologies that will detect and identify in a timely manner on-board failure modes and reconfigure the available control authority so that the vehicle maintains an acceptable level of performance for the duration of the emergency (Clements, 2003). The overall hierarchical fault tolerant structure is shown in Figure 10.

Since faults are most likely to occur at the component level, a component-based modeling system is adopted followed by a mapping from component faults to system functional behaviors (Clements, et al., 2001.)

In this model, components are modeled individually with the interactions between components represented by an interconnection structure. A functional model of the system and the mapping from the structural model to this functional model, is then defined. Performance and stability criteria are specified on the functional model of the system. Next, a hybrid hierarchical control strategy is pursued consisting of a Fault Detection and Identification (FDI) routine and a Fault Tolerant Control (FTC) strategy. The FDI routine uses a wavelet neural net to detect and identify a fault condition, based on training data from simulated fault scenarios. The FTC routine consists of a hierarchical accommodation strategy. In the presence of a fault, the high-level redistribution controller re-routes the available control authority taking advantage of any inherent redundancy in the system. The mid-level set point controller then determines set point trajectories which maintain stability of the restructured system, possibly at some degraded performance. Finally, the low-level algorithm adjusts local controller gains in response to the new set points generated by the mid-level controller.

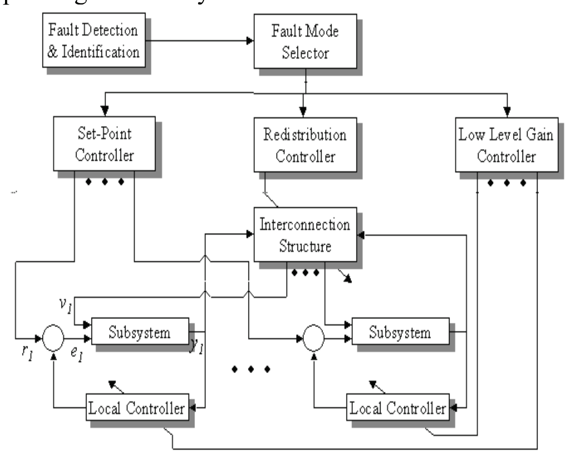


Figure 10. Hybrid hierarchical control structure

Simulation and flight test results demonstrate the effectiveness of the approach. A UAV (our GTMax) performs a bob-up maneuver. Three seconds into the maneuver, a stuck collective actuator fault is induced and the rpm controller of the helicopter's main rotor is activated. Without control reconfiguration, the vehicle becomes unstable and crashes. With control reconfiguration, the fault is accommodated and the UAV completes successfully the maneuver. Figure 11 depicts the results of the flight tests.

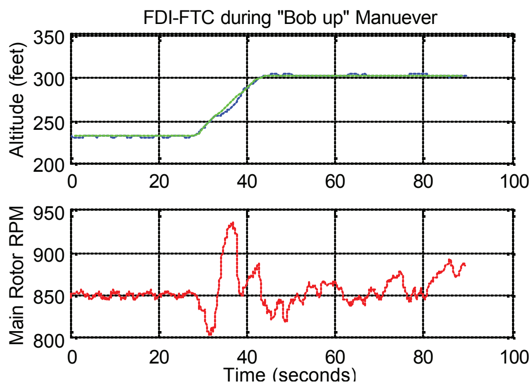


Figure 11. FDI/FTC flight test result

7 THE OPEN CONTROL PLATFORM

The Open Control Platform (OCP) is being developed for use as a software platform enabling demonstration and evaluation of advanced UAV control systems technologies being developed for the DARPA Software Enabled Control (SEC) program. The OCP enabling software technology is being developed by a team from industry and academia, led by Boeing Phantom Works, and including the Georgia Institute of Technology, Honeywell, Laboratories, and the University of California Berkeley. The OCP provides a middleware-base execution framework, Application Programmer Interfaces (APIs) simulation tools, and integration with useful software and control systems design and development tools. These OCP capabilities and features are being delivered to the control systems technology teams on the SEC program, to provide them with a platform to enable rapid development, test and migration of advanced control systems designs to embedded software. Recent tests and demonstrations of the OCP are described, including flight tests by Georgia Tech on the GTMax unmanned helicopter, and a collaborative demonstration effort between the Air Force Research Laboratory, Boeing, and Northrop Grumman (Wills, et al., 2000a, 2000b, 2002).

The OCP architecture that is being developed will provide the distributed control and reconfigurable architecture that allows control and intelligence algorithms at all levels and timescales to interact in a decoupled, real-time and distributed fashion. The mid-level SEC algorithms for Mode Transitioning and Fault Tolerant Control illustrated in Figure 12 attempt to raise the degree of autonomy of UAVs by developing control algorithms that can handle abnormal situations.

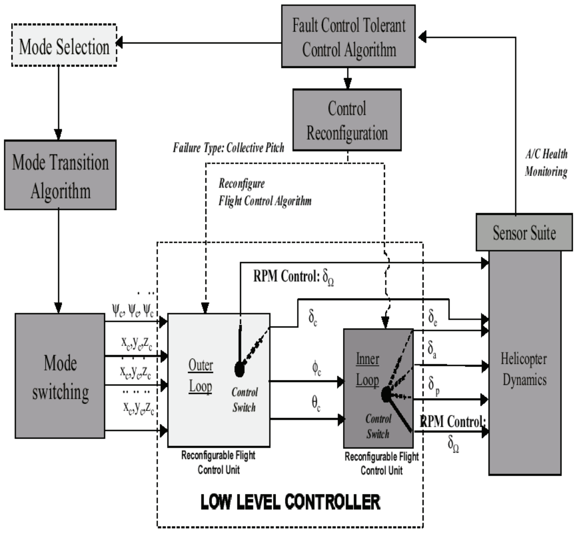


Figure 12. Flight control reconfiguration

Figure 13 shows a schematic of the required distributed control functionality, for the dynamic reconfiguration of modules through the OCP.

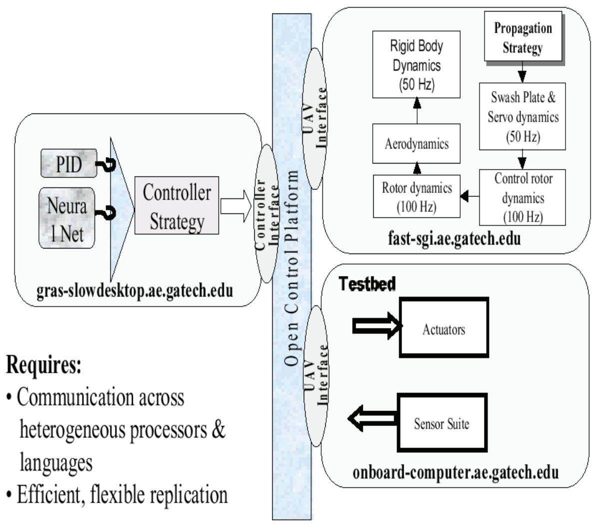


Figure 13. Distributed control functionality

8 COORDINATED CONTROL OF MULTIPLE UAVS

Swarms of heterogeneous UAVs may be commanded in the near future to execute complex missions in uncertain and unfriendly environments. R&D activities are currently under way worldwide to address challenging issues arising from the utility of multiple vehicles for surveillance, reconnaissance, rescue and often application domains. The Georgia Tech research team has been developing a novel architecture for the coordinated control of multiple Unmanned Aerial Vehicles (UAVs) and a differential game theoretical approach to formation control and collision avoidance (Vachtsevanos and Tang, 2004). The hierarchical architecture features an upper level with global situation awareness and team mission planning, a middle level with local knowledge, formation control and obstacle avoidance, and a low level that interfaces with onboard baseline controllers, sensors, communication and weapon systems. Each level consists of several interacting agents with dedicated functions. The formation control problem is viewed as a Pursuit Game of n pursuers and n evaders. Stability of the formation of vehicles is guaranteed if the vehicles can reach their destinations within a specified time, assuming that the destination points are avoiding the vehicles in an optimal fashion. A two-vehicle example is used to illustrate the approach. Collision avoidance is achieved by designing the value function so that it ensures that the two vehicles move away from one another when they come too close to each one. Simulation results are suggested to verify the performance of the proposed algorithm.

9 CONCLUSION

Unmanned Aerial Vehicles present major challenges to the designer and the end user. They require new and novel technologies to be developed, tested and implemented if such vehicles will perform actual missions reliably and robustly. Autonomy stands out as the key requirement with enabling technologies to allow such vehicles to operate safely in unstructured environments within their flight envelope, to accommodate subsystem/component failure modes without major performance degradation or loss of vehicle and to perform extreme maneuvers without violating stability limits. An integrated/hierarchical approach to vehicle instrumentation, computing, modeling and control seems to provide possible solutions. The UAV community is accomplishing major milestones towards this goal but key R&D concerns remain to be addressed. More recently, researchers have been concerned with multiple and heterogeneous UAVs flying in formation in order to take advantage of their complementary capabilities. The UAV swarm problem opens now avenues of research where the intelligent control community can contribute significantly in terms of smart coordination/cooperation technologies.

Acknowledgement

The authors would like to gratefully acknowledge the support received from Dr. John. S. Bay, DARPA, and Mr. William Koenig, AFRL. In addition, the authors would like to thank their Co-P.I.'s on this research project: Dr. J.V.R. Prasad, School of Aerospace Engineering, and Drs. Linda Wills and Bonnie Heck, School of Electrical and Computer Engineering, Georgia Tech; and Mr. Bryan Doerr, Mr. Brian Mendel, and Mr. James L. Paunicka, Boeing Phantom Works.

References

Clements, N. S., Heck, B., Vachtsevanos, G. (2001), Hierarchical Fault Tolerant Control Using Component Based Models, in *Proceedings of the 2001 European Control Conference*.

Clements, N. S. (2003), *Fault Tolerant Control of Complex Dynamical Systems Fault Tolerant Control of Complex Dynamical Systems*, Ph.D. Thesis, Georgia Tech.

DoD (2002), *UAV Roadmap 2002-2027*, Office of the Secretary of Defense (Acquisition, Technology, & Logistics), Air Warfare.

Gutierrez, L., Vachtsevanos, G., Heck, B. (2003a), An Approach to the Adaptive Mode Transition Control of Unmanned Aerial Vehicles, in *Proceedings of 2003 American Control Conference*, Denver, Colorado.

Gutierrez, L., Vachtsevanos, G., Heck, B. (2003b), A Hierarchical Architecture for the Adaptive Mode Transition Control of Unmanned Aerial Vehicles, in *Proceedings of AIAA Guidance, Navigation and Control Conference*, Austin, Texas.

Rufus, F., Vachtsevanos, G., and Heck, B. (2000a), Adaptive Mode Transition Control of Nonlinear Systems Using Fuzzy Neural Networks, *8th IEEE Mediterranean Conference on Control and Automation*, Patras, Greece.

Rufus, F., Heck, B., and Vachtsevanos, G., (2000b), Software-enabled Adaptive Mode Transition Control for Autonomous Unmanned Vehicles, in *Proceedings 19th Digital Avionics Systems Conference*, Vol. 1, pp. 1.E.1-1 -1.E.1-8.

Rufus, F., (2001), *Intelligent approaches to mode transition control*, Ph. D. Thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA.

Rufus, F., Vachtsevanos, G., and Heck, B. (2002), Real-time Adaptation of Mode Transition Controllers, *Journal of Guidance, Control, and Dynamics,* 25(1), 167-75.

Schrage, D. P., et al. (1997) *Autonomous Scout Rotorcraft Testbed (ASRT) Final Report*, Georgia Tech, July 24, 1997.

Schrage, D. P. and Vachtsevanos, G. (1999). Software Enabled Control for Intelligent UAVs. In *1999 IEEE International Conference on Control Applications.*

Vachtsevanos, G., Kim, W., Al-Hasan, S., Rufus F., and Simon, M. (1997a), Autonomous vehicles: From flight control to mission planning using fuzzy logic techniques, *International Conference on Digital Signal Processing, DSP '97,* Santorini, Greece. (pp. 977-981)

Vachtsevanos, G., Kim, W., Al-Hasan, S., Rufus, F., Simon, M., Schrage, D., and Prasad, J.V.R. (1997b), Mission Planning and Flight Control: Meeting the Challenge with Intelligent Techniques, *Journal of Advanced Computational Intelligence,* 1(1), 62-70.

Vachtsevanos, G., Tang, L., and Reimann, J., (2004), An Intelligent Approach to Coordinated Control of Multiple Unmanned Aerial Vehicles, *American Helicopter Society 60th Annual Forum*, Baltimore, MD.

Wills, L., Sander, S., Kannan, S., Kahn, A., Prasad, J.V.R., and D. Schrage (2000a), An Open Control Platform for Reconfigurable, Distributed, Hierarchical Control Systems, *Proceedings 19th Digital Avionics Systems Conferences*, Vol. 1, pp. 4D2/1 - 4D2/8, Philadelphia, PA.

Wills, L., Kannan, S., Heck, B., Vachtsevanos, G., Restrepo, C., Sander, S., Schrage, D., and Prasad, J.V.R. (2000b), An Open Software Infrastructure for Reconfigurable Control Systems, *Proceedings of the 2000 American Control Conference*, Vol.4, pp.2799-2803.

Wills, L., Kannan, S., Sander, S., Guler, M., Heck, B., Prasad, J.V.R, Schrage, D., and Vachtsevanos, G. (2001), An Open Platform for Reconfigurable Control, *IEEE Control Systems Magazine*, 21(3), 49-64.