FUSION OF HARD AND SOFT CONTROL FOR UNINHABITED AERIAL VEHICLES

Randy C. Hoover, Marco P. Schoen, and D. Subbaram Naidu

Measurement and Control Engineering Research Center,
Idaho State University, Pocatello, ID 83209, USA

Abstract: This paper develops a new approach by integrating hard and soft control strategies for Guidance, Navigation, and Control of Uninhabited Aerial Vehicles (UAVs). The desired features from both soft and hard control are incorporated in the development of a navigation controller and UAV autopilot. The navigation controller is responsible for the generation of flyable trajectories based on desired GPS waypoint destinations. The UAV autopilot is responsible for steering the UAV along these desired trajectories while maintaining stable control. *Copyright © 2005 IFAC*

Keywords: Autonomous vehicles, Fuzzy control, Intelligent control, Adaptive control

1. INTRODUCTION

Uninhabited aerial vehicles including those with combat capabilities have become a topic of research in recent years for both governmental agencies and the commercial sector (Naidu, *et al*. 1997; Wilson 2003). UAV's are capable of carrying out various missions including surveillance and intelligence, search and rescue, atmospheric data collection, and agricultural data collection. Even more recently, research has been conducted to study the viability of using UAV's in combat situations where the uninhabited combat aerial vehicle (UCAV) would not only be used for surveillance, but also engage in enemy targets while avoiding pop-up threats. The latter research requires the UCAV to compute online trajectory generation for several different coordinates, detect pop-up threats, and regenerate feasible trajectories in real time to avoid such threats. It has been shown that real time trajectory generation is feasible for future UCAV development (Rathbun and Capozzi, 2002; Prasanth, *et al.* 2001).

The current paper entails the development of an intelligent autopilot used to control and stabilize the UAV, and a navigation controller to generate appropriate heading and altitude trajectories for the autopilot to follow based on desired geographical destinations or enemy targets. The navigation controller uses the advantages of fuzzy logic to generate flyable trajectories for the UAV. The UAV autopilot uses the advantages of fuzzy logic as well as adaptive control to steer the UAV along such trajectories while maintaining complete control. It should be noted that the navigation controller in the present paper has been developed for the purpose of testing the UAV autopilot only. The navigation controller and autopilot are tested on two different airframes, the Aerosonde from Aerosonde Pty Ltd., and the North American Navion from North American Aviation. The autopilot developed in this paper has the ability to adapt to the airframe under control with little or no modification. Simulation results are shown for both of the above listed airframes.

2. NAVIGATION CONTROLLER

As mentioned previously, the navigation controller at the time of this paper has been developed for the testing of the autopilot only, and has not been optimized. The current navigation controller consists of two main parts, a waypoint generator, and a trajectory generator.

The waypoint generator (WG) simply updates the current waypoint to the next waypoint to be achieved. The entire waypoint set is generated by the UAV user and stored in a list. Once a waypoint has been reached, the WG updates the list and the UAV is steered to the next waypoint. It should be noted that the waypoint does not have to be reached explicitly, but rather a sphere has been developed around the waypoints center in which if the UAV reaches the "tolerance" sphere, the waypoint is considered reached and the WG updates to the next waypoint. The size of the sphere used in the current simulations has a diameter of 4m.

The second part of the navigation controller is to generate desired trajectories for the UAV to track in terms of UAV heading (ψ_{UAV}) and UAV altitude (H_{UAV}). The trajectories are generated based on a desired set of coordinates to be reached by the UAV, $C_d = [Lat_d \quad Lon_d \quad H_d]$, where *Lat*$_d$, *Lon*$_d$, and *H*$_d$ are the desired latitude, longitude, and altitude respectively. Once the UAV reaches the current coordinate vector *C*$_d$ within the tolerance sphere, the vector is updated and new reference trajectories are generated.

The trajectories of the navigation controller are generated based on the current UAV position and the desired UAV position. The first signal generated is the altitude trajectory in which the altitude error is computed as $e_H = H_d - H_{UAV}$, from this error, a desired altitude trajectory is generated as

$$\phi_N = H_{UAV} + f_\phi(e_H) \tag{1}$$

where $f_\phi(e_H)$ is defined by fuzzy set theory.

The second signal generated by the navigation controller is the desired heading trajectory in which the heading error is calculated based on the current UAV position, the desired UAV position, and the current UAV heading. A triangulation method has been used to compute the error in the *lat*/*lon* plane (Hoover, 2004). The error is calculated as

$$e_\psi = \psi_{wp} - \lambda \ , \tag{2}$$

where

$$\psi_{wp} = \tan^{-1}\left(\frac{\kappa}{\sigma}\right)\frac{180}{\pi} \ , \tag{3}$$

and

$$\lambda = \psi_{UAV} + \beta \quad . \tag{4}$$

In Eq. 3, ψ_{wp} is the waypoint heading with respect to the UAV position updated continuously as the UAV travels in the latitude/longitude plane, and κ and σ are defined as

$$\kappa = Lon_d - Lon_{UAV} \, , \sigma = Lat_d - Lat_{UAV} \quad . \tag{5}$$

In Eq. 4, the actual UAV heading is calculated as the current UAV heading ψ_{UAV} with the addition of any UAV sideslip β (Stevens and Lewis, 2003). From Eq. 2, the new heading trajectory can be generated as

$$\psi_N = \psi_{UAV} + f_\psi(e_\psi), \tag{6}$$

where $f_\psi(e_\psi)$ is defined by fuzzy set theory. The complete navigation portion of the controller can be seen in Fig. 1. The blocks labeled FNS are defined as the fuzzy navigation system and are discussed in detail in the next section.

3. FUZZY NAVIGATION SYSTEM DESIGN

The fuzzy navigation system is based on a Mamdani fuzzy system using a centroid of area (COA) defuzzification scheme defined as (Jang, *et al.* 1997)

$$COA = \frac{\int_Z \mu_C(z) z dz}{\int_Z \mu_C(z) dz} \ , \tag{7}$$

where $\mu_C(z)$ is the output membership function (MF) of the Mamdani fuzzy system, and *z* is the MF input. The MFs used are a combination of Gaussian MFs on the input layer, and triangular MFs on the output layer.

The FNS for the altitude portion of the navigation system is a single-input, single-output (SISO) fuzzy logic controller (FLC). The input to the FLC is the altitude error, and the output is the incremental step change in UAV altitude. The output of the FLC is then summed with the current UAV altitude to generate smooth flyable trajectories for the UAV autopilot to track Fig 2. The FLC consists of the following five rules:

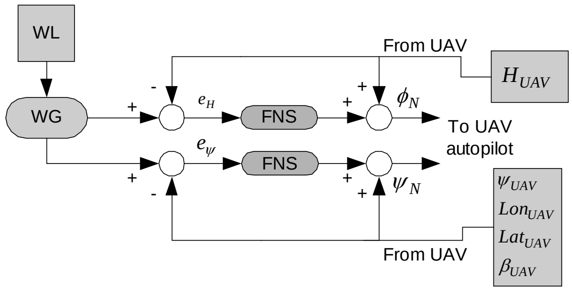

Fig. 1. Navigation controller block diagram

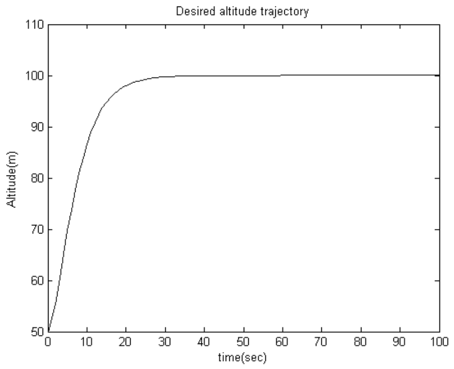

Fig. 2. Desired altitude trajectory for UAV to track for a change from 50m to 100m

If e_H is *PL* then ϕ_N is *PL*: large positive errors
If e_H is *PM* then ϕ_N is *PM*: medium positive errors
If e_H is *Z* then ϕ_N is *Z*: errors near zero
If e_H is *NM* then ϕ_N is *NM*: medium negative errors
If e_H is *NL* then ϕ_N is *NL*: large negative errors

The FNS for the heading portion of the navigation system is a SISO system defined by an FLC. The system input is the heading error e_ψ with respect to the desired waypoint, and the output is the incremental change in heading. The output of the FLC is then summed with the current UAV heading ψ_{UAV} to generate the desired heading change ψ_N. A sample 6-point heading error minimization trajectory is shown in Fig 3. The FLC consists of the following five rules:

If e_ψ is *PL* then ψ_N is *PL*: large positive errors
If e_ψ is *PM* then ψ_N is *PS*: small positive errors
If e_ψ is *Z* then ψ_N is *Z*: errors near zero
If e_ψ is *NM* then ψ_N is *NS*: small negative errors
If e_ψ is *NL* then ψ_N is *NL*: large negative errors

4. UAV AUTOPILOT

The UAV autopilot is designed to accept heading and altitude reference trajectories from the navigation system previously discussed. The autopilot will then steer the UAV along these trajectories while maintaining stable control of the airframe. The main controller consists of five sub-controllers that control the UAV attitude, roll, and velocity to achieve near perfect tracking. The controller design is also adaptive and has been shown to adapt to the given airframe without the need for major parameter adjustments. The five sub-controllers are heading, velocity, altitude-to-pitch, pitch-to-elevator, and bank-to-aileron. All five controllers utilize benefits from fuzzy set theory, as well as adaptive control theory to achieve the desired results. The entire block diagram is shown in Fig. 4.

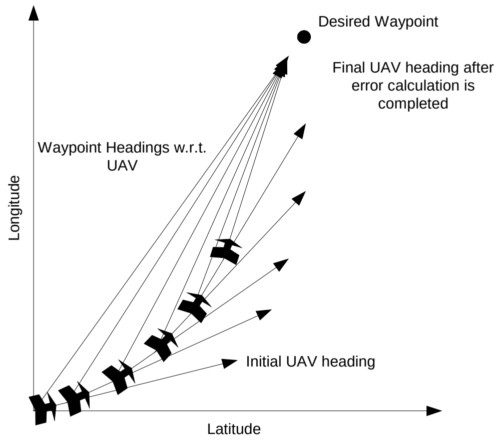
Desired Waypoint
Final UAV heading after error calculation is completed
Waypoint Headings w.r.t. UAV
Longitude
Initial UAV heading
Latitude

Fig. 3. Sample heading error minimization trajectory

The heading controller is a single-input, multi-output (SIMO) adaptive fuzzy logic controller (AFLC). The heading controller input is the error between the reference signal generated from the navigation system denoted by ψ_N, and λ, where λ is defined by Eq. 4. The controller then generates appropriate commands to control the bank angle and rudder actuator. Since the only real direct measurement of rudder actuation is measured in the UAV sideslip β, there is no direct feedback needed to achieve appropriate tracking, i.e. the heading controller directly controls the rudder actuator. The bank angle however is directly measured and direct feedback can be applied, therefore, the heading controller generates appropriate bank angle commands, and delivers them as inputs to the bank-to-aileron controller. The bank-to-aileron controller, as well as the additional four sub controllers, is a SISO AFLC. The input to the controller is the error between the bank angle reference command θ_d generated from heading controller, and the actual UAV bank angle θ_{UAV}. The output of the bank-to-aileron controller is the input signal for the aileron actuators. The output of the controller is limited to $\pm\pi/4$ to decrease the risk of actuator saturation.

The altitude-to-pitch controller is responsible for generating appropriate pitch reference trajectories for the pitch-to-elevator controller to track based upon the current UAV altitude, and the desired UAV altitude. The input signal to the altitude-to-pitch controller is error between the altitude reference trajectory generated from the navigation controller denoted here as ϕ_N, and the actual UAV altitude H_{UAV}. The pitch-to-elevator controller is responsible for driving the elevator actuator in an attempt to achieve perfect pitch tracking. The input signal to the pitch-to-elevator controller is the error between the pitch angle reference signal ϕ_d generated from the altitude controller, and the actual UAV pitch angle ϕ_{UAV}. The output of the controller is limited to $\pm\pi/4$ to decrease the risk of actuator saturation.

The velocity controller acts as a velocity hold controller in which the user determines the desired airspeed. The input to the controller is the error between a desired reference value V_d, and the actual UAV airspeed V_{UAV}. The velocity reference value is set to a specified value based upon the airframe

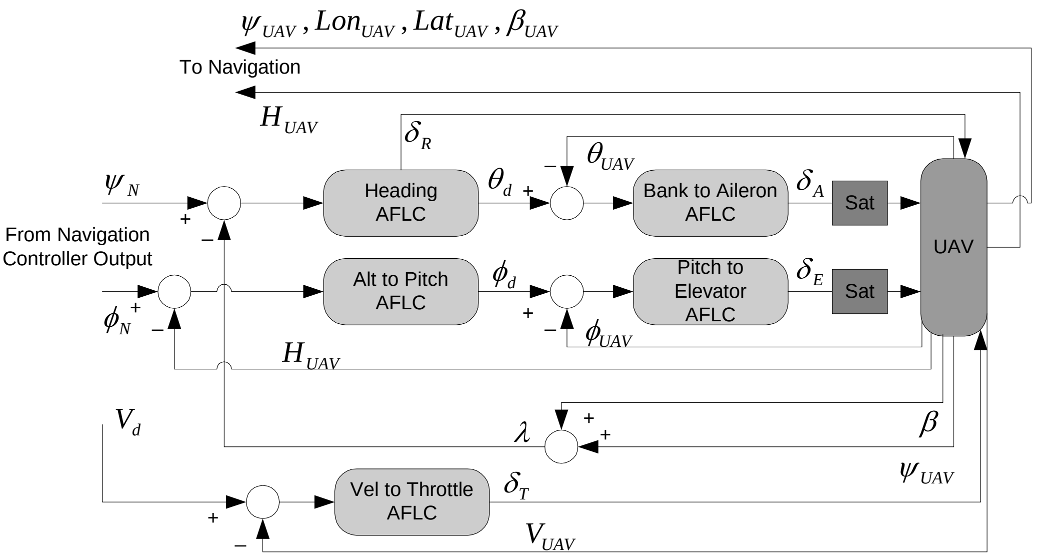

Fig. 4. UAV autopilot block diagram

under control. For the simulations performed in this paper, this value is approximately 75% of the UAV maximum airspeed. The output of the velocity controller drives the UAV throttle directly.

5. UAV AUTOPILOT DESIGN

The UAV autopilot design is based on a Mamdani fuzzy system using a COA defuzzification scheme defined by Eq. 7. The autopilot also utilizes benefits from adaptive control theory to compensate for unknown parameters. A similar approach was developed by Kim and Yuh (2002) in which an adaptive fuzzy controller was able to steer an autonomous underwater vehicle along a desired path. The adaptive capabilities are needed here to reduce any error between a desired reference input and the actual UAV output. All five of the sub-controllers have the same basic structure, the MFs used are a combination of Gaussian MFs on the input layer, and triangular MFs on the output layer. The differences in each controller are based on the position of the MFs, and the parameters determining there shape (Hoover, 2004), i.e. a Gaussian MF has the form

$$f(x,\sigma,c) = e^{-(x-c)^2/\sigma^2}, \qquad (8)$$

where c determines the center of the MF, and σ determines the shape. The output MFs are triangular in shape and again have different center and shape characteristics based on controller design.

As mentioned previously, all five controllers have adaptive capabilities built into there design. The adaptive portion of the controller adjusts the centers of the output MFs to minimize tracking error. The adaptive feature is necessary in the development of the fuzzy control design since exact actuator displacement cannot be determined prior to UAV flight. For example, Fig. 5 shows the input/output characteristics of the pitch-to-elevator controller. From the output surface it can be seen that when the error is zero, the elevator input is also zero. This is due to the centering of the output MFs about the zero reference. The problem with this is if the pitch reference signal calls for a pitch angle of 10 (deg), the elevator input would need to be greater than zero to allow for perfect tracking. Since the exact elevator input needed to achieve perfect tracking is considered to be an unknown parameter, the output MFs are updated to generate the appropriate elevator actuator signal.

The adaptation law utilized in this paper is an extension to the Lyapunov rule for model reference adaptive control (MRAC) design. Since perfect tracking of input reference signal is required, the adaptive control design could be thought of as a MRAC with the reference model being unity. This is due to the fact that in general terms the adaptation scheme used in MRAC evaluates the error between the system output, and the output of a reference model that has been developed by the control designer to achieve desired performance. The

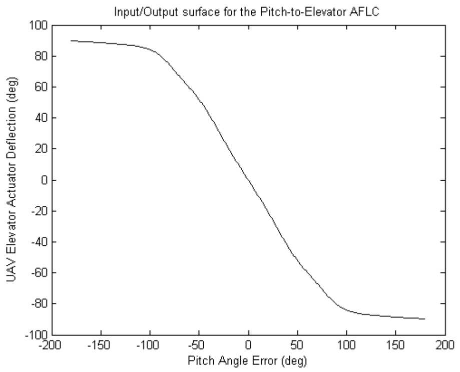

Fig. 5. Input/Output characteristics of the Pitch-to-Elevator AFLC. Error vs. Elevator input.

controller parameters are then updated based on this error. In the case of perfect tracking however,

$$y_m - y_{UAV} \equiv y_r - y_{UAV} , \tag{9}$$

where y_m is the model output, y_r is the reference signal input, and y_{UAV} is the output of the UAV in generic terms. Based on Eq. 9, a reference model is no longer needed when trying to achieve perfect tracking performance.

The adaptation scheme utilized is as follows:

Consider the instantaneous error between the desired the actual output of the UAV. The derivative of the square of this error is (Farinwata, *et al.* 2000)

$$\frac{\partial e_k^2}{\partial w_i} = 2e_k \frac{\partial e_k}{\partial w_i} = -2\Phi_k^i e_k , \tag{10}$$

where Φ_k^i is the output weights of the i^{th} MF, and k is the discrete time interval based upon the sample time of the system. From this, it can be seen that the output weight of the i^{th} rule should be adjusted as (Farinwata, *et al.* 2000)

$$w_{k+1}^i = w_k^i + \gamma * \Phi_k^i * e_k , \tag{11}$$

where γ is considered to be the adaptation rate. The adaptation law in Eq. 11 is used in all five sub-controllers with the difference being in the parameter vector Φ , and the adaptation rate γ .

Since all five sub-controllers are designed in a similar fashion with only the MF parameters making them different, a general outline of their design will be presented here. The sub-controllers, with the exception of the heading controller, are all SISO AFLC. The input to each controller is the error between the reference signal, and the actual signal measured from the UAV. The output of each controller is either a reference trajectory for the input of a lower level controller, or a signal to drive an actuators control surface. Each controller uses five Gaussian MFs on the input layer, and five triangular MFs on the output layer. The output layer parameters are then adapted based upon the learning rate γ, and the input error. Each controller is designed using Mamdani max/min relations, COA defuzzification, and five rules. The rules for each controller are of the form

If e_x is *PL* then Φ_y is *PL*: large positive errors
If e_x is *PM* then Φ_y is *PS*: small positive errors
If e_x is *Z* then Φ_y is *Z*: errors near zero
If e_x is *NM* then Φ_y is *NS*: small negative errors
If e_x is *NL* then Φ_y is *NL*: large negative errors

where e_x is the error between the reference signal and its respective output, and Φ_y is the output weight response for the AFLC. The output weights are then updated according to the error minimization process.

The heading controller is a SIMO AFLC system in which the bank angle output is a reference signal for the bank-to-aileron controller to track, and the rudder output drives the rudder actuator control surface directly. Mamdani max/min relations are used with the COA defuzzification scheme, and five rules for each signal being generated. The rules for the heading controller are of the form

If e_x is *PL* then Θ_y is *PL* and δ_R is *PL*
If e_x is *PM* then Θ_y is *PS* and δ_R is *SL*
If e_x is *Z* then Θ_y is *Z* and δ_R is *Z*
If e_x is *NM* then Θ_y is *NS* and δ_R is *NS*
If e_x is *NL* then Θ_y is *NL* and δ_R is *NL*

where e_x is the error between the navigation control reference trajectory, and λ defined by Eq. 4, Θ_y is the output weight response for the AFLC, and δ_R is the rudder actuator deflection.

6. SIMULATION RESULTS

The fuzzy navigation controller and adaptive fuzzy autopilot are tested using two fully nonlinear aircraft models. The first model is the Aerosonde UAV that has been used to carry out several missions with a variety of objectives. The second model is the North American Navion that is a bottom wing design and has been used by the USAF for reconnaissance and light cargo carrying. Both aircraft mathematical models were developed by Unmanned Dynamics and are included in their AeroSim blockset for use with Matlab/Simulink. The first simulation is a nonplanar trajectory in which the Aerosonde UAV is started at waypoint W1, and is to be driven to waypoints W2 through W7. The trajectory flown verifies that the UAVs navigation controller and autopilot can generate flyable trajectories in all four quadrants of the latitude, longitude plane. The UAV trajectory with desired waypoints are shown in Fig. 6, with the bank angle and pitch angle tracking references in Figs. 7 and 8 respectively.

The second simulation is of the same nonplanar trajectory but the airframe under control has changed to the North American Navion. As shown in Fig. 9, the Navion is able to fly the trajectory hitting every waypoint at the prescribe altitude, latitude, and longitude. It should be noted that the Aerosonde and the Navion are two very different airframes with different flight characteristics, but the AFLC autopilot is able to adapt to control each airframe with no human intervention. It can also be seen from Figs. 7,8, and 10, that the low level controllers that drive the control surface actuators are able to track almost perfectly the reference inputs generated by the mid level controllers. Although no simulation results have been presented here, the UAV autopilot has also shown very promising results in the event of atmospheric disturbances.

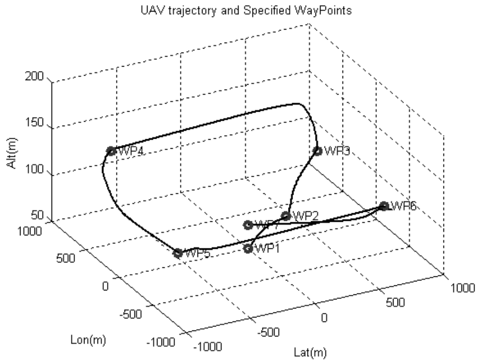

Fig. 6. Aerosonde UAV trajectory with desired 7-waypoint crossing

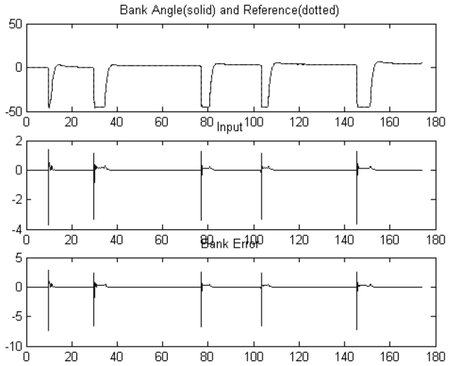

Fig. 7. UAV bank angle, reference input generated by the heading AFLC, aileron actuator input, and error

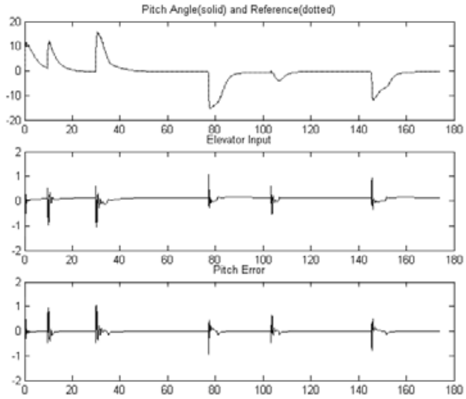

Fig. 8. UAV pitch angle, reference input generated by the Alt-to-Pitch AFLC, elevator actuator input, and error

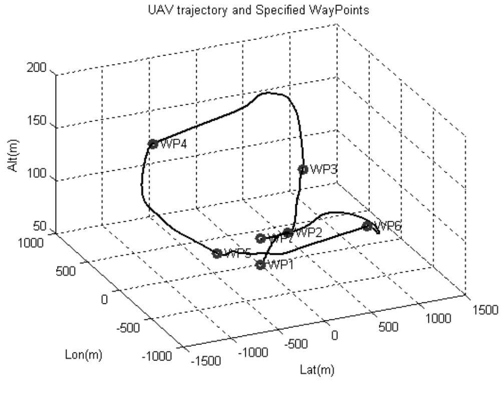

Fig. 9. North American Navion flying the same 7-waypoint trajectory as the Aerosonde

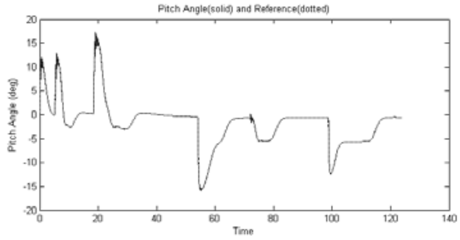

Fig. 10. North American Navion pitch angle and reference input for 7-waypoint trajectory

7. CONCLUSION

This paper presents a navigation controller and autopilot design for uninhabited aerial vehicles that draws benefits from both hard and soft control techniques. The navigation controller incorporates the use of fuzzy logic to generate heading and altitude trajectories based on desired waypoint destinations. The UAV autopilot utilizes fuzzy logic as well as adaptive control to steer the UAV along these reference trajectories while maintaining complete control of the UAV. It has been shown that the autopilots adaptive capabilities can control multiple aircraft configurations without the need to adjust major controller parameters. Future research will focus on the feasibility of producing optimal reference trajectories with improved performance characteristics, and updating the velocity controller to allow more extensive UAV maneuvers.

REFERENCES

Farinwata, S. S., D. Filev, and R. Langari (2000). *Fuzzy Control: Synthesis and Analysis*, pg. 341. John Wiley & Sons, England.

Hoover, R. C. (2004). *Fusion of Hard and Soft Computing for Guidance, Navigation, and Control of Uninhabited Aerial Vehicles.* Master Thesis, Idaho State University, Pocatello, ID.

Jang, J. –S. R., C. –T. Sun, and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, pg. 75. Prentice Hall, Upper Saddle River, NJ.

Kim T. W., and J. Yuh (2002). Application of on-line neuro-fuzzy controller to AUVs. *Information Sciences*, **Vol. 145**, 169-182.

Naidu, D. S., Banda, S. S., and Chandler, P. R. (1997). *Guidance and control strategies for tactical unmanned air vehicles: Research areas,* Technical Report T. R. 97/98 008, Idaho State University, Pocatello, Idaho.

Prasanth, R. K., J. D. Boskovic, S. Li, and R. K. Mehra (2001). *Initial Study of Autonomous Trajectory Generation for Unmanned Aerial Vehicles*, Proceedings of the40th IEEE Conference on Decision and Control.

Rathbun, D. and B. Capozzi (2002). Evolutionary Approaches to Path Planning Through Uncertain Environments, *AIAA UAV conference in Portsmouth VA*, #2002-3455.

Stevens, B. L., and F. L. Lewis, (2002). *Aircraft Control and Simulation*, Wiley, New York, NY.

Wilson, J. R., (2003). *UAVs: A worldwide roundup*. Aerospace America, **Vol. 6**, 30-35.